

---

# Documentation de Bugzilla

*Version 5.0*

**L'équipe Bugzilla**

12 July 2015



|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>À propos de ce guide</b>                                     | <b>1</b>  |
| 1.1      | Introduction . . . . .  | 1         |
| 1.2      | Évaluer Bugzilla . . . . .                                      | 1         |
| 1.3      | Obtenir plus d'aide . . . . .                                   | 1         |
| 1.4      | Conventions du document . . . . .                               | 1         |
| 1.5      | Licence . . . . .   | 2         |
| 1.6      | Crédits . . . . .   | 2         |
| <br>     |   |           |
| <b>2</b> | <b>Guide utilisateur</b>  | <b>3</b>  |
| 2.1      | Créer un compte . . . . .                                       | 3         |
| 2.2      | Rapporter des bogues . . . . .                                  | 4         |
| 2.3      | Anatomie d'un bogue . . . . .                                   | 5         |
| 2.4      | Éditer un bogue . . . . .                                       | 8         |
| 2.5      | Rechercher des bogues . . . . .                                 | 10        |
| 2.6      | Rapports et graphiques . . . . .                                | 14        |
| 2.7      | Trucs et astuces . . . . .                                      | 15        |
| 2.8      | Préférences utilisateur . . . . .                               | 16        |
| 2.9      | Extensions installées . . . . .                                 | 18        |
| <br>     |   |           |
| <b>3</b> | <b>Guide d'installation et de maintenance</b>                   | <b>19</b> |
| 3.1      | Démarrage rapide (Ubuntu Linux 14.04) . . . . .                 | 19        |
| 3.2      | Linux . . . . .   | 23        |
| 3.3      | Windows . . . . .   | 27        |
| 3.4      | Mac OS X . . . . .  | 30        |
| 3.5      | Configuration post-installation essentielle . . . . .           | 33        |
| 3.6      | Configuration post-installation facultative . . . . .           | 35        |
| 3.7      | Migrer à partir d'un autre logiciel de suivi de bogue . . . . . | 37        |
| 3.8      | Déplacer Bugzilla d'une machine à une autre . . . . .           | 37        |
| 3.9      | Mise à jour . . . . .   | 38        |
| 3.10     | Sauvegardes . . . . .   | 50        |
| 3.11     | Contrôle d'intégrité . . . . .                                  | 51        |
| 3.12     | Fusionner les comptes . . . . .                                 | 51        |
| 3.13     | Une installation, plusieurs instances . . . . .                 | 51        |
| <br>     |   |           |
| <b>4</b> | <b>Guide d'administration</b>                                   | <b>53</b> |
| 4.1      | Paramètres . . . . .  | 53        |
| 4.2      | Préférences par défaut . . . . .                                | 63        |
| 4.3      | Utilisateurs . . . . .  | 64        |

|           |  |            |
|-----------|--|------------|
| 4.4       | Catégories, produits, composants, versions et jalons . . . . .       | 67         |
| 4.5       | Étiquettes . . . . .   | 72         |
| 4.6       | Champs personnalisés . . . . .                                       | 74         |
| 4.7       | Valeurs autorisées . . . . .   | 76         |
| 4.8       | Workflow des états de bogue . . . . .                                | 76         |
| 4.9       | Groupes et restrictions de groupes . . . . .                         | 77         |
| 4.10      | Mots-clés . . . . .  | 79         |
| 4.11      | Notifications . . . . .  | 80         |
| 4.12      | Citations . . . . .  | 82         |
| 4.13      | Extensions installées . . . . .                                      | 82         |
| <b>5</b>  | <b>Guide d'intégration et de personnalisation</b>                    | <b>83</b>  |
| 5.1       | FAQ sur la personnalisation . . . . .                                | 83         |
| 5.2       | Langues . . . . .  | 83         |
| 5.3       | Thèmes . . . . .   | 84         |
| 5.4       | Templates . . . . .  | 84         |
| 5.5       | Extensions . . . . .   | 88         |
| 5.6       | API . . . . .  | 90         |
| <b>6</b>  | <b>WebService API Reference</b>                                      | <b>93</b>  |
| 6.1       | Core API v1 . . . . .  | 93         |
| <b>7</b>  | <b>MySQL</b>   | <b>157</b> |
| 7.1       | Installer . . . . .  | 157        |
| 7.2       | Ajouter un utilisateur . . . . .                                     | 157        |
| 7.3       | Changer la configuration . . . . .                                   | 158        |
| 7.4       | Autoriser la table des fichiers joints à dépasser les 4 Go . . . . . | 158        |
| <b>8</b>  | <b>PostgreSQL</b>  | <b>161</b> |
| 8.1       | Ajouter un utilisateur . . . . .                                     | 161        |
| 8.2       | Autorisations d'accès . . . . .                                      | 161        |
| <b>9</b>  | <b>Oracle</b>  | <b>163</b> |
| 9.1       | Créer un nouveau tablespace . . . . .                                | 163        |
| 9.2       | Ajouter un utilisateur Oracle . . . . .                              | 163        |
| 9.3       | Configurer le serveur Web . . . . .                                  | 164        |
| <b>10</b> | <b>SQLite</b>  | <b>165</b> |
| <b>11</b> | <b>Apache</b>  | <b>167</b> |
| 11.1      | Sécuriser Apache . . . . .   | 167        |
| 11.2      | Apache with mod_cgi . . . . .  | 167        |
| 11.3      | Apache with mod_perl . . . . .                                       | 168        |
| <b>12</b> | <b>Apache</b>  | <b>171</b> |
| 12.1      | Installation . . . . .   | 171        |
| 12.2      | Droits d'accès Apache . . . . .                                      | 171        |
| 12.3      | Port et DocumentRoot . . . . .                                       | 172        |
| 12.4      | Activer la gestion de CGI . . . . .                                  | 172        |
| 12.5      | Indiquer l'emplacement de Bugzilla . . . . .                         | 172        |
| 12.6      | Fichiers journaux . . . . .  | 172        |
| 12.7      | Redémarrer Apache . . . . .  | 173        |
| <b>13</b> | <b>Microsoft IIS</b>   | <b>175</b> |
| 13.1      | Créer une nouvelle Application . . . . .                             | 175        |

|      |  |     |
|------|--|-----|
| 13.2 | Configurer le document par défaut . . . . .  | 175 |
| 13.3 | Ajouter un mappage de gestionnaire . . . . . | 175 |
| 13.4 | Application Bugzilla . . . . .               | 176 |
| 13.5 | Problèmes courants . . . . .                 | 176 |



---

## À propos de ce guide

---

### 1.1 Introduction

Ceci est la documentation de la version 5.0 de Bugzilla, un système de suivi de bogues de mozilla.org. Bugzilla est un logiciel professionnel qui permet de suivre des millions de bogues et tickets pour des centaines d'organisations dans le monde.

La version la plus à jour de ce document se trouve sur la [Page de documentation de Bugzilla](#).

### 1.2 Évaluer Bugzilla

Si vous voulez essayer Bugzilla pour voir s'il vous convient, vous pouvez le faire sur [Landfill](#), notre serveur de test. La [FAQ de Bugzilla](#) peut aussi être utile, car elle répond à bon nombre de questions posées par les gens pour savoir si Bugzilla est ce qu'ils cherchent.

### 1.3 Obtenir plus d'aide

Si ce document ne répond pas à vos questions, nous maintenons un [Forum Mozilla](#) qui peut être accédé par un groupe de nouvelles, une liste de diffusion ou sur le Web par Google Group. Veuillez d'abord [effectuer une recherche sur celui-ci](#) puis y poser votre question si vous n'avez pas trouvé de réponse.

Si vous souhaitez obtenir une réponse garantie, un support commercial est [disponible](#) pour Bugzilla par nombre de personnes ou d'organisations.

### 1.4 Conventions du document

Ce document utilise les conventions suivantes :

|   |
|---|
| <b>Avertissement :</b> Ceci est un avertissement, auquel vous devriez prêter attention. |
|---|

---

**Note :** Ceci est une note, pour votre information.

---

Un nom de fichier ou un chemin de nom de fichier est affiché comme ceci :  
`/chemin/vers/nom_de_fichier.ext`

---

Une commande à saisir dans un shell est affichée comme ceci : **commande –arguments**

Un échantillon de code est affiché comme ceci :

```
Première ligne de code  
Deuxième ligne de code  
...
```

Cette documentation est maintenue en format [reStructured Text](#) utilisant le système de documentation [Sphinx](#). Elle a été réécrite récemment et contient sûrement des bogues. Si vous en découvrez, merci de les rapporter dans le composant [Bugzilla Documentation](#) de l'installation Bugzilla de Mozilla. Si vous voulez faire une correction, ce serait très bien. Les modifications sont plus claires lorsqu'elles sont soumises sous forme de diff de fichier texte, joint à un bogue. Il existe un *Guide de style* pour vous aider.

## 1.5 Licence

Bugzilla est un logiciel [libre](#) et [open source](#), ce qui signifie (entre autres) que vous pouvez le télécharger, l'installer et l'utiliser à votre guise sans acquérir de licence d'utilisation ou payer. N'est-ce pas génial ?

Le code de Bugzilla est disponible sous licence [Mozilla Public License 2.0 \(MPL\)](#), et spécifiquement la variante qui est incompatible avec les licences secondaires. Cependant, si vous souhaitez seulement l'installer et l'utiliser, vous n'avez pas besoin de vous tracasser à ce sujet ; cela ne s'applique que si vous souhaitez redistribuer le code ou les modifications que vous y auriez apporté.

La documentation de Bugzilla est disponible sous la licence [Creative Commons CC-BY-SA International License 4.0](#), ou toute version ultérieure.

## 1.6 Crédits

Les personnes listées ci-dessous ont apporté des contributions significatives à la création de cette documentation :

Andrew Pearson, Ben FrantzDale, Byron Jones, Dave Lawrence, Dave Miller, Dawn Endico, Eric Hanson, Gervase Markham, Jacob Steenhagen, Joe Robins, Kevin Brannen, Martin Wulffeld, Matthew P. Barnson, Ron Teitelbaum, Shane Travis, Spencer Smith, Tara Hernandez, Terry Weissman, Vlad Dascalu, Zach Lipton.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).



---

## Guide utilisateur

---

### 2.1 Créer un compte

Si vous voulez utiliser Bugzilla, vous devez commencer par créer un compte. Consultez l'administrateur responsable de votre installation Bugzilla pour connaître l'URL à laquelle vous connecter pour y accéder. Si vous désirez tester, utilisez une installation de [Landfill](#).

Le processus de création de compte est similaire à ala plupart des autres sites Web.

1. Sur la page d'accueil, cliquez sur le lien *Nouveau compte* situé en haut ou en bas de la page. Saisissez maintenant votre adresse électronique puis cliquez sur le bouton *Envoyer*.

---

**Note :** Si le lien *Nouveau compte* n'est pas disponible, cela signifie que l'administrateur de cette installation a désactivé la création automatique de compte. Lui seul peut créer des comptes utilisateurs. Une raison possible est que cette installation est privée.

---

2. Dans quelques instants, et si votre enregistrement est accepté, vous devriez recevoir un courriel sur l'adresse électronique que vous avez fournie, qui contient votre nom de connexion (généralement identique à l'adresse électronique), et deux URL avec un jeton (une chaîne aléatoire générée par l'installation) pour confirmer ou annuler votre enregistrement. C'est un moyen pour empêcher les utilisateurs d'abuser avec la création de comptes, par exemple en saisissant des adresses électroniques inexistantes ou qui ne sont pas les leurs.
3. Si vous confirmez votre inscription, Bugzilla vous demandera votre vrai nom (optionnel, mais recommandé) et votre mot de passe. Suivant la configuration, un minimum de complexité peut être requis pour votre mot de passe.
4. Maintenant, il vous suffit de cliquer sur le lien *Connexion* au bas de la page dans votre navigateur, de saisir l'adresse électronique et le mot de passe que vous avez choisis dans le formulaire de connexion et de cliquer sur le bouton *Se connecter*.

Vous êtes maintenant connecté. Bugzilla utilise des cookies pour se souvenir que vous vous êtes connecté, donc, à moins d'avoir désactivé les cookies ou d'avoir changé d'adresse IP, vous ne devriez pas avoir à vous identifier à nouveau pendant votre session.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 2.2 Rapporter des bogues

### 2.2.1 Rapporter un nouveau bogue

Des années d'expérience dans l'écriture de bogues ont été distillées pour votre satisfaction dans les [recommandations d'écriture de bogue](#). Bien que certains des conseils soient spécifiques à Mozilla, les principes de base (reproductibilité, spécificité, isolation du produit utilisé, version du produit, le composant incriminé, la plateforme matérielle, et le système d'exploitation utilisé au moment du bogue) assurent des correctifs appropriés pour le bogue que vous avez rencontré.

---

**Note :** Si vous voulez rapporter un bogue pour voir comment Bugzilla fonctionne, vous pouvez le faire sur une de nos installations de test sur [Landfill](#). Merci de ne pas le faire sur l'installation Bugzilla de production.

---

La procédure pour ouvrir un bogue est la suivante :

1. Cliquez sur le lien *Rapporter* disponible dans le pied des pages ou sur le lien *Rapporter un nouveau bogue* affiché sur la page d'accueil de l'installation de Bugzilla.
2. Vous devez d'abord sélectionner le produit dans lequel vous avez trouvé un bogue.
3. Vous voyez maintenant un formulaire où vous pouvez indiquer le composant (la partie du produit affectée par le bogue que vous avez découvert ; si vous n'en avez aucune idée, sélectionnez *Général* si un tel composant existe), la version du programme que vous utilisez, le système d'exploitation et la plateforme sur laquelle vous exécutez le programme, et la gravité du bogue (si le bogue que vous avez trouvé plante le programme, c'est probablement un bogue *major* ou *critical* ; s'il s'agit d'une coquille quelque part, c'est un bogue *minor* ; s'il s'agit de quelque chose que vous voudriez voir mis en œuvre, choisissez *enhancement*).
4. Vous devez maintenant fournir un résumé court mais descriptif du bogue que vous avez trouvé. `Mon programme plante tout le temps` est un résumé très pauvre et n'aide pas du tout les développeurs. Essayez quelque chose de plus significatif ou votre bogue sera probablement ignoré à cause du manque de précision. L'étape suivante est de donner une liste très détaillée des étapes pour reproduire le problème que vous avez rencontré. Essayez de limiter ces étapes au minimum nécessaire pour reproduire le problème. Cela facilitera la vie aux développeurs et la probabilité qu'ils traitent votre bogue dans un délai raisonnable sera beaucoup plus grande.

---

**Note :** Assurez-vous que tout ce qui se trouve dans le résumé figure aussi dans le premier commentaire. Les résumés sont souvent mis à jour et cela assurera que les informations d'origine soient facilement accessibles.

---

5. Lorsque vous rapportez le bogue, vous pouvez également joindre un document (un jeu de test, un correctif, ou une copie d'écran du problème).
6. Suivant l'installation de Bugzilla que vous utilisez et le produit pour lequel vous rapportez le bogue, vous pouvez aussi demander aux développeurs de considérer votre bogue de différentes manières (comme demander la revue du correctif que vous venez de joindre, demander que votre bogue bloque la prochaine version du produit et beaucoup d'autres requêtes spécifiques au produit).
7. C'est maintenant le bon moment pour relire votre rapport de bogue. Retirer toutes les fautes d'orthographe, sans quoi votre bogue pourrait ne pas être trouvé par les développeurs qui exécutent des requêtes sur des mots spécifiques, et par conséquent, votre bogue ne serait pas porté à leur attention. Assurez-vous aussi de n'avoir pas oublié d'informations importantes que les développeurs devraient savoir pour reproduire le problème et que la description du problème est suffisamment claire et explicite. Si vous pensez que votre rapport de bogue est prêt, la dernière étape est de cliquer sur le bouton *Soumettre* pour ajouter votre rapport à la base de données.

### 2.2.2 Clonage d'un bogue existant

Bugzilla a une fonctionnalité qui vous permet de cloner un bogue existant. Le bogue nouvellement créé héritera de la plupart des paramètres de l'ancien bogue. Ceci vous permet de reporter plus facilement dans un nouveau bogue des

problèmes similaires. Pour utiliser cette fonctionnalité, rendez-vous dans le bogue que vous voulez cloner, puis cliquer sur le lien *Cloner ce bogue* sur la page du bogue. Ceci vous amènera à la page *Rapporter un bogue* qui sera pré-remplie avec les valeurs de l'ancien bogue. Vous pouvez changer ces valeurs et/ou en ajouter d'autres si nécessaire.

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 2.3 Anatomie d'un bogue

Le cœur de Bugzilla se situe dans l'écran affiché pour un bogue en particulier. Vous remarquerez que les libellés pour la plupart des champs sont des hyperliens ; en cliquant dessus, vous ferez apparaître une aide contextuelle sur ce champ en particulier. Les champs marqués avec \* peuvent ne pas être présents sur toutes les installations Bugzilla.

**Résumé :** Une phrase résumant le problème, affichée dans l'en-tête à côté du numéro de bogue.

**État (et Résolution) :** Ceci définit l'état exact dans lequel se trouve le bogue - de l'état non confirmé comme bogue à l'état corrigé et confirmé par l'assurance qualité. Les différentes valeurs possibles pour l'état et la résolution sont documentées dans l'aide contextuelle pour ces éléments.

**Alias :** Un texte très court et unique pour le bogue, qui peut être utilisé pour le rechercher à la place de son numéro.

**Produit et composant :** Les bogues sont divisés en produits et composants, un produit ayant un ou plusieurs composants.

**Version :** Le champ `Version` est habituellement utilisé pour les versions d'un produit qui a été publié et est défini pour indiquer quelles versions d'un composant a le problème particulier exposé dans le bogue.

**Matériel (Plateforme et OS) :** Ceci indique l'environnement dans lequel le bogue a été trouvé.

**Importance (Priorité et Gravité) :** Le champ `Priorité` est utilisé pour classer les bogues en fonction de leur priorité, soit par le responsable soit par une autre personne ayant autorité pour gérer son temps, comme le chef de projet par exemple. Il est de bon ton de ne pas changer ceci sur les bogues d'autres personnes. Les valeurs par défaut vont de P1 à P5.

Ceci indique la gravité du problème - de `blocker` (l'application est inutilisable) à `trivial` (problème cosmétique mineur). Vous pouvez aussi utiliser ce champ pour indiquer si le bogue est une demande d'amélioration.

**\*Jalon cible :** Une version future pour laquelle le bogue devra être corrigé. Par exemple, les jalons du projet Bugzilla pour les futures versions de Bugzilla sont 4.4, 5.0, 6.0, etc. Les jalons ne sont pas limités à des nombres cependant : vous pouvez utiliser toute chaîne de texte, comme des dates par exemple.

**Assigné à :** La personne responsable de la correction du bogue.

**\*Contact QA :** La personne responsable de l'assurance qualité du bogue.

**URL :** Une URL associée au bogue, le cas échéant.

**\*Tableau blanc :** Un champ de saisie de texte libre pour ajouter de courtes notes ou des marqueurs à un bogue.

**Mots-clés :** L'administrateur peut définir des mots-clés que vous pouvez utiliser pour marquer et catégoriser les bogues – par exemple, le Projet Mozilla a des mots-clés comme `plantage` et `régression`.

**Mots-clés personnels :** Au contraire des `Mots-clés` qui sont globaux et visibles par tous, les `Mots-clés personnels` ne sont vus et modifiables que par leur auteur. Les modifier n'enverra pas de courriels aux autres utilisateurs. Utilisez-les pour suivre les bogues qui vous intéressent en utilisant votre propre système de classification.

**Dépendances (Dépend de et Bloque) :** Si ce bogue ne peut être corrigé avant que d'autres bogues soient corrigés (Dépend de), ou si ce bogue empêche d'autres bogues d'être corrigés (Bloque). Leurs numéros sont indiqués ici.

Cliquer sur le lien *Afficher l'arbre / le graphique de dépendance* affiche les relations de dépendance du bogue sous forme d'arborescence. Vous pouvez modifier la profondeur de l'arbre et masquer les bogues résolus à partir de cette page. Vous pouvez réduire/développer les dépendances pour chaque bogue non terminal, en utilisant les boutons `[-]/[+]` qui apparaissent avant le résumé.

**Rapporteur** : La personne qui a rapporté le bogue.

**Modifié le** : Date et heure de la dernière modification du bogue.

**Copie à** : Une liste de gens qui reçoivent des courriels quand des changements surviennent sur le bogue.

**Ignorer les courriels de bogue** : Sélectionner ceci pour ne jamais plus recevoir de courriel au sujet de ce bogue.  
Voir aussi *Préférences de messagerie*.

**\*Voir aussi** : Bogues dans cette installation de Bugzilla, d'autres installations de Bugzilla ou d'autres logiciels de suivi de bogue, relatifs à ce bogue.

**Étiquettes** : Une étiquette indique un état qui peut être affecté à des bogues ou des fichiers joints pour indiquer que les bogues ou les fichiers joints sont dans un état particulier. Chaque installation peut définir son propre jeu d'étiquettes. Voir *Étiquettes*.

**\*Horodatage** : Ce formulaire peut être utilisé pour l'horodatage. Pour utiliser cette fonctionnalité, vous devez avoir l'appartenance de groupe spécifié par le paramètre `timetrackinggroup`. Voir *Informations d'horodatage* pour plus d'informations.

**Est. initiale** Ce champ affiche l'estimation initiale du temps escompté.

**Est. actuelle** Ce champ affiche le temps estimé actuel. Ce nombre est calculé à partir des chiffres de `Heures travaillées` et `Heures restantes`.

**Heures travaillées** Ce champ affiche le nombre d'heures travaillées.

**Heures restantes** Ce champ affiche le résultat de la soustraction `Est. actuelle - Heures travaillées`. Cette valeur + `Heures travaillées` deviendra la nouvelle `Est. actuelle`.

**% d'achèvement** Ce champ affiche le pourcentage d'accomplissement de la tâche.

**Gain** Ce champ affiche le nombre d'heures d'avance par rapport à l'estimation d'origine.

**Échéance** Ce champ affiche la date d'échéance pour le bogue.

**Fichiers joints** : Vous pouvez joindre des fichiers (par exemple des jeux de test ou des correctifs) aux bogues. S'il y a des fichiers joints, ils sont listés dans cette section. Voir *Fichiers joints* pour plus d'informations.

**Commentaires additionnels** : Vous pouvez ajouter votre grain de sel à la discussion sur le bogue ici, si vous avez quelque chose à ajouter qui vaut la peine d'être mentionné.

### 2.3.1 Étiquettes

Les étiquettes sont un moyen d'indiquer un état particulier pour un bogue ou un fichier joint, soit + ou -. La signification de ces valeurs sont spécifiques à l'étiquette et ne peuvent par conséquent être décrites dans cette documentation. À titre d'exemple, définir une étiquette intitulée `revue` à + peut indiquer que le bogue ou le fichier joint a été approuvé, alors que - peut indiquer que le bogue ou le fichier joint a été refusé.

Une étiquette apparaît dans les rapports de bogues et sur les pages de détails des fichiers joints avec le nom de l'utilisateur qui a défini l'étiquette, abrégé. Par exemple, si Jack définit l'étiquette `revue` à +, il apparaîtra comme `Jack : revue [ + ]`.

Une étiquette qui a été demandée à un autre utilisateur apparaîtra avec le nom de l'utilisateur qui l'a demandée devant le libellé de l'étiquette et le nom de l'utilisateur à qui a été demandée l'étiquette apparaîtra après le libellé de l'étiquette, entre parenthèses. Par exemple, si Jack demande à Jill une revue, cela s'affichera ainsi : `Jack : review [ ? ] (Jill)`.

Vous pouvez naviguer dans les requêtes en attente qui vous sont demandées ou que vous avez demandées en sélectionnant *Mes requêtes* dans le pied de page. Vous pouvez aussi y voir celles demandées par d'autres ou à d'autres, par produit, composant et nom d'étiquette dans cette page. Notez que vous pouvez utiliser - pour spécifier des étiquettes n'ayant pas de nom d'utilisateur dans le champ *Demandé à*.

## Un exemple simple

Un développeur pourrait demander à son manager si un bogue doit être corrigé avant la version 2.0. Ils pourraient avoir à faire ce choix pour *beaucoup* de bogues, et décident donc de formaliser le processus. Donc :

1. L'administrateur de Bugzilla crée une étiquette `bloque2.0` pour les bogues du produit. Cette étiquette s'affiche dans l'écran d'affichage de bogue avec le texte *bloque2.0* suivi d'une liste déroulante. La liste déroulante contient quatre valeurs : une espace, ?, - et +.
2. Le développeur définit l'étiquette à ?.
3. Le manager voit l'étiquette *bloque2.0* avec la valeur ?.
4. Si le manager pense que la fonctionnalité doit être intégrée dans le produit avant que la version 2.0 soit publiée, il définit l'étiquette à +. Sinon, il choisit -.
5. Dès lors, chaque utilisateur de Bugzilla qui consulte le bogue saura si le bogue sera corrigé ou pas avant la sortie de la version 2.0.

## À propos des étiquettes

Les étiquettes peuvent prendre quatre valeurs :

- ? Un utilisateur fait une demande sur l'état. (Penser à cela comme une question qui est posée).
- L'état a été défini négativement. (La réponse à la question est non).
- + L'état a été défini positivement. (La réponse à la question est oui).
- \_ non défini s'affiche sous la forme d'une espace. Cela signifie que personne n'a exprimé d'opinion (ou personne n'a demandé à un autre utilisateur son avis) sur le sujet couvert par l'étiquette.

## Demandes d'étiquettes

Si une étiquette a été définie comme *demandé*, et qu'un utilisateur a assez de droits pour la demander (voir ci-dessous), l'utilisateur peut définir l'état de l'étiquette à ?. Cet état indique que quelqu'un (c'est-à-dire « le demandeur ») demande à quelqu'un d'autre de définir l'étiquette soit à + soit à -.

Si une étiquette est définie comme *sollicité*, une boîte de texte apparaît à côté de l'étiquette dans laquelle le demandeur peut saisir le nom d'un utilisateur Bugzilla. Cette personne (la personne « sollicitée ») recevra un courriel l'informant de la demande et pointerà vers le bogue ou le fichier joint en question.

Si une étiquette n'a *pas* été définie comme *sollicité* par l'administrateur de Bugzilla, alors aucune boîte de texte n'apparaîtra à côté de l'étiquette. Une demande pour définir cette étiquette ne peut être demandée spécifiquement à un utilisateur ; ces demandes sont ouvertes à tout le monde. Dans Bugzilla, cela s'appelle « demander au vent » (asking the wind). Un utilisateur peut « demander au vent » pour toute étiquette en laissant la boîte de texte vide.

## Étiquettes de fichiers joints

Il existe deux types d'étiquettes : les étiquettes de bogues et les étiquettes de fichiers joints.

Les étiquettes de fichiers joints sont utilisées pour poser une question sur un fichier joint spécifique d'un bogue.

Beaucoup d'installations Bugzilla utilisent ceci pour demander à un développeur de faire la revue de code d'un autre développeur avant de déposer le correctif dans le dépôt. Ils attachent le code à un rapport de bogue et définissent une étiquette *revue* à *revue ? reviewer@example.com*. `reviewer@example.com` est alors notifié par courriel qu'il doit faire la revue d'un fichier joint et l'approuver ou le rejeter.

Pour un utilisateur de Bugzilla, les étiquettes de fichier joint se voit en trois endroits :

1. Sur la liste des fichiers joints sur la page d'un bogue, vous pouvez voir l'état courant de toute étiquette ayant été définie à ?, + ou -, l'utilisateur l'ayant demandée et celui à qui elle a été demandée.

2. Lors de l'édition d'un bogue, vous pouvez voir les étiquettes qui peuvent être définies et celles qui l'ont déjà été. L'écran *Détails* d'un fichier joint permet de définir les étiquettes à ?, -, + ou les remettre à blanc.
3. Les demandes sont listées dans la page *File d'attente des requêtes*, qui est accessible à partir du lien *Mes requêtes* (si vous êtes identifié) ou le lien *Requêtes* (si vous n'êtes pas identifié), visible sur toutes les pages.

### Étiquettes de bogue

Les étiquettes de bogue sont utilisées pour définir un état sur le bogue lui-même. Vous pouvez voir les étiquettes de bogue dans les écrans d'édition d'un bogue et des requêtes comme indiqué ci-dessus.

Seuls les utilisateurs bénéficiant de suffisamment de privilèges (voir plus bas) peuvent définir des étiquettes sur les bogues. Cela n'inclut pas nécessairement le responsable, le rapporteur ou les utilisateurs ayant la permission editbugs.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 2.4 Éditer un bogue

### 2.4.1 Fichiers joints

Les fichiers joints sont utilisés pour attacher des fichiers relatifs à un bogue : correctifs, copies d'écran, fichiers logs, ou tout autre fichier binaire ou trop gros pour la zone de commentaire.

Vous devez utiliser des fichiers à joindre plutôt que des commentaires pour les gros volumes de texte ASCII, comme les traces, les fichiers de débogage ou les fichiers journaux. Ainsi, cela ne remplit pas inutilement le bogue pour celui qui voudrait le lire et cela évite la réception de gros courriels inutiles pour les personnes qui suivent le bogue.

Vous devez ajuster les copies d'écrans. Il n'est pas nécessaire d'afficher tout l'écran pour signaler un problème sur un pixel.

Bugzilla stocke et utilise un type de contenu (content-type) pour chaque fichier joint (par ex. : text/html). Pour télécharger un fichier joint avec un type de contenu différent (par ex. : application/xhtml+xml), vous pouvez outrepasser ceci en ajoutant un paramètre `content_type` à l'URL, par ex. : `&content_type=text/plain`.

Également, vous pouvez saisir l'URL qui pointe vers le fichier plutôt que de le télécharger vers le serveur. Par exemple, ceci est utile si vous voulez pointer vers une application externe, un site Web ou un très gros fichier. Notez qu'il n'y a pas de garantie que le fichier source soit toujours disponible ni que son contenu reste inchangé.

Un autre moyen de joindre des données est de coller du texte directement dans le champ texte, et Bugzilla le convertira en fichier joint. C'est très pratique quand vous faites du copier-coller et que vous ne voulez pas mettre le texte dans un fichier temporaire d'abord.

### 2.4.2 Étiquettes

Pour définir une étiquette, sélectionnez soit +, soit - dans le menu déroulant à côté du nom de l'étiquette dans la liste *Étiquettes*. La signification de ces valeurs sont spécifiques à l'étiquette et ne peuvent par conséquent être décrites dans cette documentation. À titre d'exemple, définir une étiquette intitulée *revue* à + peut indiquer que le bogue ou le fichier joint a été approuvé, alors que - peut indiquer que le bogue ou le fichier joint a été refusé.

Pour enlever une étiquette, cliquez sur le menu déroulant et sélectionnez la valeur vide. Notez que marquer un fichier joint comme obsolète annule automatiquement toutes les requêtes en attente pour le fichier joint.

Si votre administrateur a activé les requêtes d'étiquettes, demandez une étiquette en sélectionnant ? dans le menu déroulant et saisissez le nom de l'utilisateur à qui vous demandez l'étiquette dans le champ de texte à côté du menu.

### 2.4.3 Informations d'horodatage

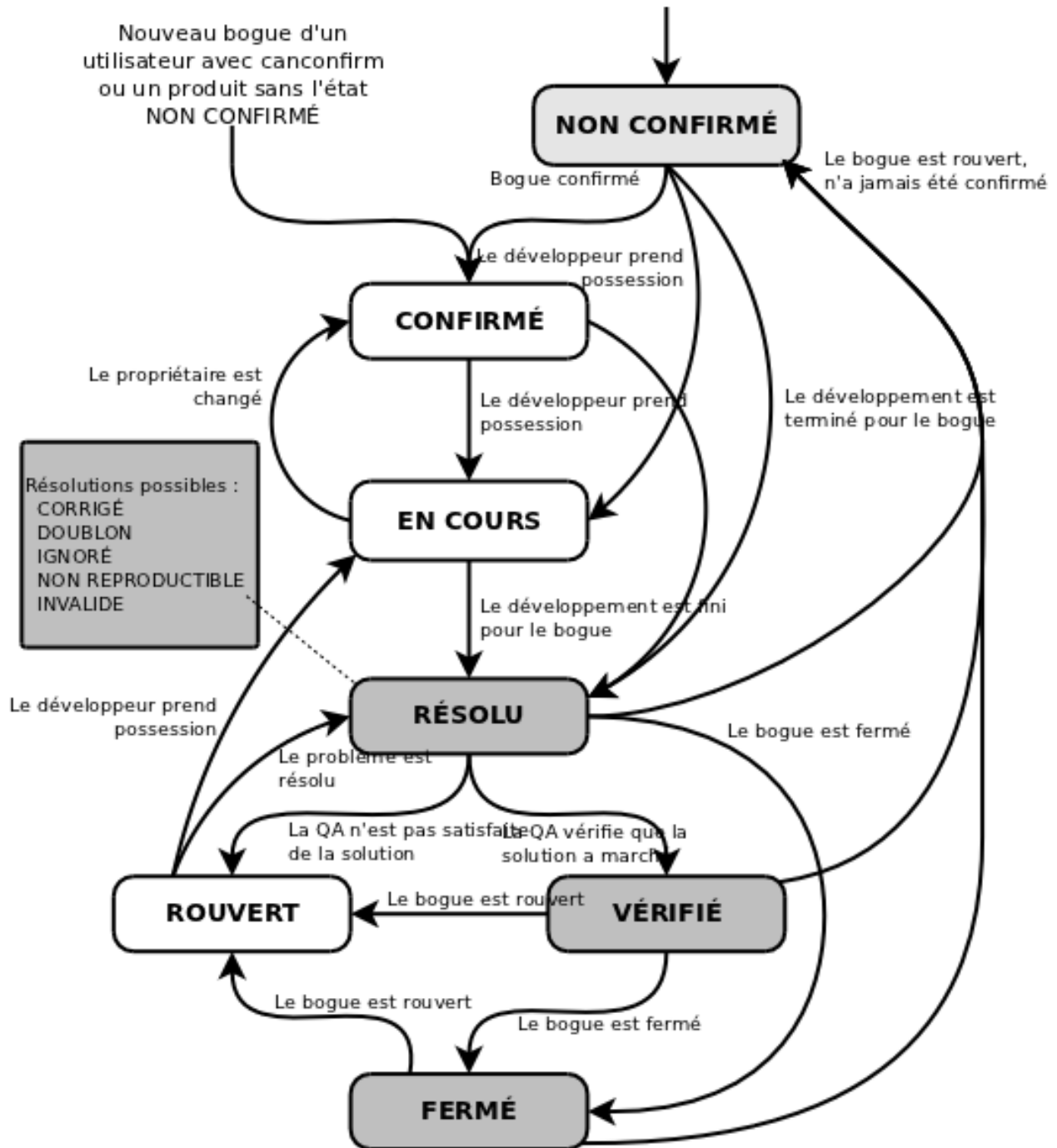
Les utilisateurs qui appartiennent au groupe spécifié par le paramètre `timetrackinggroup` ont accès aux champs relatifs au temps. Les développeurs peuvent voir les dates limites et les estimations de temps pour corriger les bogues, et peuvent fournir le temps passé sur ces bogues. Les utilisateurs qui n'appartiennent pas à ce groupe peuvent seulement voir l'échéance, mais pas la modifier. Les autres champs relatifs au temps leur restent invisibles.

À tout moment, un résumé du temps passé par les développeurs sur les bogues est accessible à partir des listes de bogues en cliquant sur le bouton `Résumé du temps passé` ou dans chaque bogue individuellement en cliquant sur le lien `Résumé du temps passé` dans le tableau d'horodatage. La page `summarize_time.cgi` vous permet de voir ces informations soit par développeur, soit par bogue, et peut être divisé par mois pour avoir plus de détails sur le temps passé par les développeurs.

Dès qu'un bogue est marqué `RÉSOLU`, le temps restant estimé pour corriger le bogue est défini à zéro. Ceci permet aux personnes de l'assurance qualité de le définir à nouveau pour leur propre usage, et il sera de nouveau défini à zéro quand le bogue sera marqué `FERMÉ`.

### 2.4.4 Cycle de vie d'un bogue

Le cycle de vie d'un bogue est personnalisable pour s'adapter aux besoins de votre organisation, consulter [Workflow des états de bogue](#). L'image ci-dessous contient une représentation graphique de ce cycle de vie, utilisant les états de bogue par défaut. Si vous voulez personnaliser cette image pour votre site, le fichier du diagramme est disponible dans le format XML natif de [Dia](#).



Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 2.5 Rechercher des bogues

Bugzilla dispose de différentes options de recherche.

**Note :** Les recherches dans Bugzilla sont insensibles à la casse et aux accents, avec l'utilisation de systèmes de gestion



---

de bases de données MySQL ou Oracle. Cependant, lorsque Bugzilla est utilisé avec PostgreSQL, certaines recherches sont sensibles à la casse. Ceci est dû à la façon dont PostgreSQL traite la sensibilité à la casse et aux accents.

---

### 2.5.1 Recherche rapide

La recherche rapide est un outil de recherche sous la forme d'une simple boîte de texte pouvant utiliser les méta-caractères pour indiquer ce que l'on cherche. Par exemple, en saisissant `toto|titi` dans la boîte de texte, cela effectuera une recherche de `toto` ou `titi` dans les champs Résumé et Tableau blanc d'un bogue ; en ajoutant `:ProduitX`, la recherche s'effectuera seulement dans ce produit.

Vous pouvez l'utiliser pour trouver un bogue par son numéro ou son alias aussi.

### 2.5.2 Recherche simple

La recherche simple est utile pour trouver un bogue en particulier. Cela fonctionne comme un moteur de recherche Internet : saisissez quelques mots-clés et c'est parti.

### 2.5.3 Recherche avancée

La page de recherche avancée est utilisée pour produire une liste de bogues correspondant à des critères précis. [Vous pouvez l'essayer sur Landfill](#).

La page de recherche a des contrôles pour sélectionner différentes valeurs pour tous les champs disponibles d'un bogue, comme décrit plus haut. Pour certains champs, plusieurs valeurs peuvent être sélectionnées. Dans ce cas, Bugzilla affiche les bogues dont le contenu du champ correspond à au moins une des valeurs sélectionnées. Si aucune n'est sélectionnée, alors le champ peut prendre n'importe quelle valeur.

Après avoir lancé une recherche, vous pouvez l'enregistrer en tant que Recherche enregistrée, qui apparaîtra alors dans le pied de page. Si vous êtes dans le groupe défini par le paramètre `querysharegroup`, vous pouvez partager vos recherches avec d'autres utilisateurs, consulter [Recherches enregistrées](#) pour plus de détails.

### 2.5.4 Recherche personnalisée

Les recherches élaborées sont faites en utilisant la fonctionnalité *Recherche avancée* dans la page *Recherche avancée*.

Les critères de recherche ici affinent encore plus l'ensemble de résultats renvoyés par une requête. Il est possible de rechercher des bogues avec des combinaisons élaborées de critères.

La plus simple des recherches booléennes n'a qu'un seul terme. Ces recherches permettent au *champ* sélectionné à gauche d'être comparé en utilisant un *opérateur* ayant une *valeur* spécifique. En utilisant les boutons *Et*, *Ou* et *Ajouter une nouvelle ligne*, des termes supplémentaires peuvent être ajoutés à la requête, affinant ainsi encore la liste des bogues renvoyés par la requête.

Il y a trois champs par colonne pour une recherche personnalisée :

- *Champ* : les éléments recherchés
- *Opérateur* : l'opérateur de comparaison
- *Valeur* : la valeur à laquelle le champ est comparé

La liste des *champs* disponibles contient tous les champs définis pour un bogue, y compris les champs personnalisés et aussi quelques pseudo-champs comme *Nom réel*, *Jours écoulés depuis le dernier changement*, *Temps d'inactivité du responsable* et d'autres qui peuvent être utiles.

Il y a un large panel d'*opérateurs* disponibles, tous n'ayant pas forcément de sens suivant le champ. Il existe diverses opérations de correspondance de chaîne de texte (y compris les expressions régulières), des comparaisons numériques

(qui fonctionnent aussi pour les dates) et la possibilité de rechercher un changement d'information—quand un champ a été changé, ce qui a changé et qui l'a fait. Il existe aussi des opérateurs spéciaux comme *est vide* et *n'est pas vide*, car Bugzilla ne peut pas faire la différence entre une valeur de champ laissée vide intentionnellement ou par accident.

Vous pouvez avoir un nombre de lignes arbitraire, et la liste déroulante pour chacune d'elles définit comment elles sont liées—*Correspond à TOUS les critères qui suivent séparément*, *Correspond à AU MOINS UN des critères qui suivent* ou *Correspond à TOUS les critères qui suivent dans le même champ*. La différence entre la première et la troisième peut être illustrée par une recherche sur un commentaire. Si vous avez la recherche :

```
Commentaire    contient la chaîne    « Fred »
Commentaire    contient la chaîne    « Barney »
```

alors, selon la première (séparément) la recherche renverrait des bogues où « Fred » apparaîtrait dans un commentaire et « Barney » dans le même commentaire ou dans un autre, alors que pour la troisième (dans le même champ), les deux chaînes devraient apparaître dans le même commentaire.

### Fonctionnalités avancées

Si vous cliquez sur le lien *Afficher les fonctionnalités avancées*, d'autres possibilités apparaîtront. Vous pouvez rejeter (PAS) toute ligne en cochant une case (voir ci-dessous) ou un groupe de lignes entre parenthèses. Dans chaque ensemble de parenthèses, vous avez le choix de les combiner toutes en utilisant TOUS (c'est-à-dire ET) ou AU MOINS UN (c'est-à-dire OU).

### Négation

À première vue, la négation semble superflue. Plutôt que rechercher :

```
PAS (« Résumé » « contient la chaîne » « toto »),
```

on peut rechercher :

```
(« Résumé » « ne contient pas la chaîne » « toto »).
```

Cependant, la recherche :

```
(« Copie à » « ne contient pas la chaîne » « @mozilla.org »)
```

trouverait tout bogue pour lequel quiconque dans la liste Copie à ne contient pas @mozilla.org alors que :

```
PAS (« Copie à » « contient la chaîne » « @mozilla.org »)
```

trouverait tout bogue pour lequel il n'y a personne dans la liste Copie à qui contient la chaîne. De même, l'utilisation de la négation permet aussi la construction d'expressions complexes utilisant des termes séparés par Ou puis négativés. La négation permet des requêtes telles que :

```
PAS ((« Produit » « est égal à » « mise à jour ») OU
(« Composant » « est égal à » « Documentation »))
```

pour trouver des bogues qui ne sont ni dans le produit mise à jour, ni dans le composant Documentation ou :

```
PAS ((« Commentateur » « est égal à » « %assignee% ») OU
(« Composant » « est égal à » « Documentation »))
```

pour trouver des bogues qui ne sont pas liés à la documentation et pour lesquels le responsable n'a jamais fait de commentaires.

## Substitution de pronom

Quelquefois, une requête a besoin de comparer un champ relatif à l'utilisateur (tel que `ReportedBy`) avec un rôle utilisateur spécifique (comme l'utilisateur exécutant la requête ou l'utilisateur à qui chaque bogue est assigné).

Quand l'opérateur est *est égal à* ou *n'est pas égal à*, la valeur peut être `%reporter%`, `%assignee%`, `%qacontact%` ou `%user%`. Le pronom d'utilisateur se réfère à l'utilisateur qui exécute la requête, ou, dans le cas des rapports de notifications, l'utilisateur qui sera destinataire du rapport. Les pronoms `reporter`, `assignee` et `qacontact` se réfèrent aux champs correspondants dans le bogue.

Les tableaux booléens vous permettent aussi de saisir un nom de groupe dans tout champ relatif à un utilisateur si l'opérateur est *est égal à*, *n'est pas égal à* ou *contient une des chaînes*. Ceci vous permettra de faire des requêtes sur tout membre appartenant (ou pas) au groupe spécifié. Le nom du groupe doit être saisi en suivant la syntaxe `%group.toto%`, où `toto` est le nom du groupe. Donc, si vous recherchez des bogues rapportés par tout utilisateur appartenant au groupe `editbugs`, vous pouvez alors saisir :

```
rapporteur est égal à %group.editbugs%
```

### 2.5.5 Listes de bogues

Si vous lancez une recherche, une liste des bogues correspondants sera renvoyée.

Le format de cette liste est configurable. Par exemple, elle peut être triée en cliquant sur les en-têtes de colonnes. D'autres fonctionnalités utiles peuvent être accédées en utilisant les liens au bas de la liste :

**Format long :** Ceci donne une grande page avec les champs `Résumé` non modifiables de chaque bogue.

**XML (icône) :** Produit la liste de bogues au format XML.

**CSV (icône) :** Produit la liste de bogues comme des valeurs séparées par des virgules, pour l'importer par exemple dans un tableur.

**Flux (icône) :** Produit la liste de bogues sous forme de flux Atom. Copiez ce lien dans votre lecteur de flux préféré. Si vous utilisez Firefox, vous pouvez aussi enregistrer la liste sous forme de marque-page dynamique en cliquant sur l'icône de marque-page dynamique dans la barre d'adresse. Pour limiter le nombre de bogues dans le flux, ajouter un paramètre `limit=n` à l'URL.

**iCalendar (icône) :** Produit la liste sous forme de fichier iCalendar. Chaque bogue est représenté sous forme de tâche dans l'agenda importé.

**Changer les colonnes :** Modifie les attributs des bogues apparaissant dans la liste.

**Changer plusieurs bogues à la fois :** Si votre compte a suffisamment de droits et qu'il apparaît plus d'un bogue dans la liste, ce lien est affiché et vous permet d'apporter le même changement sur tous les bogues de la liste. Par exemple, changer leur responsable.

**Envoyer un courriel aux responsables des bogues :** Si plus d'un bogue apparaît dans la liste et qu'il y a au moins deux responsables distincts, ce lien est affiché pour permettre d'envoyer facilement un courriel à tous les responsables de la liste de bogues.

**Modifier la recherche :** Si vous n'obtenez pas exactement les résultats que vous escomptiez, vous pouvez revenir à la page de requête par ce lien et faire des ajustements sur la requête pour obtenir de meilleurs résultats.

**Enregistrer sous :** Vous pouvez donner un nom à la requête et l'enregistrer ; un lien apparaîtra dans votre pied de page et vous permettra d'y accéder et de l'exécuter rapidement plus tard.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 2.6 Rapports et graphiques

En plus de la liste de bogues standard, Bugzilla dispose deux autres moyens pour afficher un ensemble de bogues. Ce sont les rapports (qui donnent différentes vues de l'état actuel de la base de données) et les graphiques (qui tracent les changements dans des ensembles de bogues particuliers en fonction du temps).

### 2.6.1 Rapports

Un rapport est une vue de l'état actuel de la base de données.

Vous pouvez obtenir des rapports sous forme de tableaux HTML ou sous forme graphique avec des diagrammes linéaires, circulaires ou des histogrammes. Ils ont chacun une page distincte pour leur paramétrage, mais ce sont de proches cousins. Une fois que vous avez défini et affiché un rapport, vous pouvez basculer entre les différentes vues à volonté.

Les deux types de rapport sont basés sur l'idée d'une définition d'un ensemble de bogues en utilisant l'interface de recherche standard et en choisissant certains aspects de cet ensemble à tracer sur les axes horizontal et/ou vertical. Vous pouvez aussi obtenir une forme de rapport en trois dimensions en choisissant d'avoir de multiples images ou tableaux.

Vous pouvez par exemple utiliser le formulaire de recherche pour choisir `Tous les bogues du produit ContrôleDuMonde` et tracer leur gravité en fonction de leur composant pour voir quel composant a eu le plus grand nombre de bogues critiques rapportés.

Quand vous avez défini vos paramètres et que vous cliquez sur *Générer le rapport*, vous pouvez basculer entre les formats HTML, CSV, histogramme, linéaire et circulaire. (Note : les diagrammes circulaires ne sont disponibles que si vous n'avez pas défini d'axe vertical car les diagrammes circulaires n'en ont pas). Les autres contrôles s'expliquent d'eux-mêmes ; vous pouvez changer la taille de l'image si vous trouvez que les textes se superposent ou si les barres d'histogrammes sont trop fines.

### 2.6.2 Graphiques

Un graphique est une vue de l'état de la base de données en fonction du temps.

Bugzilla dispose actuellement de deux systèmes de graphiques. Les `Anciens graphiques` et les `Nouveaux graphiques`. Les anciens graphiques font partie de Bugzilla depuis longtemps ; ils tracent chaque état et résolution pour chaque produit et c'est tout. Ils sont obsolètes et vont bientôt disparaître. Nous n'en dirons pas plus à leur sujet. Les nouveaux graphiques sont le futur. Ils vous permettent de tracer tout ce que vous pouvez définir comme recherche.

---

**Note :** Les deux types de rapports nécessitent que l'administrateur définisse le script de collection de données. Si vous ne pouvez pas voir les graphiques, demandez-lui s'il a exécuté le script.

---

Une ligne individuelle sur un rapport est appelé collection. Toutes les collections sont organisées en *Catégories* et *Sous-catégories*. Les collections que définit automatiquement Bugzilla utilisent le nom de produit comme catégorie et les noms de composants comme sous-catégories, mais il n'est pas obligatoire pour vous de suivre cette norme de nommage avec vos rapports si vous ne le voulez pas.

Les collections peuvent être publiques ou privées. Tout le monde peut voir les collections publiques dans la liste, mais les collections privées ne peuvent être vues que par leurs créateurs. Seuls les administrateurs peuvent rendre les collections publiques. Il ne peut y avoir deux collections, même privées, ayant le même ensemble de catégorie, sous-catégorie et nom. Donc, si vous créez des collections privées, une idée est d'avoir la catégorie qui s'intitule comme votre nom d'utilisateur.

## Créer des graphiques

Vous créez un graphique en sélectionnant un nombre de collections dans la liste et en cliquant sur le bouton *Ajouter à la liste* pour chacune. Dans la liste des collections à tracer, vous pouvez définir le libellé que portera la collection dans la légende du graphique, et demander aussi à Bugzilla d'additionner un certain nombre de collections (par exemple, vous pouvez additionner les collections représentant les bogues *RÉSOLU*, *VÉRIFIÉ* et *FERMÉ* dans un produit en particulier pour obtenir une collection représentant tous les bogues résolus dans un produit).

Si vous avez ajouté par erreur une collection à la liste, sélectionnez-la en cochant la case et cliquez sur *Supprimer*. Quand vous avez ajouté plus d'une collection, une ligne *Grand total* apparaît automatiquement en bas de la liste. Si vous ne la voulez pas, supprimez-la comme vous supprimeriez toute autre ligne.

Vous pouvez aussi choisir de créer un graphique sur une certaine période et de cumuler les résultats, c'est-à-dire de tracer chacun en utilisant le précédent comme base, de sorte que la première ligne donne la somme de toutes les collections. Il est plus facile d'essayer que de l'expliquer :-).

Quand une collection est dans la liste, on peut réaliser certaines actions dessus. Par exemple, éditer les paramètres de la collection (nom, fréquence, etc.) si vous l'avez créée ou si vous êtes administrateur.

Quand vous êtes satisfait, cliquez sur *Générez ce rapport* pour voir le graphique.

## Créer de nouvelles collections

Vous pouvez aussi créer vos propres collections. Pour faire cela, cliquez sur le lien *Créer un nouveau jeu de données* sur la page de création de graphiques. Ceci vous amène sur une interface de recherche où vous pouvez définir la recherche que Bugzilla tracera. Au bas de la page, choisissez la catégorie, la sous-catégorie et le nom de votre nouvelle collection.

Si vous avez les droits suffisants, vous pouvez rendre cette collection publique et augmenter la fréquence de collecte des données à moins de sept jours, qui est la valeur par défaut.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 2.7 Trucs et astuces

Cette section fournit des astuces et les bonnes pratiques pour Bugzilla qui ont été développées.

### 2.7.1 Hyperlien automatique

Les commentaires dans Bugzilla sont en texte brut : saisir `<U>` produira `<, U, >` plutôt que du texte souligné. Cependant, Bugzilla fera automatiquement des hyperliens pour certaines parties du texte des commentaires. Par exemple, le texte `http://www.bugzilla.org` sera transformé en hyperlien : <http://www.bugzilla.org>. D'autres chaînes peuvent être transformées en lien :

- bogue 12345
- bogues 123, 456, 789
- commentaire 7
- commentaires 1, 2, 3, 4
- bogue 23456, commentaire 53
- attachment 4321
- <mailto:georges@exemple.com>
- [georges@exemple.com](mailto:georges@exemple.com)
- <ftp://ftp.mozilla.org>
- La plupart des autres types d'URL

Le corollaire ici, c'est que si vous saisissez un numéro de bogue dans un commentaire, vous devez mettre le mot `bugue` devant pour qu'il soit transformé en hyperlien pour la commodité des autres.

### 2.7.2 Commentaires

Si vous changez les champs d'un bogue, ne faites de commentaire que si vous avez quelque chose de pertinent à dire ou que Bugzilla impose que vous le fassiez. Dans le cas contraire, vous spammeriez les gens inutilement avec des courriels de bogues. Pour prendre un exemple : un utilisateur peut définir ses préférences pour filtrer les messages pour ne pas recevoir de courriel quand quelqu'un s'ajoute dans la liste `Copie à` d'un bogue (ce qui arrive souvent). Si vous vous ajoutez à la liste `Copie à` et que vous ajoutez un commentaire disant `Je m'ajoute dans la liste Copie à`, alors cette personne recevra un courriel sans intérêt qu'elle n'aurait pas reçu autrement.

N'utilisez pas de signature dans les commentaires. Signer avec votre prénom (`Jean`) est acceptable, si vous le faites par la force de l'habitude, mais des signatures complètes de style courriel ou liste de diffusion avec quatre lignes d'`ASCII art` ne le sont pas.

Si vous pensez qu'un bogue que vous avez rapporté a été incorrectement marqué comme `DOUBLON` d'un autre, veuillez l'indiquer dans votre bogue, pas dans celui dont il est le doublon. Vous pouvez mettre en copie la personne ayant marqué votre bogue comme étant un doublon si celle-ci n'est pas déjà en copie.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 2.8 Préférences utilisateur

Une fois connecté, vous pouvez personnaliser différents aspects de Bugzilla en cliquant sur le lien *Préférences* dans le pied de page. Les préférences sont divisées en cinq volets :

### 2.8.1 Préférences générales

Cet onglet vous permet de modifier plusieurs paramètres par défaut de Bugzilla. Les administrateurs ont la possibilité de supprimer des préférences de cette liste. Il est donc possible que vous ne puissiez toutes les voir.

### 2.8.2 Préférences de messagerie

Cet onglet permet d'activer ou de désactiver la notification par courriel en fonction d'événements spécifiques.

En général, les utilisateurs ont pratiquement tout contrôle sur la quantité de courriels que Bugzilla leur envoie. si vous voulez recevoir le maximum de courriels possible, cliquez sur le bouton *Activer tous les courriels*. Si vous ne voulez pas recevoir de courriels du tout de la part de Bugzilla, cliquez sur le bouton *Désactiver tous les courriels*.

---

**Note :** Un administrateur de Bugzilla peut empêcher un utilisateur de recevoir des courriels de bogues en cliquant sur la case *Envoi des courriels de bogues désactivé* lors de l'édition d'un compte utilisateur. C'est une décision drastique qui ne devrait être prise que pour les comptes désactivés, car cela outrepasserait les préférences de messagerie individuelles de l'utilisateur.

---

Il y a deux options globales : *Envoyer un courriel quand quelqu'un me demande de définir une étiquette* et *Envoyer un courriel quand quelqu'un a défini une étiquette que j'avais demandée*. Celles-ci définissent la réception de courriels en fonction des étiquettes. Leur utilisation est simple ; sélectionnez les cases à cocher si vous voulez que Bugzilla vous envoie des courriels pour l'une ou l'autre des conditions.

Si vous voulez paramétrer votre courriel de bogues pour quelque chose entre *Tous les courriels* et *Aucun courriel*, le tableau *Options spécifiques du champ/destinataire* vous permet de faire cela. Les lignes du tableau définissent les événements qui peuvent arriver à un bogue – des choses comme : un fichier a été joint, de nouveaux commentaires ont été faits, la priorité a changé, etc. Les colonnes du tableau définissent votre relation avec le bogue :

Pour affiner la gestion de la quantité de courriels de bogues, décidez des événements pour lesquels vous voulez recevoir des courriels de bogues ; décidez ensuite si vous voulez les recevoir tout le temps (cochez alors la case dans chaque colonne), ou seulement lorsque vous avez une certaine relation avec un bogue (cochez la case pour ces colonnes seulement). Par exemple : si vous ne voulez pas recevoir de courriel quand quelqu'un s'ajoute à la liste *Copie à*, vous pouvez décocher toutes les cases dans la ligne *Le champ « Copie à : » est modifié*. Comme autre exemple, si vous ne voulez jamais recevoir de courriel sur les bogues que vous avez rapportés à moins qu'ils ne soient résolus, vous devez décocher toutes les cases de la colonne *Rapporteur* sauf celle sur la ligne *Le bogue est résolu ou rouvert*.

---

**Note :** Bugzilla ajoute l'en-tête `X-Bugzilla-Reason` à tous les courriels qu'il envoie, décrivant la relation du destinataire (Assigné à, Rapporteur, Contact QA, Copie à ou Votant) au bogue. Cet en-tête peut être utilisé pour affiner le filtrage du côté client.

---

Bugzilla a une fonctionnalité appelée *Surveillance d'utilisateurs*. Quand vous saisissez un ou plusieurs comptes utilisateurs séparés par des virgules (généralement des adresses électroniques) dans la boîte de texte, vous recevez une copie de tous les courriels de bogues envoyés à ces utilisateurs (en fonction des paramètres de sécurité). Cette fonctionnalité puissante permet des transitions en douceur lorsque les développeurs changent de projet ou lorsque les utilisateurs partent en vacances.

Tous les utilisateurs indiqués dans le champ *Utilisateurs vous surveillant* vous ont ajoutés dans leur liste *Surveillance d'utilisateur* et peuvent obtenir des courriels de bogues en fonction de votre relation avec le bogue et leurs paramètres *Options spécifiques du champ/destinataire*.

La section *Ignorer les bogues* vous permet d'indiquer une liste de bogues séparés par des virgules pour lesquels vous ne voulez recevoir aucune notification par courriel. Supprimer un bogue de cette liste réactivera la notification par courriel pour ce bogue. Ceci est particulièrement utile par exemple si vous êtes le rapporteur d'un bogue générant beaucoup de commentaires et pour lequel vous n'êtes plus intéressé ou si vous surveillez quelqu'un recevant des notifications pour un tel bogue.

### 2.8.3 Recherches enregistrées

Dans ce volet, vous pouvez voir et exécuter toutes recherches enregistrées que vous avez créées, et aussi toutes autres recherches enregistrées partagées par d'autres membres du groupe défini par le paramètre `querysharegroup`. Les recherches enregistrées peuvent être ajoutées au pied de page à partir de cet écran. Si quelqu'un partage une recherche avec un groupe, il sera autorisé à y *assigner des utilisateurs*. La personne qui partage la recherche peut choisir de faire afficher la recherche dans le pied de page des membres directs du groupe par défaut.

### 2.8.4 Information de compte

Sur cet onglet, vous pouvez changer les informations de base de votre compte, y compris votre mot de passe, votre adresse électronique et votre vrai nom. Pour des raisons de sécurité, pour modifier quelque chose sur cette page, vous devez saisir votre mot de passe *actuel* dans le champ *Mot de passe* au sommet de la page. Si vous tentez de modifier votre adresse électronique, un courriel de confirmation est envoyé sur l'ancienne et la nouvelle adresses, contenant un lien pour confirmer le changement. Ceci aide à empêcher le piratage de compte.

### 2.8.5 Clés des API

Les clés d'API sont utilisées pour authentifier les appels aux API de web services. Vous pouvez créer plus d'une clé si nécessaire. Chaque clé d'API peut optionnellement avoir une description qui peut vous aider à vous rappeler à quoi

sert la clé.

Sur cette page, vous pouvez révoquer ou ré-autoriser la clé, changer la description de clés d'API existantes pour votre connexion. Une clé révoquée signifie qu'elle ne peut pas être utilisée. Les descriptions sont là pour votre seule information et sont optionnelles.

Vous pouvez également créer une nouvelle clé d'API en sélectionnant la case à cocher sous la section *Nouvelle clé d'API* de la page.

### 2.8.6 Permissions

C'est une page purement informative qui souligne vos permissions actuelles sur cette installation de Bugzilla.

Vous trouverez une liste complète des permissions ci-dessous. Seuls les utilisateurs ayant des privilèges *editusers* peuvent modifier les permissions d'autres utilisateurs.

**admin** Indique que l'utilisateur est un administrateur.

**bz\_canusewhineatothers** Indique que l'utilisateur peut configurer les rapports de notifications pour d'autres utilisateurs.

**bz\_canusewhines** Indique que l'utilisateur peut configurer les rapports de notifications pour lui-même.

**bz\_quip\_moderators** Indique que l'utilisateur peut modérer les citations.

**bz\_sudoers** Indique que l'utilisateur peut accomplir des actions à la place d'autres utilisateurs.

**bz\_sudo\_protect** Indique d'autres utilisateurs ne peuvent pas se faire passer pour cet utilisateur.

**canconfirm** Indique que l'utilisateur peut confirmer un bogue ou le marquer comme doublon.

**creategroups** Indique que l'utilisateur peut créer ou détruire des groupes.

**editbugs** Indique que l'utilisateur peut modifier tous les champs d'un bogue.

**editclassifications** Indique que l'utilisateur peut créer, détruire et modifier des classements.

**editcomponents** Indique que l'utilisateur peut créer, détruire et modifier des composants.

**editkeywords** Indique que l'utilisateur peut créer, détruire et modifier des mots-clés.

**editusers** Indique que l'utilisateur peut modifier ou désactiver des utilisateurs.

**tweakparams** Indique que l'utilisateur peut modifier les Paramètres.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 2.9 Extensions installées

Bugzilla peut être amélioré en utilisant des extensions (voir *Extensions*). Si une extension contient de la documentation dans le format approprié et que vous avez compilé votre propre copie de la documentation de Bugzilla en utilisant `makedocs.pl`, alors la documentation de vos extensions installées apparaîtront [ici](#).

Votre installation Bugzilla dispose des extensions suivantes (lors de votre dernière compilation de la documentation) :

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).



---

## Guide d'installation et de maintenance

---

**Note :** Si vous voulez seulement *utiliser* Bugzilla, vous n'avez pas besoin de l'installer. Ce chapitre n'est pas fait pour vous. Demandez à votre administrateur Bugzilla l'adresse pour y accéder dans votre navigateur Web. Vous voudrez peut-être consulter la section *Guide utilisateur*.

---

Bugzilla peut être installé sur Linux, Windows, Mac OS X et peut-être d'autres systèmes d'exploitation. Cependant, si vous installez Bugzilla sur une machine dédiée et que vous avez le contrôle du système d'exploitation à utiliser, l'équipe Bugzilla vous recommande vivement d'utiliser Linux car c'est un système d'exploitation extrêmement paramétrable, stable et robuste qui fournit un environnement idéal pour Bugzilla. Dans ce cas, vous voudrez peut-être consulter la section *Démarrage rapide*.

### 3.1 Démarrage rapide (Ubuntu Linux 14.04)

Ce guide de démarrage rapide propose l'installation de Bugzilla de la façon la plus simple possible pour les utilisateurs pouvant choisir leur environnement. Il décrit l'installation en utilisant Ubuntu Linux 14.04 LTS, Apache et MySQL, et installe Bugzilla comme page d'accueil par défaut. Il nécessite d'être familier avec l'environnement Linux et les lignes de commande.

#### 3.1.1 Le matériel

Ubuntu 14.04 LTS Server nécessite un processeur 64 bits. Bugzilla lui-même n'a pas de prérequis en dehors de ça, si ce n'est de choisir du matériel fiable. Vous pouvez aussi utiliser une machine virtuelle 64 bits ou une instance cloud sur laquelle vous avez un accès `root`.

#### 3.1.2 Installer le système d'exploitation

Téléchargez [Ubuntu Server 14.04 LTS](#) à partir du site de Ubuntu et suivez les [instructions d'installation](#). Quelques indications :

- Choisir le nom du serveur.
- Pour la création du premier utilisateur Linux, utiliser le nom `bugzilla`, et lui associer un mot de passe fort et le noter quelque part.
- Il n'est pas nécessaire d'avoir un répertoire `home` chiffré.
- Choisir le partitionnement proposé par défaut (sauf bien sûr, quand le défaut est `Non` et qu'il est nécessaire de cliquer sur `Oui` pour continuer).
- Choisir `Installer les mises à jour de sécurité automatiquement` sauf si vous voulez les faire manuellement.

- Dans les options d'installation, choisir `Serveur OpenSSH` et `Serveur LAMP`.
- Définir un mot de passe fort pour l'utilisateur `root` de MySQL et le noter quelque part.
- Installer le gestionnaire de démarrage `Grub` pour le `Master Boot Record`.

Redémarrer le serveur à la fin de l'installation.

### 3.1.3 Se substituer à l'utilisateur `root`

Connectez-vous en `ssh` sur la machine avec l'utilisateur `bugzilla` et démarrez une console. Puis :

```
sudo su
```

### 3.1.4 Installer les prérequis

```
apt-get install git nano
```

```
apt-get install apache2 mysql-server libappconfig-perl libdate-calc-perl libtemplate-perl libmime-perl build-essential libdatetime-timezone-perl libdatetime-perl libemail-sender-perl libemail-mime-perl libemail-mime-modifier-perl libdbi-perl libdbd-mysql-perl libcgi-pm-perl libmath-random-isaac-perl libmath-random-isaac-xs-perl apache2-mpm-prefork libapache2-mod-perl2 libapache2-mod-perl2-dev libchart-perl libxml-perl libxml-twig-perl perlmagick libgd-graph-perl libtemplate-plugin-gd-perl libsoap-lite-perl libhtml-scrubber-perl libjson-rpc-perl libdaemon-generic-perl libtheschwartz-perl libtest-taint-perl libauthen-radius-perl libfile-slurp-perl libencode-detect-perl libmodule-build-perl libnet-ldap-perl libauthen-sasl-perl libtemplate-perl-doc libfile-mimeinfo-perl libhtml-formattext-withlinks-perl libgd-dev lynx-cur python-sphinx
```

Ceci prend un peu de temps. Ces commandes sont séparées en deux parties de sorte que vous puissiez faire les étapes suivantes (jusqu'à l'étape 7) dans un autre terminal sans attendre la fin de l'exécution de la deuxième commande. Si vous démarrez un autre terminal, vous devez à nouveau exécuter la commande `sudo su`.

### 3.1.5 Télécharger Bugzilla

Récupérez-le à partir de notre dépôt Git :

```
cd /var/www
```

```
rm -rf html
```

```
git clone --branch bugzilla-X.X-stable https://git.mozilla.org/bugzilla/bugzilla.html
```

(où `X.X` est le numéro à deux chiffres de la version stable de Bugzilla que vous voulez - par ex. : 4.4)

```
cd html
```

### 3.1.6 Configurer MySQL

Les commandes suivantes utilisent l'éditeur minimaliste `nano`, mais vous pouvez utiliser n'importe quel autre éditeur de texte.

```
nano /etc/mysql/my.cnf
```

Définissez les valeurs suivantes, qui augmentent la taille maximale autorisée pour les fichiers joints et permettent la recherche de termes et mots courts :

- Modifiez la ligne 52 : `max_allowed_packet=100M`
- Ajoutez une nouvelle ligne 31, dans la section `[mysqld]` : `ft_min_word_len=2`

Enregistrez et quittez.

Ajoutez ensuite un utilisateur MySQL pour Bugzilla :

```
mysql -u root -p -e "GRANT ALL PRIVILEGES ON bugs.* TO bugs@localhost IDENTIFIED BY '$db_pass'"
```

Remplacez `$db_pass` par un mot de passe fort et notez-le quelque part. En lançant la commande ci-dessus, le mot de passe de l'utilisateur `root` de MySQL que vous avez configuré en installant Ubuntu vous sera demandé. Il est conseillé d'utiliser un mot de passe différent pour `$db_pass`.

Redémarrez MySQL :

```
service mysql restart
```

### 3.1.7 Configurer Apache

```
nano /etc/apache2/sites-available/bugzilla.conf
```

Copier le contenu suivant et enregistrez le fichier :

```
ServerName localhost

<Directory /var/www/html>
  AddHandler cgi-script .cgi
  Options +ExecCGI
  DirectoryIndex index.cgi index.html
  AllowOverride Limit FileInfo Indexes Options
</Directory>
```

```
a2ensite bugzilla
```

```
a2enmod cgi headers expires
```

```
service apache2 restart
```

### 3.1.8 Vérifier la configuration

Bugzilla fournit un script appelé `checksetup.pl` qui vous aidera dans le processus d'installation. Il devra être exécuté à deux reprises. La première fois, il génère un fichier (appelé `localconfig`) pour les informations de connexion à la base de données, et la seconde fois (étape 10), il utilise les informations que vous avez indiquées dans ce fichier pour paramétrer la base de données.

```
cd /var/www/html
```

```
./checksetup.pl
```

### 3.1.9 Éditer `localconfig`

```
nano localconfig
```

Vous devrez définir les valeurs suivantes :

- Ligne 29 : pour `$webservergroup` indiquez `www-data`
- Ligne 67 : pour `$db_pass` indiquez le mot de passe pour l'utilisateur MySQL `bugs` que vous avez créé un peu plus tôt

### 3.1.10 Vérifier la configuration (à nouveau)

Exécutez le script `checksetup.pl` à nouveau pour configurer la base de données.

**`./checksetup.pl`**

Le script demandera une adresse électronique, un nom réel et un mot de passe pour créer le premier compte Bugzilla, lequel sera administrateur. Notez l'adresse électronique et le mot de passe que vous avez indiqués.

### 3.1.11 Tester le serveur

**`./testserver.pl http://localhost/`**

Tous les tests devraient s'exécuter avec succès. Vous observerez des avertissements concernant l'obsolescence du module `Perl Chart::Base`. Ignorez-les.

### 3.1.12 Accéder à Bugzilla avec un navigateur Web

Accès à la page d'accueil :

**`lynx http://localhost/`**

Il n'est pas vraiment possible d'utiliser Bugzilla avec `Lynx`, mais vous pourrez accéder à la page d'accueil pour valider visuellement que Bugzilla est démarré et qu'il fonctionne.

Vous pourriez vouloir configurer votre DNS pour que le serveur soit accessible à l'aide d'un nom plutôt que de son adresse IP. Ce paramétrage n'est pas dans le périmètre de ce document. Le serveur Bugzilla est accessible sur votre réseau local sur l'URL suivante `http://<adresse IP>/`, où `<adresse IP>` est (à moins que votre paramétrage réseau soit complexe) la valeur affichée dans `inet addr` en exécutant la commande **`ifconfig eth0`**.

### 3.1.13 Configurer Bugzilla

Sur la page d'accueil de Bugzilla, cliquez sur *Se connecter* dans l'en-tête de la page, puis utilisez identifiants de l'utilisateur administrateur que vous avez défini à l'étape 10.

Cliquez sur le lien *Paramètres* puis définissez les paramètres suivants dans la section *Paramètres requis* :

— `urlbase` : `http://<nom_du_serveur>/` ou `http://<adresse IP>/`

Cliquez sur *Enregistrer les modifications* au bas de la page.

Il existe plusieurs façons pour que Bugzilla puisse envoyer des courriels. La façon la plus simple consiste à utiliser Gmail, ce que nous allons faire ici. Rendez-vous sur <https://gmail.com> et créez un nouveau compte Gmail pour Bugzilla. Ensuite, cliquez sur la section *Courriel* dans la page des Paramètres en utilisant le lien dans la colonne de gauche et indiquez les valeurs de paramètres suivantes :

- `mail_delivery_method` : SMTP
- `mailfrom` : `nouvelle_adresse_gmail@gmail.com`
- `smtpserver` : `smtp.gmail.com:465`
- `smtp_username` : `nouvelle_adresse_gmail@gmail.com`
- `smtp_password` : `nouveau_mot_de_passe_gmail`
- `smtp_ssl` : On

Cliquez sur le bouton *Enregistrez les modifications* au bas de la page.

Et voilà. :-)

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 3.2 Linux

Beaucoup de distributions GNU Linux/BSD incluent Bugzilla et ses dépendances dans leur système de gestion de paquets natif. Installer Bugzilla avec un accès `root` sur tout système GNU Linux/BSD devrait être aussi facile que de trouver le paquet Bugzilla dans l'application de gestion de paquet et de l'installer en utilisant la syntaxe de commande normale. Il peut y avoir toutefois un peu de paramétrage supplémentaire à faire.

Si vous installez votre machine à partir de zéro, *Démarrage rapide (Ubuntu Linux 14.04)* contient probablement les meilleures instructions dans votre cas.

### 3.2.1 Installer les paquets

Utilisez les gestionnaires de paquets de votre distribution pour installer Perl, votre moteur de base de données préféré (MySQL dans le doute) et un serveur Web (Apache dans le doute). Certaines distributions proposent même un paquet Bugzilla, cependant, celui-ci peut-être un peu vieux.

Les commandes ci-dessous installeront ceux-ci et d'autres paquets nécessaires au fonctionnement de Bugzilla. Si un paquet ne s'installe pas ou si le nom n'est pas trouvé, retirez-le de la liste et relancez la commande. Si vous utilisez un autre moteur de base de données ou un autre serveur Web, substituez les noms de paquets avec les libellés appropriés.

#### Fedora et Red Hat

Les commandes suivantes installeront le paquet Bugzilla fourni par Red Hat :

```
yum install bugzilla httpd mysql-server
```

Cependant, si vous utilisez cette voie, vous devez lire [bug 415605](#), qui détaille certains problèmes avec le paquet `Email::Send` package. Ensuite, vous pourrez alors consulter *configuration de la base de données*. Il peut être utile de savoir que Fedora stockent les fichiers Bugzilla dans `/usr/share/bugzilla`. C'est donc dans ce répertoire qu'il faudra exécuter `checksetup.pl`.

Si vous voulez installer une version de Bugzilla à partir du projet Bugzilla, vous aurez alors besoin des paquets suivants :

```
yum install httpd mysql-server mod_perl mod_perl-devel httpd-devel graphviz patchutils gcc  
'perl(Apache2 : :SizeLimit)' 'perl(Authen : :Radius)' 'perl(Authen : :SASL)' 'perl(Cache : :Mem-  
cached)' 'perl(CGI)' 'perl(Chart : :Lines)' 'perl(Daemon : :Generic)' 'perl(Date : :Format)'  
'perl(DateTime)' 'perl(DateTime : :TimeZone)' 'perl(DBI)' 'perl(Digest : :SHA)' 'perl(Email : :MIME)'  
'perl(Email : :MIME : :Attachment : :Stripper)' 'perl(Email : :Reply)' 'perl(Email : :Sender)' 'perl(Encode)'  
'perl(Encode : :Detect)' 'perl(File : :MimeType : :Magic)' 'perl(File : :Slurp)' 'perl(GD)' 'perl(GD : :Graph)'  
'perl(GD : :Text)' 'perl(HTML : :FormatText : :WithLinks)' 'perl(HTML : :Parser)' 'perl(HTML : :Scrub-  
ber)' 'perl(IO : :Scalar)' 'perl(JSON : :RPC)' 'perl(JSON : :XS)' 'perl(List : :MoreUtils)' 'perl(LWP : :User-  
Agent)' 'perl(Math : :Random : :ISAAC)' 'perl(MIME : :Parser)' 'perl(mod_perl2)' 'perl(Net : :LDAP)'  
'perl(Net : :SMTP : :SSL)' 'perl(PatchReader)' 'perl(SOAP : :Lite)' 'perl(Template)' 'perl(Template : :Plu-  
gin : :GD : :Image)' 'perl(Test : :Taint)' 'perl(TheSchwartz)' 'perl(URI)' 'perl(XMLRPC : :Lite)'  
'perl(XML : :Twig)'
```

Si vous utilisez RHEL6, vous devrez activer le canal `RHEL Server Optional` dans RHN pour obtenir certains paquets.

Si vous projetez d'utiliser SQLite comme moteur de base de données, vous devrez alors aussi installer les paquets appropriés.

## Ubuntu et Debian

### apt-get install git nano

```
apt-get install apache2 mysql-server libappconfig-perl libdate-calc-perl libtemplate-perl libmime-perl build-essential libdatetime-timezone-perl libdatetime-perl libemail-sender-perl libemail-mime-perl libemail-mime-modifier-perl libdbi-perl libdbd-mysql-perl libcgi-pm-perl libmath-random-isaac-perl libmath-random-isaac-xs-perl apache2-mpm-prefork libapache2-mod-perl2 libapache2-mod-perl2-dev libchart-perl libxml-perl libxml-twig-perl perlmagick libgd-graph-perl libtemplate-plugin-gd-perl libsoap-lite-perl libhtml-scrubber-perl libjson-rpc-perl libdaemon-generic-perl libtheschwartz-perl libtest-taint-perl libauthen-radius-perl libfile-slurp-perl libencode-detect-perl libmodule-build-perl libnet-ldap-perl libauthen-sasl-perl libtemplate-perl-doc libfile-mimeinfo-perl libhtml-formattext-withlinks-perl libgd-dev lynx-cur graphviz python-sphinx
```

Si vous projetez d'utiliser SQLite comme moteur de base de données, vous devrez alors aussi installer les paquets appropriés.

## Gentoo

### emerge -av bugzilla

installera Bugzilla et toutes ses dépendances. Si vous n'avez pas le flag vhosts USE activé, Bugzilla sera installé dans `/var/www/localhost/bugzilla`.

Vous pourrez ensuite vous rendre dans *Configuration de la base de données*.

### 3.2.2 Perl

Testez la version de Perl installée avec la commande suivante :

```
$ perl -v
```

Bugzilla nécessite au minimum la version Perl 5.10.1.

### 3.2.3 Bugzilla

Le meilleur moyen d'obtenir Bugzilla est de le faire par git :

```
git clone --branch bugzilla-X.X-stable https://git.mozilla.org/bugzilla/bugzilla
```

Exécutez la commande ci-dessus dans votre répertoire `home`, en remplaçant `X.X` avec les deux nombres de la version stable de Bugzilla que vous désirez - par ex. `4.4`.

Si ce n'est pas possible, vous pouvez télécharger l'archive de Bugzilla <<http://www.bugzilla.org/download/>>'.

Placez Bugzilla dans le répertoire approprié, accessible par l'utilisateur du serveur Web (probablement `apache` ou `www-data`). Un bon emplacement est soit directement à la racine du serveur Web (souvent `/var/www/html`) soit dans `/usr/local`, avec un lien symbolique vers la racine du serveur Web ou un alias dans la configuration du serveur Web.

|   |
|---|
| <p><b>Avertissement :</b> La distribution Bugzilla par défaut n'est PAS conçue pour être placée dans le répertoire <code>cgi-bin</code>. Ceci inclut également tout répertoire configuré en utilisant la directive <code>ScriptAlias</code> d'Apache.</p> |
|---|

### 3.2.4 Perl Modules

Bugzilla nécessite de nombreux modules Perl. Vous pouvez les installer globalement en utilisant le gestionnaire de paquets de votre distribution, ou en installant des copies utilisées seulement pour Bugzilla. Parfois, Bugzilla peut nécessiter une version d'un module Perl plus récent que celui fourni par votre distribution, auquel cas, vous devrez installer une copie de cette version utilisée par Bugzilla seulement.

Vous devrez alors être `root`, par ex. en utilisant la commande `su`. Vous devrez rester `root` jusqu'à la fin de l'installation. Ceci peut être évité dans certaines circonstances si vous êtes membre du groupe du serveur Web, mais être `root` est plus facile et fonctionne toujours.

Pour vérifier si vous avez tous les modules nécessaires, exécutez la commande :

```
./checksetup.pl --check-modules
```

Vous pouvez lancer cette commande autant de fois que nécessaire.

Si vous n'avez pas déjà tous les modules nécessaires, vous pouvez les installer en utilisant le gestionnaire de paquets de votre distribution. Vous pouvez alternativement installer tous les modules manquants localement (c-à-d. seulement pour Bugzilla) comme ceci :

```
./install-module.pl --all
```

Vous pouvez également indiquer un nom de module individuel :

```
./install-module.pl <modulename>
```

---

**Note :** Si vous utilisez un paquet Bugzilla de votre distribution et que voulez installer des modules Perl en utilisant CPAN (par ex. en utilisant `install-module.pl`), vous pourrez avoir besoin d'installer les paquets de développement pour MySQL et GD avant d'installer les modules Perl relatifs. Les noms de ces paquets peuvent varier d'une distribution à l'autre mais sont souvent appelés `<nom_du_paquet>-devel`.

---

### 3.2.5 Serveur Web

Tout serveur Web en mesure d'exécuter des scripts CGI peut être utilisé. Nous avons des instructions spécifiques pour les suivants :

- *Apache*

### 3.2.6 Moteur de base de données

Bugzilla peut fonctionner avec les moteurs de base de données MySQL, PostgreSQL, Oracle et SQLite. Vous n'avez besoin que d'un seul de ces moteurs pour utiliser Bugzilla. MySQL est le plus couramment utilisé. SQLite est adapté pour des installations de test car il ne nécessite pas de configuration. Configurez votre serveur en suivant les instructions ci-dessous :

- *MySQL*
- *PostgreSQL*
- *Oracle*
- *SQLite*

### 3.2.7 localconfig

Vous devez maintenant exécuter `checksetup.pl` à nouveau, cette fois sans l'argument `--check-modules`.

```
./checksetup.pl
```

Cette fois, `checksetup.pl` devrait vous dire que tous les modules appropriés sont installés et affichera un message à ce sujet, et générera un fichier de sortie appelé `localconfig`. Ce fichier contient les paramètres par défaut pour un grand nombre de paramètres de Bugzilla.

Ouvrez ce fichier dans votre éditeur. Les deux seules valeurs que vous avez *besoin* de changer sont `$db_driver` et `$db_pass`, respectivement le type de base de données et le mot de passe pour l'utilisateur qui créera pour vous la base de données. Choisissez un mot de passe compliqué (pour la simplicité, il ne devrait pas contenir d'apostrophe) et saisissez-le dans le fichier. `$db_driver` peut être `mysql`, `Pg` (PostgreSQL), `oracle` ou `SQLite`.

Définissez la valeur de `$webservergroup` avec le nom groupe avec lequel votre serveur Web s'exécute.

- Fedora/Red Hat : `apache`
- Debian/Ubuntu : `www-data`
- Mac OS X : `_www`
- Windows : ignorez ce paramètre car il n'est pas utile pour ce système d'exploitation

Les autres options dans le fichier `localconfig` sont documentées avec leurs commentaires. Si vous avez une configuration de base de données non standard, vous aurez peut-être besoin de modifier d'autres paramètres `$db_*`.

---

**Note :** Sous Oracle, `$db_name` devrait en fait être le nom du SID de votre base de données (par ex. XE si vous utilisez Oracle XE).

---

### 3.2.8 checksetup.pl

Ensuite, exécutez `checksetup.pl` une nouvelle fois :

**`./checksetup.pl`**

Il confirmera à nouveau que tous les modules sont présents et remarquera la modification du fichier `localconfig`, en supposant que vous l'avez modifié à votre convenance. Il compile ensuite les modèles de l'interface utilisateur, se connecte à la base de données en utilisant l'utilisateur `bugs` que vous avez créé et le mot de passe que vous avez défini et crée enfin la base de données `bugs` et les tables à l'intérieur.

Après cela, il demande des détails sur le compte administrateur. Bugzilla peut avoir plusieurs administrateurs –vous pouvez en créer d'autres plus tard– mais il en a besoin d'un pour démarrer. Saisissez l'adresse électronique d'un administrateur, son nom complet, et un mot de passe approprié pour Bugzilla.

`checksetup.pl` se terminera alors. Vous pouvez relancer `checksetup.pl` à tout moment si vous le souhaitez.

### 3.2.9 Bravo

Votre installation Bugzilla devrait à présent fonctionner. Vérifiez-le en exécutant la commande suivante :

**`./testserver.pl http://<your-bugzilla-server>/`**

Si elle passe sans erreur, accédez à `http://<your-bugzilla-server>/` dans votre navigateur –vous devriez alors voir la page d'accueil de Bugzilla. Bien sûr, si vous avez installé Bugzilla dans un sous-répertoire, assurez-vous que celui-ci figure dans URL.

Ensuite, consultez *Configuration post-installation essentielle*.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).



## 3.3 Windows

Faire fonctionner Bugzilla sous Windows est plus difficile que sous Linux, peu de développeurs Bugzilla l'utilise et Windows est donc moins bien supporté. Nous vous recommandons vivement de ne pas utiliser Windows en prévision d'une installation importante. Cependant, si vous êtes déterminé à continuer, voici comment.

### 3.3.1 ActiveState Perl

ActiveState est une distribution populaire de Perl pour Windows.

Téléchargez l'installateur ActiveState Perl 5.12.4 ou version supérieure sur le [site Web de ActiveState](#).

ActiveState Perl utilise un installateur Windows standard. Procédez à l'installation en conservant les paramètres par défaut, ce qui installera Perl dans le répertoire `C:\Perl`. Il n'est pas recommandé d'installer Perl dans un répertoire dont le nom contient une espace, tel que `C:\Program Files`.

Une fois l'installation terminée, fermez votre session Windows et reconnectez-vous pour que les changements dans la variable d'environnement `PATH` soient pris en compte.

### 3.3.2 Bugzilla

Le meilleur moyen de récupérer Bugzilla est de l'obtenir avec Git. Téléchargez et installez Git à partir du [site Web de Git](#), puis exécutez la commande suivante :

```
git clone --branch bugzilla-X.X-stable https://git.mozilla.org/bugzilla/bugzilla C:\bugzilla
```

où `X.X` est le numéro de version à deux chiffres de la version stable que vous voulez installer (par ex. : `4.4`).

La suite de cette documentation suppose que vous avez installé Bugzilla dans le répertoire `C:\bugzilla`. Ajustez les chemins d'accès en conséquence si ce n'est pas le cas.

S'il n'est pas possible d'utiliser Git (par ex. parce que votre machine n'a pas d'accès à Internet), vous pouvez [télécharger une archive de Bugzilla](#) et la recopier sur votre machine. Bugzilla est livré sous forme d'archive (extension `.tar.gz`), qui peut être décompressée par tout archiveur Windows reconnaissant ce format.

### 3.3.3 Modules Perl

Bugzilla nécessite de nombreux modules Perl. Ils sont disponibles dans le dépôt de ActiveState et sont installés avec l'outil `ppm`. Vous pouvez l'utiliser en ligne de commande, comme ci-dessous, ou saisir la commande `ppm` pour obtenir l'interface graphique.

Si vous utilisez un proxy ou un pare-feu, vous pourriez rencontrer des difficultés pour utiliser PPM. Ceci est abordé dans la [FAQ ActivePerl](#).

Installez les modules suivants avec la commande :

```
ppm install <nom_du_module>
```

- CGI.pm
- Digest-SHA
- TimeDate
- DateTime
- DateTime-TimeZone
- DBI
- Template-Toolkit
- Email-Sender
- Email-MIME

- URI
- List-MoreUtils
- Math-Random-ISAAC
- File-Slurp
- JSON-XS
- Win32
- Win32-API

Les modules suivants activent diverses fonctionnalités optionnelles de Bugzilla :

- GD
- Chart
- Template-GD
- GDTextUtil
- GDGraph
- MIME-tools
- libwww-perl
- XML-Twig
- PatchReader
- perl-ldap
- Authen-SASL
- Net-SMTP-SSL
- RadiusPerl
- SOAP-Lite
- XMLRPC-Lite
- JSON-RPC
- Test-Taint
- HTML-Parser
- HTML-Scrubber
- Encode
- Encode-Detect
- Email-Reply
- HTML-FormatText-WithLinks
- TheSchwartz
- Daemon-Generic
- mod\_perl
- Apache-SizeLimit
- File-MimeInfo
- IO-stringy
- Cache-Memcached
- File-Copy-Recursive
- GraphViz

|  |
|--|
| <p><b>Avertissement :</b> Ces listes ont été extraites du code source de Bugzilla et n'ont pas été testées. Veuillez nous faire savoir si vous rencontrez des erreurs.</p> |
|--|

---

**Note :** Le script `install-module.pl` ne fonctionne pas avec ActivePerl sous Windows.

---

### 3.3.4 Serveur Web

Tout serveur Web capable d'exécuter des scripts CGI peut fonctionner. Nous avons des instructions spécifiques pour les suivants :

- *Apache*
- *Microsoft IIS*

### 3.3.5 Moteurs de base de données

Bugzilla peut fonctionner avec MySQL, PostgreSQL, Oracle et SQLite. Un seul de ces moteurs de base de données est nécessaire pour utiliser Bugzilla. MySQL est le plus couramment utilisé et c'est le seul avec lequel les instructions pour Windows ont été testées. SQLite est pratique pour des installations de tests car il ne nécessite pas de configuration. Configurez votre serveur selon les instructions ci-dessous :

- *MySQL*
- *PostgreSQL*
- *Oracle*
- *SQLite*

### 3.3.6 localconfig

Vous devez maintenant exécuter `checksetup.pl` à nouveau, cette fois sans l'argument `--check-modules`.

#### checksetup.pl

Cette fois, `checksetup.pl` devrait vous dire que tous les modules appropriés sont installés et affichera un message à ce sujet, et générera un fichier de sortie appelé `localconfig`. Ce fichier contient les paramètres par défaut pour un grand nombre de paramètres de Bugzilla.

Ouvrez ce fichier dans votre éditeur. Les deux seules valeurs que vous avez *besoin* de changer sont `$db_driver` et `$db_pass`, respectivement le type de base de données et le mot de passe pour l'utilisateur qui créera pour vous la base de données. Choisissez un mot de passe compliqué (pour la simplicité, il ne devrait pas contenir d'apostrophe) et saisissez-le dans le fichier. `$db_driver` peut être `mysql`, `Pg` (PostgreSQL), `oracle` ou `SQLite`.

Définissez la valeur de `$webservergroup` avec le nom groupe avec lequel votre serveur Web s'exécute.

- Fedora/Red Hat : `apache`
- Debian/Ubuntu : `www-data`
- Mac OS X : `_www`
- Windows : ignorez ce paramètre car il n'est pas utile pour ce système d'exploitation

Les autres options dans le fichier `localconfig` sont documentées avec leurs commentaires. Si vous avez une configuration de base de données non standard, vous aurez peut-être besoin de modifier d'autres paramètres `$db_*`.

---

**Note :** Sous Oracle, `$db_name` devrait en fait être le nom du SID de votre base de données (par ex. XE si vous utilisez Oracle XE).

---

### 3.3.7 checksetup.pl

Ensuite, exécutez `checksetup.pl` une nouvelle fois :

#### checksetup.pl

Il confirmera à nouveau que tous les modules sont présents et remarquera la modification du fichier `localconfig`, en supposant que vous l'avez modifié à votre convenance. Il compile ensuite les modèles de l'interface utilisateur, se connecte à la base de données en utilisant l'utilisateur `bugs` que vous avez créé et le mot de passe que vous avez défini et crée enfin la base de données `bugs` et les tables à l'intérieur.

Après cela, il demande des détails sur le compte administrateur. Bugzilla peut avoir plusieurs administrateurs –vous pouvez en créer d'autres plus tard– mais il en a besoin d'un pour démarrer. Saisissez l'adresse électronique d'un administrateur, son nom complet, et un mot de passe approprié pour Bugzilla.

`checksetup.pl` se terminera alors. Vous pouvez relancer `checksetup.pl` à tout moment si vous le souhaitez.

### 3.3.8 Bravo

Votre installation Bugzilla devrait à présent fonctionner. Vérifiez-le en exécutant la commande suivante :

```
testserver.pl http ://<your-bugzilla-server>/
```

Si elle passe sans erreur, accédez à `http://<your-bugzilla-server>/` dans votre navigateur –vous devriez alors voir la page d’accueil de Bugzilla. Bien sûr, si vous avez installé Bugzilla dans un sous-répertoire, assurez-vous que celui-ci figure dans URL.

Si vous ne voyez pas la page d’accueil de Bugzilla, mais « It works !!! », c’est qu’Apache n’a pas pris en compte vos modifications dans le fichier `httpd.conf`. Si vous êtes sous Windows 7 ou versions suivantes, cela peut être dû à une nouvelle fonctionnalité appelée « VirtualStore ». [Ce billet de blog](#) peut vous aider à résoudre ce problème.

Si vous obtenez un message « Internal Error... », c’est peut-être parce que `ScriptInterpreterSource Registry-Strict` n’est pas défini dans votre *Configuration Apache*. Vérifiez à nouveau si ce paramètre est correctement défini.

Ensuite, consultez *Configuration post-installation essentielle*.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 3.4 Mac OS X

---

**Note :** L’équipe Bugzilla a très peu d’expertise sur Mac et nous n’avons pas été capables de réussir l’installation sur la dernière version de Mac OS X. Nous y étions presque cependant. Si vous avez réussi, dites-nous comment et nous pourrions mettre à jour la documentation !

---

### 3.4.1 Installer les paquets

OS X 10.7 fournit Perl 5.12 et Apache 2.2. Installez les paquets supplémentaires suivants :

- git : Téléchargez un installateur à partir du [site Web de Git](#).
- MySQL : Téléchargez un installateur à partir du [site Web de MySQL](#).

### 3.4.2 Bugzilla

Le meilleur moyen d’obtenir Bugzilla est de le faire par git :

```
git clone --branch bugzilla-X.X-stable https://git.mozilla.org/bugzilla/bugzilla
```

Exécutez la commande ci-dessus dans votre répertoire `home`, en remplaçant `X.X` avec les deux nombres de la version stable de Bugzilla que vous désirez - par ex. `4.4`. Ceci placera Bugzilla dans le répertoire `$HOME/bugzilla`.

Si ce n’est pas possible, vous pouvez [télécharger l’archive de Bugzilla](#).

### 3.4.3 Bibliothèques système additionnelles

Apple ne fournit pas la bibliothèque GD avec Mac OS X. Bugzilla en a besoin si vous voulez afficher des graphiques de bogues, et vous devrez l’installer avant d’essayer d’installer le module Perl GD.

Vous pouvez utiliser [MacPorts](#), [Homebrew](#) ou [Fink](#), chacun d’eux pouvant installer les programmes Linux courants sur Mac OS X.

---

Si vous n'avez encore aucun d'entre eux installé, choisissez-en un et suivez les instructions pour l'installer. Puis, utilisez-le pour installer le paquet `gd2` (MacPorts/Fink) ou le paquet `gd` (Brew).

Le gestionnaire de paquets peut vous inviter à installer des dépendances qu'il faudra accepter.

---

**Note :** Pour éviter les conflits avec les logiciels installés par défaut par Apple, Fink crée sa propre arborescence sur `/sw` où il installe la plupart des logiciels. Ceci signifie que les bibliothèques et les en-têtes seront situés dans `/sw/lib` et `/sw/include` au lieu de `/usr/lib` et `/usr/include`. Quand le script de configuration du module Perl GD demandera où se trouve `libgd`, assurez-vous d'indiquer `/sw/lib`.

---

### 3.4.4 Modules Perl

Bugzilla nécessite de nombreux modules Perl. Sous Mac OS X, le moyen le plus simple est d'installer des copies locales (plutôt que des copies système globales) des modules que vous n'avez pas encore. Cependant, si vous voulez les installer pour tout le système, exécutez les commandes qui suivent en tant que root avec l'argument **-global**.

Pour vérifier si vous disposez déjà de tous les modules nécessaires, exécutez la commande suivante :

```
perl checksetup.pl --check-modules
```

Vous pouvez exécuter cette commande autant de fois que nécessaire.

Pour installer localement tous les modules manquants, exécutez la commande suivante :

```
perl install-module.pl --all
```

### 3.4.5 Serveur Web

Tout serveur Web en mesure d'exécuter des scripts CGI peut être utilisé. Nous avons des instructions spécifiques pour les suivants :

- *Apache*

Vous devrez créer un lien symbolique pour que le serveur Web puisse localiser Bugzilla :

```
cd /Library/WebServer/Documents
```

```
sudo ln -s $HOME/bugzilla bugzilla
```

Dans les *Préférences système* → *Partage*, cocher la case *Partage Web* pour démarrer Apache.

### 3.4.6 Database Engine

Bugzilla peut fonctionner avec les moteurs de base de données MySQL, PostgreSQL, Oracle et SQLite. Vous n'avez besoin que d'un seul de ces moteurs pour utiliser Bugzilla. MySQL est le plus couramment utilisé sous Mac OS X —en fait, nous avons pas connaissance de personnes utilisant autre chose. Configurez votre serveur en suivant les instructions ci-dessous :

- *MySQL*
- *PostgreSQL*
- *Oracle*
- *SQLite*

### 3.4.7 localconfig

Vous devez maintenant exécuter `checksetup.pl` à nouveau, cette fois sans l'argument `--check-modules`.

```
perl checksetup.pl
```

Cette fois, `checksetup.pl` devrait vous dire que tous les modules appropriés sont installés et affichera un message à ce sujet, et générera un fichier de sortie appelé `localconfig`. Ce fichier contient les paramètres par défaut pour un grand nombre de paramètres de Bugzilla.

Ouvrez ce fichier dans votre éditeur. Les deux seules valeurs que vous avez *besoin* de changer sont `$db_driver` et `$db_pass`, respectivement le type de base de données et le mot de passe pour l'utilisateur qui créera pour vous la base de données. Choisissez un mot de passe compliqué (pour la simplicité, il ne devrait pas contenir d'apostrophe) et saisissez-le dans le fichier. `$db_driver` peut être `mysql`, `Pg` (PostgreSQL), `oracle` ou `SQLite`.

Définissez la valeur de `$webservergroup` avec le nom groupe avec lequel votre serveur Web s'exécute.

- Fedora/Red Hat : `apache`
- Debian/Ubuntu : `www-data`
- Mac OS X : `_www`
- Windows : ignorez ce paramètre car il n'est pas utile pour ce système d'exploitation

Les autres options dans le fichier `localconfig` sont documentées avec leurs commentaires. Si vous avez une configuration de base de données non standard, vous aurez peut-être besoin de modifier d'autres paramètres `$db_*`.

---

**Note :** Sous Oracle, `$db_name` devrait en fait être le nom du SID de votre base de données (par ex. XE si vous utilisez Oracle XE).

---

### 3.4.8 checksetup.pl

Ensuite, exécutez `checksetup.pl` une nouvelle fois :

#### **perl checksetup.pl**

Il confirmera à nouveau que tous les modules sont présents et remarquera la modification du fichier `localconfig`, en supposant que vous l'avez modifié à votre convenance. Il compile ensuite les modèles de l'interface utilisateur, se connecte à la base de données en utilisant l'utilisateur `bugs` que vous avez créé et le mot de passe que vous avez défini et crée enfin la base de données `bugs` et les tables à l'intérieur.

Après cela, il demande des détails sur le compte administrateur. Bugzilla peut avoir plusieurs administrateurs –vous pouvez en créer d'autres plus tard– mais il en a besoin d'un pour démarrer. Saisissez l'adresse électronique d'un administrateur, son nom complet, et un mot de passe approprié pour Bugzilla.

`checksetup.pl` se terminera alors. Vous pouvez relancer `checksetup.pl` à tout moment si vous le souhaitez.

### 3.4.9 Bravo

Votre installation Bugzilla devrait à présent fonctionner. Vérifiez-le en exécutant la commande suivante :

**perl testserver.pl http://<your-bugzilla-server>/**

Si elle passe sans erreur, accédez à `http://<your-bugzilla-server>/` dans votre navigateur –vous devriez alors voir la page d'accueil de Bugzilla. Bien sûr, si vous avez installé Bugzilla dans un sous-répertoire, assurez-vous que celui-ci figure dans URL.

Ensuite, consultez *Configuration post-installation essentielle*.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 3.5 Configuration post-installation essentielle

Bugzilla est configuré dans les paramètres d'administration. Connectez-vous avec le compte administrateur que vous avez défini lors de la dernière exécution de `checksetup.pl`, puis cliquez sur *Administration* dans l'en-tête, puis sur *Paramètres*. Vous verrez alors les différentes sections des paramètres.

### 3.5.1 Paramètres

Il y a beaucoup de paramètres importants à définir (ou à décider de ne pas changer).

Les premiers d'entre eux sont dans la section *Paramètres requis*.

- `urlbase` : L'URL racine de cette installation de Bugzilla.
- `sslbase` : L'URL racine de toutes les URL HTTPS (SSL) de Bugzilla.
- `ssl_redirect` : Définissez ce paramètre si vous voulez que tout le monde soit redirigé vers la version SSL. Recommandé si vous avez paramétré SSL.
- `cookiebase` : Bugzilla utilise des cookies pour se rappeler des utilisateurs. Pour que les cookies soient correctement définis, vous devrez définir une racine pour les cookies. Si Bugzilla est à la racine de votre domaine, vous n'avez pas besoin de modifier la valeur par défaut.

Vous pourriez vouloir indiquer votre adresse électronique dans le paramètre `maintainer` dans la section *Général*. Cela permettra aux utilisateurs de savoir qui contacter en cas de problèmes.

Si vous voulez que personne ne puisse lire les pages de Bugzilla, définissez le paramètre `requirelogin` dans la section *Authentification utilisateur* et modifier ou effacer la valeur du paramètre `createemailregexp`.

### 3.5.2 Courriel

Bugzilla nécessite l'utilisation de courriels. Il existe plusieurs possibilités. Le plus simple est d'utiliser une adresse de Gmail ou d'un autre fournisseur pour faire le travail, mais vous pouvez aussi utiliser un serveur de messagerie local ou en installer un sur le serveur hébergeant Bugzilla.

Les paramètres concernant les courriels sont définis dans la section *Courriel*.

#### Utiliser un autre serveur de messagerie

Cette section correspond au choix `mail_delivery_method` dans la partie SMTP.

Cette méthode traite les courriels en utilisant un serveur de messagerie existant. Votre organisation en dispose peut-être déjà d'un pour la gestion des courriels internes et préfère l'utiliser pour des raisons de confidentialité. Si c'est le cas, vous aurez besoin des informations suivantes :

- Le nom de domaine du serveur (paramètre : `smtpserver`)
- Le nom d'utilisateur et le mot de passe à utiliser (paramètres : `smtp_username` et `smtp_password`)
- Si le serveur utilise SSL (paramètre : `smtp_ssl`)
- L'adresse à partir de laquelle envoyer les courriels (paramètre : `mailfrom`)

Si votre organisation ne dispose pas de son propre serveur de messagerie, vous pouvez utiliser les services de différents fournisseurs.

#### Gmail

Rendez-vous sur <https://gmail.com> et créez un nouveau compte Gmail pour Bugzilla. Ensuite, renseignez les paramètres suivants dans la section *Courriel* :

- `mail_delivery_method` : SMTP
- `mailfrom` : `nouvelle_adresse_gmail@gmail.com`

- smtpserver : smtp.gmail.com :465
- smtp\_username : nouvelle\_adresse\_gmail@gmail.com
- smtp\_password : nouveau\_mot\_de\_passe\_gmail
- smtp\_ssl : On

### Utiliser son propre serveur de messagerie

Cette section correspond au choix de `mail_delivery_method` de Sendmail.

À moins de savoir ce que vous faites et de pouvoir gérer les possibles problèmes de courriels indésirables, de rejet et de listes noires, il n'est pas recommandé de configurer votre serveur de messagerie juste pour Bugzilla. Cependant, si vous souhaitez le faire, voici quelques indications.

Sous GNU/Linux, tout MTA (Mail Transfer Agent) compatible avec Sendmail suffira. Sendmail, Postfix, qmail et Exim sont des exemples de MTA courants. Sendmail est le MTA original d'Unix, mais les autres sont plus faciles à configurer et par conséquent, beaucoup de gens remplacent Sendmail par Postfix ou Exim. Ce sont des remplacements transparents donc Bugzilla ne fera pas la différence.

Si vous utilisez Sendmail, la version 8.7 ou supérieure est requise. Si vous utilisez un MTA compatible avec Sendmail, il doit être congruent avec au moins la version 8.7 de Sendmail.

Sous OS X 10.3 et versions suivantes, Postfix est utilisé comme solution de serveur de messagerie par défaut. Postfix fournit un exécutable qui se comporte comme sendmail et suffit pour utiliser Bugzilla.

Sous Windows, si vous n'arrivez pas à utiliser la gestion SMTP intégrée à Bugzilla (par ex. parce que les modules Perl nécessaires ne sont pas disponibles), vous pouvez utiliser Sendmail avec une petite application appelée `sendmail.exe`, qui fournit des fonctionnalités compatibles sendmail et encapsule la communication SMTP pour un autre serveur de messagerie. Comme Bugzilla, `sendmail.exe` peut être configuré pour journaliser les communications SMTP dans un fichier pour pouvoir déboguer en cas de problème.

Le paramétrage d'un MTA est en dehors du périmètre de ce document. Consultez le manuel du MTA que vous avez choisi pour des instructions d'installation détaillées. Chacun de ces programmes a ses propres fichiers de configuration où vous devez configurer certains paramètres pour vous assurer que les courriels seront distribués correctement. Ils sont mis en œuvre en tant que services et vous devez vous assurer que le MTA est dans la liste de démarrage automatique des services de votre machine.

Si un simple courriel envoyé avec la ligne de commande `mail` fonctionne, alors Bugzilla devrait fonctionner correctement.

### Dépannage

Si vous rencontrez des problèmes, vérifiez que le serveur SMTP peut être joint pas votre serveur Bugzilla et que les identifiants de connexion sont valides. Si le paramétrage vous semble correct et que vos courriels ne sont toujours pas envoyés, vérifiez si votre distribution utilise SELinux ou AppArmor. Ceux-ci peuvent empêcher votre serveur Web d'envoyer des courriels. Le paramètre booléen SELinux `httpd_can_sendmail` devra peut-être être défini à « True ».

Si cela ne fonctionne toujours pas, activez le paramètre `smtp_debug` et consultez les journaux de votre serveur Web.

### 3.5.3 Produits, composants, versions et jalons

Les bogues dans Bugzilla sont classés dans des Produits, lesquels contiennent des composants. Vous pouvez facultativement activer le paramètre `useclassifications`. Les Catégories sont un niveau supérieur des Produits).

Bugzilla fournit par défaut un produit appelé « TestProduct », qui contient un seul composant, appelé avec beaucoup d'imagination « TestComponent ». Vous pouvez créer vos propres produits et composants. Vous pouvez avoir un seul composant par produit si vous le voulez. Les produits ont des versions (qui représentent la version du logiciel dans



lequel le bogue a été trouvé) et des jalons cibles (qui représentent la future version dans lequel le bogue sera corrigé –ou, pour les bogues avec l'état RÉSOLU, la version dans laquelle ils ont été résolus).

Quand vos produits seront créés, vous pourrez supprimer le produit « TestProduct » (ce qui supprimera automatiquement le composant « TestComponent »). Veuillez noter que si vous avez créé un bogue dans « TestProduct » pour faire un essai, vous devrez le déplacer dans un autre produit avant de pouvoir supprimer « TestProduct ».

Maintenant, vous voudrez peut-être effectuer d'autres paramétrage optionnels dans *Configuration post-installation facultative*.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 3.6 Configuration post-installation facultative

Bugzilla dispose de nombreuses fonctionnalités optionnelles. Cette section décrit comment les configurer ou les activer.

### 3.6.1 Tâches récurrentes

Plusieurs des fonctionnalités ci-dessous nécessitent d'écrire un script à exécuter à intervalles réguliers. La méthode pour ce faire varie selon les systèmes d'exploitation.

#### Linux

Exécutez :

**crontab -e**

Ceci doit ouvrir le fichier crontab dans votre éditeur. Ajoutez la ligne cron des sections ci-dessous pour activer la fonctionnalité correspondante.

#### Windows

Windows propose un gestionnaire de tâches. Pour exécuter un script spécifique, suivez les instructions ci-dessous :

1. *Panneau de configuration* → *Tâches programmées* → *Ajouter une tâche programmée*
2. Suivant
3. Parcourir
4. Rechercher `perl.exe` (normalement `C:\Perl\bin\perl.exe`)
5. Donnez un nom à la tâche, par ex. `Bugzilla <nom_du_script>`
6. Paramétrez la tâche pour s'exécuter au moment et pendant l'intervalle choisis
7. Si vous exécutez Apache avec un utilisateur particulier, pas en tant que SYSTEM, saisissez cet utilisateur ici. Sinon, créez un compte qui a les droits d'écriture dans le répertoire de Bugzilla et utilisez-le ici
8. Cochez *Ouvrir les propriétés avancées...* et cliquez sur *Terminer*
9. Ajoutez le nom du script à la fin du champ *Exécuter*. Par exemple : `C:\Perl\bin\perl.exe C:\Bugzilla\<nom_du_script>`
10. Modifiez le champ *Démarrer* dans en indiquant le répertoire de Bugzilla

### 3.6.2 Graphiques de bogues

Si vous avez installé les modules Perl nécessaires indiqués par le script `checksetup.pl`, vous pouvez demander à Bugzilla de collecter régulièrement des statistiques pour obtenir des graphiques et des tableaux.

Sous GNU/Linux, utilisez une ligne `cron` comme suit :

```
5 0 * * * cd <votre-répertoire-bugzilla> && ./collectstats.pl
```

Sous Windows, programmez le script `collectstats.pl` pour s'exécuter quotidiennement.

Après deux jours, vous pourrez voir des graphiques de bogues dans la page `Rapports`.

### 3.6.3 Notifications

Les notifications sont une fonctionnalité de Bugzilla qui peut venir ennuyer régulièrement les utilisateurs à des moments spécifiés. En utilisant cette fonctionnalité, les utilisateurs peuvent exécuter des recherches enregistrées à des moments spécifiques (c-à-d. le 15 du mois à minuit) ou à intervalles réguliers (c-à-d. toutes les 15 minutes le dimanche). Les résultats de la recherche sont envoyés à l'utilisateur, soit dans un seul courriel, soit un courriel par bogue, accompagné d'un texte descriptif. Pour que les notifications fonctionnent, un script Perl spécial doit être exécuté à intervalles réguliers. Plus de détails sont disponibles dans *Notifications*.

Sous GNU/Linux, utilisez une ligne `cron` comme suit :

```
*/15 * * * * cd <your-bugzilla-directory> && ./whine.pl
```

Sous Windows, programmez le script `whine.pl` pour s'exécuter toutes les 15 minutes.

### 3.6.4 Notifications pour les bogues non triés

Il est possible que des bogues stagnent dans un état `NON CONFIRMÉ`. Bugzilla a un système spécifique pour relancer les responsables de ces bogues par courriel.

Sous GNU/Linux, utilisez une ligne `cron` comme suit :

```
55 0 * * * cd <your-bugzilla-directory> && ./whineatnews.pl
```

Sous Windows, programmez le script `whineatnews.pl` pour s'exécuter quotidiennement.

### 3.6.5 Graphiques de dépendance

Bugzilla peut générer des graphiques de dépendance (relations dépend de/bloque) entre les bogues, si vous installez le paquet appelé `dot`.

#### Linux

Indiquez le chemin d'accès complet à la commande `dot` (provenant du paquet `graphviz`) dans le paramètre `webdotbase`. Par ex. `/usr/bin/dot`.

#### Windows

Téléchargez et installez Graphviz du site [Web Graphviz](#). Indiquez le chemin d'accès complet à l'exécutable `dot.exe` dans le paramètre `webdotbase`, en utilisant des barres de fraction (« / ») comme séparateurs de chemin. Par ex. : `C:/Program Files/ATT/Graphviz/bin/dot.exe`.

### 3.6.6 Documentation

Bugzilla contient une documentation d'aide complète, écrite en format *reStructured Text*. Une copie générique compilée est disponible sur [bugzilla.readthedocs.org](http://bugzilla.readthedocs.org), et les liens d'*Aide* pointe vers ce site par défaut. Si vous voulez compiler et utiliser une copie locale de la documentation, peut-être parce que vous avez ajouté des extensions Bugzilla comportant de la documentation, ou parce que vos utilisateurs n'ont pas accès à Internet à partir de leurs machines, alors :

— Installez *Sphinx* (paquet `python-sphinx` sous Debian/Ubuntu)

Puis exécutez le script `docs/makedocs.pl` dans votre répertoire Bugzilla.

Bugzilla détectera automatiquement que vous avez compilé la documentation et fera le lien vers celle-ci plutôt que la copie sur Internet. N'oubliez pas de la recompiler quand vous faites une mise à jour de Bugzilla ou quand vous installez de nouvelles extensions.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 3.7 Migrer à partir d'un autre logiciel de suivi de bogue

Bugzilla dispose d'un outil permettant de migrer à partir d'autres logiciels de suivi de bogue : `Bugzilla::Migrate`. Il dispose de l'infrastructure dont vous aurez besoin, mais nécessite d'écrire un module pour définir les spécificités du système duquel vous venez. Il en existe un pour *Gnats*. Si vous en écrivez un pour un système populaire, merci de partager votre code avec nous.

Bugzilla fournit également un script, `importxml.pl`, permettant d'importer des bogues dans Bugzilla au format XML. Vous pouvez trouver des exemples en cliquant sur le lien *XML* au bas de la page d'un bogue dans Bugzilla. Vous devrez lire le script pour voir comment il gère les erreurs, les valeurs par défaut, la création de valeurs non existantes, etc.

Utilisez de préférence `Bugzilla::Migrate` si c'est possible.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 3.8 Déplacer Bugzilla d'une machine à une autre

Il est parfois nécessaire de déplacer une installation de Bugzilla sur une autre machine. Cette page explique comment en supposant que le serveur Web et la base de données de Bugzilla sont sur la même machine, et que vous déplacez les deux.

Nous vous conseillons d'installer la même version de Bugzilla sur le nouveau serveur –la *mise à jour* pourra être effectuée ensuite dans une étape séparée. Mais si vous voulez installer en même temps que le transfert une nouvelle version, cela devrait aussi fonctionner.

1. Arrêtez Bugzilla en vous rendant dans la page d'accueil, puis *Administration* | *Paramètres* | *Général* et en ajoutant un texte approprié dans le paramètre `shutdownhtml`.
2. Faites une *sauvegarde* de la base de données.
3. Sur votre nouvelle machine, installez Bugzilla en suivant les instructions sur *Guide d'installation et de maintenance*. Consultez sur l'ancienne machine les valeurs utilisées pour la configuration, par ex. pour MySQL.
4. Copiez le répertoire `data` et le fichier `localconfig` de l'ancienne machine vers la nouvelle.
5. Si quelque chose a changé pour la configuration de la base de données (l'emplacement du serveur, le nom d'utilisateur, le mot de passe, etc.), mettez à jour les variables appropriées dans le fichier `localconfig`.

6. Si la nouvelle URL de votre installation Bugzilla est différente de la précédente, mettez à jour le paramètre `urlbase` dans le fichier `data/params.json` en utilisant un éditeur de texte.
7. Copiez la sauvegarde de la base de données sur votre nouveau serveur.
8. Créez une base de données vide `bugs` sur votre nouveau serveur. Pour MySQL, la commande est celle qui suit :  
**`mysql -u root -p -e "CREATE DATABASE bugs DEFAULT CHARACTER SET utf8 ;"`**
9. Importez votre fichier de sauvegarde dans votre nouvelle base de données `bugs`. À nouveau, pour MySQL :  
**`mysql -u root -p bugs < $BACKUP_FILE_NAME`**  
Si vous obtenez une erreur `Packet too large` ou `MySQL server has gone away`, vous devez ajuster la valeur de `max_allowed_packet` dans votre fichier `my.cnf` (d'habitude `/etc/my.cnf`) pour faire correspondre ou dépasser la valeur configurée dans votre ancienne installation de MySQL.  
Si vous rencontrez *n'importe quelle erreur* pendant cette étape, vous devez trouver pourquoi, supprimer la base de données, la créer à nouveau et la réimporter, comme décrit précédemment.
10. Exécutez `checksetup.pl` pour vous assurez que tout est correct. (À moins d'utiliser une nouvelle version de Bugzilla sur votre nouveau serveur, cela ne devrait pas afficher le moindre changement).  
**`./checksetup.pl`**
11. Activez votre nouvelle installation de Bugzilla en vous rendant sur la page d'accueil du nouveau serveur, puis sur *Administration | Paramètres | Général* et en supprimant le texte dans le paramètre `shutdownhtml`.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 3.9 Mise à jour

Vous pouvez effectuer une mise à jour de Bugzilla à partir de n'importe quelle version vers toute version ultérieure en une seule fois - il n'est pas nécessaire de passer par toutes les versions intermédiaires à moins d'avoir changé de méthode pour obtenir le code.

**Avertissement :** La mise à jour est un processus à sens unique. Vous ne pouvez pas revenir à une version précédente de Bugzilla. Si vous souhaitez le faire, vous devrez restaurer votre système à partir d'une sauvegarde. Pour les données critiques ou les grosses installations, nous vous conseillons de tester la mise à jour sur une copie de votre environnement de production.

Bugzilla utilise le système de contrôle de version Git pour conserver son code. Une installation moderne de Bugzilla consistera à récupérer le code de la dernière version stable à partir de notre dépôt Git. Ceci rend la mise à jour beaucoup plus facile. Si ce n'est pas déjà le cas pour votre installation, consultez *Mettre à jour avec Git*.

Avant Git, nous utilisions Bazaar et encore avant, CVS. Si votre installation de Bugzilla a été faite à partir de l'un de ces systèmes de contrôle de version, vous devrez effectuer la mise à jour en trois étapes :

1. Mettez à jour Bugzilla avec la dernière version stable disponible pour votre série.
2. Migrez vers Git en restant sur exactement la même version.
3. Mettez à jour vers la dernière version de Bugzilla en utilisant les instructions pour *Mettre à jour avec Git*.

Consultez *Migrer à partir de Bazaar* ou *Migrer à partir de CVS* selon votre cas.

Certaines installations de Bugzilla ont été effectuées en téléchargeant une archive du code (fichier « `tar.gz` »). Cependant, les dernières archives contiennent les informations nécessaires pour l'utilisation d'un système de contrôle de version. Vous pourrez donc utiliser les instructions pour Git, Bzr ou CVS.

Si vous ne savez pas à quelle catégorie vous appartenez, pour connaître le système de contrôle de version que votre copie de Bugzilla utilise, recherchez à la racine de votre installation Bugzilla un des répertoires suivants :

- `.git` : votre installation contient les informations pour Git - suivez les instructions *Mettre à jour avec Git*.
- `.bzzr` : votre installation contient les informations pour Bazaar - suivez les instructions *Migrer à partir de Bazaar*.
- CVS : votre installation contient les informations pour CVS - suivez les instructions *Migrer à partir de CVS*.
- Si aucun des répertoires ci-dessus n'est présent : vous avez utilisé pour votre installation une très vieille archive - suivez les instructions *Migrer à partir d'une archive*.

Il est également possible, particulièrement si votre serveur n'est pas paramétré pour accéder à Internet ou ne peut l'être, de procéder à la mise à jour en utilisant une archive. Consultez *Mettre à jour en utilisant une archive*.

Quel que soit le moyen utilisé, vous pourrez avoir besoin d'aide pour *Mettre à jour un Bugzilla personnalisé ou avec des extensions*.

### 3.9.1 Mettre à jour avec Git

Mettre à jour Bugzilla est très simple, et vous pouvez faire la mise à jour à partir de n'importe quelle version vers une version ultérieure en une seule fois - il n'est pas nécessaire de passer par des étapes intermédiaires. Il existe un script appelé `checksetup.pl` fourni dans Bugzilla qui effectuera automatiquement la migration de la base de données.

#### Avant de mettre à jour

Avant de démarrer la mise à jour, il y a quelques étapes importantes à réaliser :

1. Lisez les *Notes de version* de la version vers laquelle vous allez mettre à jour, particulièrement la section *Comment migrer à partir d'une version précédente*.
2. Consultez la page de Contrôle d'intégrité (*Contrôle d'intégrité*) de votre installation avant de mettre à jour. Essayez de corriger tous les avertissements produits sur cette page avant d'aller plus loin ou vous pourriez avoir des problèmes pendant la mise à jour.
3. Faites une sauvegarde de votre base de données Bugzilla. *CECI EST TRÈS IMPORTANT*. Si quelque chose se passe mal pendant la mise à jour, votre installation peut être corrompue et irrécupérable. Avoir une sauvegarde est une sécurité.

#### Si vous avez modifié votre installation Bugzilla

Si vous avez modifié le code ou les templates de votre installation Bugzilla, alors la mise à jour nécessite un peu plus d'effort et de réflexion. Une discussion sur les diverses méthodes de mise à jour en fonction du degré et des méthodes de personnalisation locaux se trouve dans *Choisir une méthode de personnalisation*.

Plus l'écart de version est important, plus il sera difficile de mettre à jour si vous avez fait des personnalisations locales. Une mise à jour d'une version 4.2. vers une version 4.2.1 devrait se faire sans peine, même si vous avez fortement personnalisé votre installation. Mais passer d'une version 2.18 à une version 4.2 un gros travail de ré-écriture de vos changements locaux pour utiliser les nouveaux fichiers, logique, templates, etc. Si vous n'avez pas fait de changement locaux du tout cependant, alors la mise à jour devrait représenter approximativement la même quantité de travail, quelle que soit la version que vous utilisez actuellement.

Si vous avez fait des personnalisations, vous devriez faire la mise à jour sur une copie de votre environnement de production et vous assurez que toutes vos personnalisations fonctionnent encore. Si ce n'est pas le cas, effectuez leur portage et les tests de sorte que tout soit prêt quand vous procéderez à la réelle mise à jour de votre environnement de production.

Vous pouvez vérifier s'il y a des personnalisations locales de code en utilisant la commande :

**git diff**

S'il n'y a pas de résultat, lancez alors la commande :

### git log | head

et vérifiez si le dernier « commit » a été fait par l'équipe de Bugzilla ou par vous. S'il apparaît que celui-ci a été fait par nous, alors vous n'avez pas de personnalisations locales du code.

### Démarrer la mise à jour

1. Fermez votre installation Bugzilla en ajoutant un texte explicatif dans le paramètre shutdownhtml.
2. Effectuez toutes les *sauvegardes* nécessaires. *CECI EST TRÈS IMPORTANT*. Si quelque chose tourne mal pendant la mise à jour, les sauvegardes vous permettront de revenir à une situation stable.

### Télécharger la nouvelle version de Bugzilla

Dans les commandes ci-dessous, \$BUGZILLA\_HOME représente le répertoire dans lequel Bugzilla est installé. En supposant que vous avez suivi les instructions d'installation et que votre version de Bugzilla est un « checkout » d'une branche stable, vous pouvez obtenir la dernière version en exécutant les commandes suivantes :

```
cd $BUGZILLA_HOME
```

#### git pull

Si vous voulez mettre à jour vers une version plus récente de Bugzilla, vous devrez de plus exécuter la commande suivante :

#### git checkout bugzilla-X.X-stable

où X.X est le numéro de version à deux chiffres de la version stable que vous voulez installer (par ex. : 4.4).

---

**Note :** N'essayez pas de revenir à une version antérieure de Bugzilla par ce moyen : cela ne fonctionnera pas.

---

Si vous avez des personnalisations locales, git essaiera de les fusionner. Si cela échoue, vous devrez mettre en place le plan que vous avez envisagé quand vous avez détecté ces personnalisations dans les étapes précédentes.

### Mettre à jour la base de données

Exécutez `checksetup.pl`. Ceci effectuera tout ce qui est nécessaire pour convertir votre base de données et les paramètres pour la nouvelle version.

```
cd $BUGZILLA_HOME
```

```
./checksetup.pl
```

**Avertissement :** Pour certaines mises à jour, exécuter `checksetup.pl` sur de grosses installations (75 000 bogues ou plus) peut prendre beaucoup de temps, et même plusieurs heures, si par exemple les index doivent être reconstruits. Si la durée de l'indisponibilité de votre installation est un problème pour vous, vous pouvez déterminer le temps nécessaire en effectuant la mise à jour sur un serveur de test avec les données de production.

`checksetup.pl` peut aussi indiquer que des modules Perl supplémentaires sont nécessaires, ou des versions plus récentes. Vous devrez les installer soit globalement, soit localement en utilisant le script `install-module.pl`.

### Terminer la mise à jour

1. Réactivez Bugzilla en effaçant le texte saisi dans le paramètre shutdownhtml.

2. Lancez un nouveau *Contrôle d'intégrité* sur votre installation mise à jour. Il est recommandé de corriger tout problème rencontré immédiatement. Ne pas le faire peut entraîner des dysfonctionnements de Bugzilla.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 3.9.2 Migrer à partir de Bazaar

La procédure pour migrer vers Git se fait comme suit. L'idée est de changer de système de contrôle de version sans changer la version de Bugzilla que vous utilisez, pour minimiser les risques de conflits ou de problèmes. Le changement de version peut donc se faire dans une étape séparée.

#### Télécharger le code avec Git

Tout d'abord, vous devez trouver la version de Bugzilla que vous utilisez. Elle devrait se trouver dans le coin supérieur droit de la page d'accueil. Si ce n'est pas le cas, ouvrez le fichier `Bugzilla/Constants.pm` dans votre répertoire Bugzilla et recherchez `BUGZILLA_VERSION`.

Ensuite, vous devrez télécharger une copie supplémentaire de votre version *actuelle* de Bugzilla à partir du dépôt Git, et la placer dans un répertoire séparé à côté de votre installation Bugzilla existante (nous supposons qu'elle se trouve dans un répertoire appelé `bugzilla`).

Pour faire cela, vous aurez besoin d'une copie du programme `git`. Toutes les distributions Linux en ont une. Recherchez dans votre gestionnaire de paquets `git`. Sous Windows ou Mac OS X, vous pouvez [télécharger le binaire officiel](#).

Une fois Git installé, exécutez ces commandes pour récupérer une copie de Bugzilla :

```
git clone https://git.mozilla.org/bugzilla/bugzilla bugzilla-new
```

```
cd bugzilla-new
```

```
git checkout release-$VERSION
```

Remplacez `$VERSION` avec le nombre à trois chiffres de la version de votre installation actuelle de Bugzilla, par ex. `4.2.2`. (Si le dernier chiffre est un 0, omettez-le – utilisez donc `4.4` pour la première version de la série 4.4).

Vous verrez un message indiquant 'detached HEAD'. Ne vous inquiétez pas, votre tête est toujours solidement attachée à vos épaules.

#### Sauvegarder les personnalisations locales

Rendez-vous dans votre répertoire actuel de Bugzilla et exécutez la commande suivante :

```
bzr diff > patch.diff
```

Si vous avez réalisé des personnalisations dans votre installation Bugzilla et que vous l'avez fait en modifiant directement le code de Bugzilla (au lieu d'utiliser le système d'extension), alors le fichier `patch.diff` contiendra beaucoup de contenu. Conserver une copie de ce fichier et des modifications référencées par les lignes `Only in`. Si le fichier est vide ou s'il ne contient que du contenu insignifiant, vous n'avez pas fait de personnalisations de ce genre.

#### Arrêter Bugzilla

Maintenant, vous devez arrêter Bugzilla pour être sûr qu'aucun changement ne survienne pendant la bascule. Rendez-vous dans l'interface d'administration et saisissez un message approprié dans le paramètre `shutdownhtml`, qui se trouve dans la section `Général` des paramètres d'administration. Comme son nom l'indique, le code HTML y est autorisé.

C'est le bon moment pour faire des *Sauvegardes*. Nous ne devrions pas toucher à la base de données, mais on ne saurait être trop prudent.

### Copier les données et les modules

Copiez le contenu des répertoires suivants à partir de votre installation de Bugzilla actuelle vers le répertoire correspondant dans `bugzilla-new/` :

```
lib/  
data/  
template/en/custom (ce répertoire peut ne pas exister)
```

Vous devrez aussi copier toutes les extensions que vous avez écrites ou installées, qui se trouvent dans le répertoire `extensions/`. La commande **`bzr status extensions/`** devrait vous permettre de voir ce que vous avez ajouté le cas échéant.

Enfin, copiez le fichier suivant à partir de votre installation actuelle de Bugzilla dans l'emplacement correspondant dans `bugzilla-new/` :

```
localconfig
```

Ce fichier contient le mot de passe et les détails de l'accès à la base de données. Vos deux versions de Bugzilla étant identiques, cela devrait fonctionner sans problème.

### Réappliquer les personnalisations locales

Si votre fichier `patch.diff` est vide, vous pouvez passer à l'étape suivante. Dans le cas contraire, vous devez appliquer le correctif dans votre nouvelle installation. Si vous êtes sous Windows et que vous n'avez pas le programme **patch**, vous pouvez le télécharger sur [GNUWin](#). Une fois téléchargé, vous devez copier l'exécutable `patch.exe` dans le répertoire Windows.

Copiez `patch.diff` dans le répertoire `bugzilla-new` puis exécutez la commande suivante :

```
patch -p0 -dry-run < patch.diff
```

Le correctif devrait s'appliquer correctement car vous avez exactement la même version de Bugzilla dans les deux répertoires. Si c'est le cas, retirez l'argument **`-dry-run`** de la ligne de commande et relancez la commande pour appliquer les modifications. Si ce n'est pas le cas, c'est que vous avez des versions de Bugzilla différentes dans les deux répertoires.

### Basculer vers la nouvelle version

Maintenant, intervertissez les répertoires et exécutez `checksetup.pl` pour confirmer que tout va bien. À partir du répertoire contenant les répertoires `bugzilla` et `bugzilla-new`, exécutez les commandes suivantes :

```
mv bugzilla bugzilla-old
```

```
mv bugzilla-new bugzilla
```

```
cd bugzilla
```

```
./checksetup.pl
```

L'exécution de `checksetup.pl` ne devrait pas modifier votre base de données. Si c'est le cas, c'est que vous avez deux versions de Bugzilla différentes.



## Réactiver Bugzilla

Rendez-vous dans l'interface d'administration et supprimez le contenu du paramètre shutdownhtml.

## Tester Bugzilla

Utilisez Bugzilla pendant plusieurs jours pour vérifier que la bascule n'a pas eu d'effet de bord. Ensuite, si nécessaire, suivez les instructions dans *Mettre à jour avec Git* pour mettre à jour vers la dernière version de Bugzilla.

## Revenir en arrière

Si quelque chose s'est mal passé pendant le processus (par ex. votre correctif ne s'est pas appliqué ou `checksetup.pl` renvoie des erreurs), vous pouvez toujours réintervenir les répertoires (si vous êtes arrivé jusque là) et réactiver Bugzilla (si vous l'aviez désactivé) et recherchez de l'aide. Même si vous avez réactivé Bugzilla, et que vous rencontrez des problèmes peu de temps après, vous utilisez toujours la même version, donc il ne devrait pas y avoir pour revenir en arrière deux à trois jours après.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 3.9.3 Migrer à partir de CVS

La procédure pour migrer vers Git se fait comme suit. L'idée est de changer de système de contrôle de version sans changer la version de Bugzilla que vous utilisez, pour minimiser les risques de conflits ou de problèmes. Le changement de version peut donc se faire dans une étape séparée.

#### Télécharger le code avec Git

Tout d'abord, vous devez trouver la version de Bugzilla que vous utilisez. Elle devrait se trouver dans le coin supérieur droit de la page d'accueil. Si ce n'est pas le cas, ouvrez le fichier `Bugzilla/Constants.pm` dans votre répertoire Bugzilla et recherchez `BUGZILLA_VERSION`.

Ensuite, vous devrez télécharger une copie supplémentaire de votre version *actuelle* de Bugzilla à partir du dépôt Git, et la placer dans un répertoire séparé à côté de votre installation Bugzilla existante (nous supposons qu'elle se trouve dans un répertoire appelé `bugzilla`).

Pour faire cela, vous aurez besoin d'une copie du programme `git`. Toutes les distributions Linux en ont une. Recherchez dans votre gestionnaire de paquets `git`. Sous Windows ou Mac OS X, vous pouvez [télécharger le binaire officiel](#).

Une fois Git installé, exécutez ces commandes pour récupérer une copie de Bugzilla :

```
git clone https://git.mozilla.org/bugzilla/bugzilla bugzilla-new
```

```
cd bugzilla-new
```

```
git checkout release-$VERSION
```

Remplacez `$VERSION` avec le nombre à trois chiffres de la version de votre installation actuelle de Bugzilla, par ex. `4.2.2`. (Si le dernier chiffre est un 0, omettez-le – utilisez donc `4.4` pour la première version de la série 4.4).

Vous verrez un message indiquant 'detached HEAD'. Ne vous inquiétez pas, votre tête est toujours solidement attachée à vos épaules.

### Sauvegarder les personnalisations locales

Rendez-vous dans votre répertoire actuel de Bugzilla et exécutez la commande suivante :

```
cvs diff -puN > patch.diff
```

Si vous avez réalisé des personnalisations dans votre installation Bugzilla et que vous l'avez fait en modifiant directement le code de Bugzilla (au lieu d'utiliser le système d'extension), alors le fichier `patch.diff` contiendra beaucoup de contenu. Conserver une copie de ce fichier et des modifications référencées par les lignes `Only in`. Si le fichier est vide ou s'il ne contient que du contenu insignifiant, vous n'avez pas fait de personnalisations de ce genre.

### Arrêter Bugzilla

Maintenant, vous devez arrêter Bugzilla pour être sûr qu'aucun changement ne survienne pendant la bascule. Rendez-vous dans l'interface d'administration et saisissez un message approprié dans le paramètre `shutdownhtml`, qui se trouve dans la section `Général` des paramètres d'administration. Comme son nom l'indique, le code HTML y est autorisé.

C'est le bon moment pour faire des *Sauvegardes*. Nous ne devrions pas toucher à la base de données, mais on ne saurait être trop prudent.

### Copier les données et les modules

Copiez le contenu des répertoires suivants à partir de votre installation de Bugzilla actuelle vers le répertoire correspondant dans `bugzilla-new/` :

```
lib/  
data/  
template/en/custom (ce répertoire peut ne pas exister)
```

Vous devrez aussi copier toutes les extensions que vous avez écrites ou installées, qui se trouvent dans le répertoire `extensions/`. La commande `cvs status extensions/` devrait vous permettre de voir ce que vous avez ajouté le cas échéant.

Enfin, copiez le fichier suivant à partir de votre installation actuelle de Bugzilla dans l'emplacement correspondant dans `bugzilla-new/` :

```
localconfig
```

Ce fichier contient le mot de passe et les détails de l'accès à la base de données. Vos deux versions de Bugzilla étant identiques, cela devrait fonctionner sans problème.

### Réappliquer les personnalisations locales

Si votre fichier `patch.diff` est vide, vous pouvez passer à l'étape suivante. dans le cas contraire, vous devez appliquer le correctif dans votre nouvelle installation. Si vous êtes sous Windows et que vous n'avez pas le programme **patch**, vous pouvez le télécharger sur [GNUWin](#). Une fois téléchargé, vous devez copier l'exécutable `patch.exe` dans le répertoire Windows.

Copiez `patch.diff` dans le répertoire `bugzilla-new` puis exécutez la commande suivante :

```
patch -p0 -dry-run < patch.diff
```

Le correctif devrait s'appliquer correctement car vous avez exactement la même version de Bugzilla dans les deux répertoires. Si c'est le cas, retirez l'argument `-dry-run` de la ligne de commande et relancez la commande pour appliquer les modifications. Si ce n'est pas le cas, c'est que vous avez des versions de Bugzilla différentes dans les deux répertoires.

## Basculer vers la nouvelle version

Maintenant, intervertissez les répertoires et exécutez `checksetup.pl` pour confirmer que tout va bien. À partir du répertoire contenant les répertoires `bugzilla` et `bugzilla-new`, exécutez les commandes suivantes :

```
mv bugzilla bugzilla-old
```

```
mv bugzilla-new bugzilla
```

```
cd bugzilla
```

```
./checksetup.pl
```

L'exécution de `checksetup.pl` ne devrait pas modifier votre base de données. Si c'est le cas, c'est que vous avez deux versions de Bugzilla différentes.

## Réactiver Bugzilla

Rendez-vous dans l'interface d'administration et supprimez le contenu du paramètre `shutdownhtml`.

## Tester Bugzilla

Utilisez Bugzilla pendant plusieurs jours pour vérifier que la bascule n'a pas eu d'effet de bord. Ensuite, si nécessaire, suivez les instructions dans [Mettre à jour avec Git](#) pour mettre à jour vers la dernière version de Bugzilla.

## Revenir en arrière

Si quelque chose s'est mal passé pendant le processus (par ex. votre correctif ne s'est pas appliqué ou `checksetup.pl` renvoie des erreurs), vous pouvez toujours réintervenir les répertoires (si vous êtes arrivé jusque là) et réactiver Bugzilla (si vous l'aviez désactivé) et recherchez de l'aide. Même si vous avez réactivé Bugzilla, et que vous rencontrez des problèmes peu de temps après, vous utilisez toujours la même version, donc il ne devrait pas y avoir pour revenir en arrière deux à trois jours après.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 3.9.4 Migrer à partir d'une archive

La procédure pour migrer vers Git est comme suit. L'idée est de basculer vers Git sans changer la version de Bugzilla que vous utilisez pour minimiser les risques de conflits ou de problèmes. La mise à jour est effectuée ensuite dans une autre étape.

### Télécharger le code avec Git

Tout d'abord, vous devez trouver la version de Bugzilla que vous utilisez. Elle devrait se trouver dans le coin supérieur droit de la page d'accueil. Si ce n'est pas le cas, ouvrez le fichier `Bugzilla/Constants.pm` dans votre répertoire Bugzilla et recherchez `BUGZILLA_VERSION`.

Ensuite, vous devrez télécharger une copie supplémentaire de votre version *actuelle* de Bugzilla à partir du dépôt Git, et la placer dans un répertoire séparé à côté de votre installation Bugzilla existante (nous supposons qu'elle se trouve dans un répertoire appelé `bugzilla`).

Pour faire cela, vous aurez besoin d'une copie du programme `git`. Toutes les distributions Linux en ont une. Recherchez dans votre gestionnaire de paquets `git`. Sous Windows ou Mac OS X, vous pouvez [télécharger le binaire officiel](#).

Une fois Git installé, exécutez ces commandes pour récupérer une copie de Bugzilla :

```
git clone https://git.mozilla.org/bugzilla/bugzilla bugzilla-new
```

```
cd bugzilla-new
```

```
git checkout release-$VERSION
```

Remplacez \$VERSION avec le nombre à trois chiffres de la version de votre installation actuelle de Bugzilla, par ex. 4.2.2. (Si le dernier chiffre est un 0, omettez-le – utilisez donc 4.4 pour la première version de la série 4.4).

Vous verrez un message indiquant ‘detached HEAD’. Ne vous inquiétez pas, votre tête est toujours solidement attachée à vos épaules.

### Sauvegarder les personnalisations locales

Rendez-vous dans votre répertoire actuel de Bugzilla et exécutez la commande suivante :

```
diff -ru -x data -x lib -x docs -x .git -x CVS -x .cvsignore -x .bzr -x .bzrignore -x .bzrrev ../bugzilla-new . > ./patch.diff
```

Si vous avez réalisé des personnalisations dans votre installation Bugzilla et que vous l’avez fait en modifiant directement le code de Bugzilla (au lieu d’utiliser le système d’extension), alors le fichier `patch.diff` contiendra beaucoup de contenu. Conserver une copie de ce fichier et des modifications référencées par les lignes `Only in`. Si le fichier est vide ou s’il ne contient que du contenu insignifiant, vous n’avez pas fait de personnalisations de ce genre.

### Arrêter Bugzilla

Maintenant, vous devez arrêter Bugzilla pour être sûr qu’aucun changement ne survienne pendant la bascule. Rendez-vous dans l’interface d’administration et saisissez un message approprié dans le paramètre `shutdownhtml`, qui se trouve dans la section `Général` des paramètres d’administration. Comme son nom l’indique, le code HTML y est autorisé.

C’est le bon moment pour faire des *Sauvegardes*. Nous ne devrions pas toucher à la base de données, mais on ne saurait être trop prudent.

### Copier les données et les modules

Copiez le contenu des répertoires suivants à partir de votre installation de Bugzilla actuelle vers le répertoire correspondant dans `bugzilla-new/` :

```
lib/  
data/  
template/en/custom (ce répertoire peut ne pas exister)
```

Vous devrez aussi copier toutes les extensions que vous avez écrites ou installées, qui se trouvent dans le répertoire `extensions/`. Recopiez tout sous-répertoire qui n’existe pas dans votre nouvelle installation.

Enfin, copiez le fichier suivant à partir de votre installation actuelle de Bugzilla dans l’emplacement correspondant dans `bugzilla-new/` :

```
localconfig
```

Ce fichier contient le mot de passe et les détails de l’accès à la base de données. Vos deux versions de Bugzilla étant identiques, cela devrait fonctionner sans problème.

## Réappliquer les personnalisations locales

Si votre fichier `patch.diff` est vide, vous pouvez passer à l'étape suivante. Dans le cas contraire, vous devez appliquer le correctif dans votre nouvelle installation. Si vous êtes sous Windows et que vous n'avez pas le programme **patch**, vous pouvez le télécharger sur [GNUWin](#). Une fois téléchargé, vous devez copier l'exécutable `patch.exe` dans le répertoire Windows.

Copiez `patch.diff` dans le répertoire `bugzilla-new` puis exécutez la commande suivante :

```
patch -p0 --dry-run < patch.diff
```

Le correctif devrait s'appliquer correctement car vous avez exactement la même version de Bugzilla dans les deux répertoires. Si c'est le cas, retirez l'argument **--dry-run** de la ligne de commande et relancez la commande pour appliquer les modifications. Si ce n'est pas le cas, c'est que vous avez des versions de Bugzilla différentes dans les deux répertoires.

## Basculer vers la nouvelle version

Maintenant, intervertissez les répertoires et exécutez `checksetup.pl` pour confirmer que tout va bien. À partir du répertoire contenant les répertoires `bugzilla` et `bugzilla-new`, exécutez les commandes suivantes :

```
mv bugzilla bugzilla-old
```

```
mv bugzilla-new bugzilla
```

```
cd bugzilla
```

```
./checksetup.pl
```

L'exécution de `checksetup.pl` ne devrait pas modifier votre base de données. Si c'est le cas, c'est que vous avez deux versions de Bugzilla différentes.

## Réactiver Bugzilla

Rendez-vous dans l'interface d'administration et supprimez le contenu du paramètre `shutdownhtml`.

## Tester Bugzilla

Utilisez Bugzilla pendant plusieurs jours pour vérifier que la bascule n'a pas eu d'effet de bord. Ensuite, si nécessaire, suivez les instructions dans [Mettre à jour avec Git](#) pour mettre à jour vers la dernière version de Bugzilla.

## Revenir en arrière

Si quelque chose s'est mal passé pendant le processus (par ex. votre correctif ne s'est pas appliqué ou `checksetup.pl` renvoie des erreurs), vous pouvez toujours réintervenir les répertoires (si vous êtes arrivé jusque là) et réactiver Bugzilla (si vous l'aviez désactivé) et recherchez de l'aide. Même si vous avez réactivé Bugzilla, et que vous rencontrez des problèmes peu de temps après, vous utilisez toujours la même version, donc il ne devrait pas y avoir pour revenir en arrière deux à trois jours après.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 3.9.5 Mettre à jour en utilisant une archive

Si vous ne pouvez (ou ne voulez) pas utiliser Git, une autre option est toujours disponible pour obtenir la dernière archive à partir de la [Page de téléchargements](#) pour créer une nouvelle installation de Bugzilla à partir de celle-ci.

Sans système de contrôle de version pour vous aider, le processus peut être un peu plus compliqué.

#### Avant de mettre à jour

Avant de démarrer la mise à jour, il y a quelques étapes importantes à réaliser :

1. Lisez les [Notes de version](#) de la version vers laquelle vous allez mettre à jour, particulièrement la section Comment migrer à partir d'une version précédente.
2. Consultez la page de Contrôle d'intégrité (*Contrôle d'intégrité*) de votre installation avant de mettre à jour. Essayez de corriger tous les avertissements produits sur cette page avant d'aller plus loin ou vous pourriez avoir des problèmes pendant la mise à jour.
3. Faites une sauvegarde de votre base de données Bugzilla. **CECI EST TRÈS IMPORTANT**. Si quelque chose se passe mal pendant la mise à jour, votre installation peut être corrompue et irrécupérable. Avoir une sauvegarde est une sécurité.

#### Si vous avez modifié votre installation Bugzilla

Si vous avez modifié le code ou les templates de votre installation Bugzilla, alors la mise à jour nécessite un peu plus d'effort et de réflexion. Une discussion sur les diverses méthodes de mise à jour en fonction du degré et des méthodes de personnalisation locaux se trouve dans *Choisir une méthode de personnalisation*.

Plus l'écart de version est important, plus il sera difficile de mettre à jour si vous avez fait des personnalisations locales. Une mise à jour d'une version 4.2. vers une version 4.2.1 devrait se faire sans peine, même si vous avez fortement personnalisé votre installation. Mais passer d'une version 2.18 à une version 4.2 un gros travail de ré-écriture de vos changements locaux pour utiliser les nouveaux fichiers, logique, templates, etc. Si vous n'avez pas fait de changement locaux du tout cependant, alors la mise à jour devrait représenter approximativement la même quantité de travail, quelle que soit la version que vous utilisez actuellement.

Si vous avez fait des personnalisations, vous devriez faire la mise à jour sur une copie de votre environnement de production et vous assurez que toutes vos personnalisations fonctionnent encore. Si ce n'est pas le cas, effectuez leur portage et les tests de sorte que tout soit prêt quand vous procéderez à la réelle mise à jour de votre environnement de production.

Comme vous utilisez une archive et pas un DVCS, il n'est pas facile de savoir s'il y a des personnalisations de code dans votre installation. Vous pouvez demander au sein de votre organisation si c'est le cas ou télécharger une copie de la *même* version de Bugzilla que vous utilisez et comparer les deux répertoires. Si vous avez des personnalisations du code, vous devrez en faire un correctif, peut-être en effectuant un `diff` des deux répertoires, et en réappliquant plus tard ce correctif. Si vous personnalisez votre Bugzilla localement, envisager le « *rebasing* » de votre installation avec *git*.

#### Télécharger la nouvelle version de Bugzilla

Téléchargez une copie de la dernière version de Bugzilla sur la [page de téléchargements](#) dans un répertoire séparé (que nous appellerons `bugzilla-new`) à côté de votre installation existante (que nous supposerons installé dans un répertoire appelé `bugzilla`).

## Recopier les données et les modules

Copiez le contenu des répertoires suivants de votre installation Bugzilla actuelle vers les répertoires correspondants dans `bugzilla-new/` :

```
lib/  
data/  
template/en/custom (peut ne pas exister)
```

Vous devrez aussi copier les extensions que vous avez écrites ou installées, qui se trouvent dans le répertoire `extensions/`. Bugzilla fournit quelques extensions, donc si vous voulez savoir si certaines extensions installées sont les vôtres, vous pouvez faire une comparaison avec une copie vierge de votre version actuelle. Vous pouvez ignorer les extensions contenant un fichier `disabled` dans leur répertoire - celles-ci ne sont pas activées.

Enfin, copiez le fichier suivant à partir de votre installation actuelle vers le répertoire correspondant dans `bugzilla-new/` :

```
localconfig
```

Ce fichier contient les informations de connexion à votre base de données.

## Basculer vers la nouvelle version

Maintenant, intervertissez les répertoires. À partir du répertoire contenant les répertoires `bugzilla` et `bugzilla-new`, exécutez les commandes suivantes :

```
mv bugzilla bugzilla-old
```

```
mv bugzilla-new bugzilla
```

```
cd bugzilla
```

## Mettre à jour la base de données

Exécutez `checksetup.pl`. Ceci effectuera tout ce qui est nécessaire pour convertir votre base de données et les paramètres pour la nouvelle version.

```
cd $BUGZILLA_HOME
```

```
./checksetup.pl
```

**Avertissement :** Pour certaines mises à jour, exécuter `checksetup.pl` sur de grosses installations (75 000 bogues ou plus) peut prendre beaucoup de temps, et même plusieurs heures, si par exemple les index doivent être reconstruits. Si la durée de l'indisponibilité de votre installation est un problème pour vous, vous pouvez déterminer le temps nécessaire en effectuant la mise à jour sur un serveur de test avec les données de production.

`checksetup.pl` peut aussi indiquer que des modules Perl supplémentaires sont nécessaires, ou des versions plus récentes. Vous devrez les installer soit globalement, soit localement en utilisant le script `install-module.pl`.

## Terminer la mise à jour

1. Réactivez Bugzilla en effaçant le texte saisi dans le paramètre `shutdownhtml`.
2. Lancez un nouveau *Contrôle d'intégrité* sur votre installation mise à jour. Il est recommandé de corriger tout problème rencontré immédiatement. Ne pas le faire peut entraîner des dysfonctionnements de Bugzilla.

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 3.9.6 Mettre à jour un Bugzilla personnalisé ou avec des extensions

Si votre installation de Bugzilla a été personnalisée ou utilise des extensions, vous devrez adapter vos personnalisations ou extensions pour votre nouvelle version Bugzilla. Si c'est le cas, nous vous recommandons vivement de faire un test de la mise à jour sur un environnement de test.

Si votre extension provient d'une tierce partie, vérifiez si une version mise à jour est disponible pour la version de Bugzilla que vous voulez utiliser. Si ce n'est pas le cas, et que vous voulez continuer à l'utiliser, vous devrez effectuer le portage vous-même.

Si vous procédez à la mise à jour de Bugzilla à partir d'une version antérieure à la version 3.6 et que vous avez des extensions pour lesquelles aucune nouvelle version n'est disponible, vous devrez les convertir. Ceci s'explique par le fait que le format des extensions a changé dans la version 3.6. Il existe un fichier appelé `extension-convert.pl` dans le répertoire `contrib` qui peut vous aider dans cette tâche.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

Bugzilla peut notifier automatiquement aux administrateurs quand de nouvelles versions sont disponibles si le paramètre `upgrade_notification` est défini. Les administrateurs verront ces notifications lors de leur connexion à Bugzilla sur la page d'accueil. Bugzilla vérifie une fois par jour la présence de nouvelles versions. Si vous êtes derrière un proxy, vous devrez définir le paramètre `proxy_url` en conséquence. Si le proxy nécessite une authentification, utilisez la syntaxe `http://utilisateur:mot_de_passe@url_du_proxy/`.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 3.10 Sauvegardes

### 3.10.1 Base de données

Voici quelques exemples de commandes à utiliser pour sauvegarder votre base de données, en fonction du système de base de données que vous utilisez. Vous devrez peut-être modifier ces commandes selon votre configuration spécifique. Remplacez les \$VARIABLES par les valeurs appropriées pour votre installation.

#### MySQL

```
mysqldump --max-allowed-packet=32M -u $USERNAME -p $DATABASENAME > backup.sql
```

La valeur pour **max-allowed-packet** doit être la valeur que vous avez définie dans votre *fichier de configuration MySQL*, et doit être supérieure à celle du plus gros fichier joint dans votre base de données. Consulter la [documentation de mysqldump](#) pour plus d'information sur `mysqldump`.

#### PostgreSQL

```
pg_dump --no-privileges --no-owner -h localhost -U $USERNAME > bugs.sql
```



---

## 3.10.2 Bugzilla

Le répertoire Bugzilla contient certains fichiers de données et de configuration que vous voudrez sauvegarder. Une simple copie récursive du répertoire suffit.

```
cp -rp $BUGZILLA_HOME /var/backups/bugzilla
```

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 3.11 Contrôle d'intégrité

Avec le temps, il est possible que la base de données de Bugzilla devienne corrompue ou contienne des anomalies. Ceci peut survenir lors de manipulations sur la base de données effectuées en dehors de l'interface utilisateur de Bugzilla ou d'autres événements inattendus. Bugzilla contient un Contrôle d'intégrité qui réalise des vérifications basiques de la base de données et répare certains problèmes ou incohérences.

Pour exécuter le contrôle d'intégrité, connectez-vous en tant qu'administrateur et cliquez sur le lien *contrôle d'intégrité* dans la page d'administration. Tout problème identifié sera affiché en lettres rouges. Si le script n'est pas capable de corriger un problème, il présentera un lien pour la correction du problème. Cela peut nécessiter une intervention manuelle sur la base de données ou une restauration.

Le Contrôle d'intégrité peut aussi être exécuté en ligne de commande avec le script Perl `sanitycheck.pl`. Ce script peut aussi être exécuté régulièrement à l'aide d'une tâche programmée **cron**. Les résultats seront envoyés par courriel à l'adresse électronique indiquée dans la ligne de commande.

Nous vous recommandons d'exécuter régulièrement un contrôle d'intégrité.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 3.12 Fusionner les comptes

Parfois, les utilisateurs créent un second compte, peut-être parce qu'ils ne savent pas qu'ils peuvent changer l'adresse électronique associée à leur compte. Quand ils le découvrent, ils ne veulent pas abandonner l'historique de l'un ou l'autre des comptes.

Le meilleur moyen dans ce cas sera de fusionner les comptes de sorte que le compte choisi centralise l'historique pour les deux comptes. Dans le répertoire Bugzilla `contrib`, il y a un script appelé `merge-users.pl`. Bien que le code dans ce répertoire ne soit pas officiellement maintenu par l'équipe Bugzilla, ce script peut être utile pour résoudre ce genre de problème.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 3.13 Une installation, plusieurs instances

Ceci est destiné aux spécialistes. Si vous ne savez pas si vous en avez besoin, c'est que vous n'en avez pas besoin. Ceci est utile pour les administrateurs qui voudraient exécuter plusieurs instances distinctes de Bugzilla en utilisant une seule installation du code.

Ceci est possible en utilisant la variable d'environnement `PROJECT`. Quand il est accédé, Bugzilla vérifie l'existence de cette variable, et si elle est présente, utilise sa valeur pour vérifier la présence d'un fichier de configuration alternatif appelé `localconfig.<PROJECT>` au même emplacement que celui par défaut (`localconfig`). Il vérifie aussi la présence de modèles personnalisés dans le répertoire nommé `<PROJECT>` au même emplacement que celui par défaut (`template/<langcode>`). Par défaut, c'est le répertoire `template/en/default` donc les modèles de `PROJECT` se trouveraient dans `template/en/PROJECT`.

Pour définir une installation alternative, exporter la variable `PROJECT=toto` avant de lancer **checksetup.pl** pour la première fois. Il en résultera un fichier nommé `localconfig.toto` au lieu de `localconfig`. Modifiez ce fichier comme décrit plus haut, avec la référence à une nouvelle base de données, et relancez **checksetup.pl** pour la populer. C'est tout.

Maintenant, vous devez paramétrer le serveur Web pour lui passer cette variable d'environnement quand il est accédé via une URL alternative, comme un hôte virtuel par exemple. Ce qui suit est un exemple de ce que vous pouvez faire avec Apache, cela peut différer pour les autres serveurs Web.

```
<VirtualHost 12.34.56.78:80>
    ServerName bugzilla.example.com
    SetEnv PROJECT toto
</VirtualHost>
```

N'oubliez pas aussi d'exporter cette variable avant d'accéder à Bugzilla par d'autres voies, comme les tâches programmées de `cron` par exemple.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

---

## Guide d'administration

---

Pour les utilisateurs ayant les privilèges admin, Bugzilla peut être administré en utilisant le lien *Administration* dans l'en-tête. Les contrôles d'administration sont divisés en plusieurs sections :

### 4.1 Paramètres

Bugzilla se configure en changeant divers paramètres accessibles à partir du lien *Paramètres* de la page *Administration* (la page *Administration* est accessible en cliquant sur le lien *Administration* dans le pied de page). Les paramètres sont divisés en plusieurs catégories, accessibles par le menu à gauche.

#### 4.1.1 Paramètres requis

Les paramètres principaux obligatoires pour une installation de Bugzilla sont définis ici. Ces paramètres doivent être définis avant qu'une nouvelle installation de Bugzilla soit fonctionnelle. Les administrateurs doivent revoir cette liste avant de déployer une nouvelle installation de Bugzilla.

**urlbase** Définit le nom de domaine complet et le chemin d'accès du serveur Web de cette installation de Bugzilla.

Par exemple, si la page de recherche est `http://www.toto.com/bugzilla/query.cgi`, `urlbase` doit être défini à `http://www.toto.com/bugzilla/`.

**ssl\_redirect** Quand ceci est activé, Bugzilla forcera des connexions HTTPS (SSL), en redirigeant automatiquement tout utilisateur essayant d'utiliser une connexion non-SSL. Bugzilla enverra alors aussi les liens en utilisant `sslbase` dans les courriels au lieu de `urlbase`.

**sslbase** Définit le nom de domaine complet et le chemin d'accès au serveur Web pour les connexions HTTPS (SSL) de cette installation de Bugzilla. Par exemple, si la page principale de Bugzilla est `https://www.toto.com/bugzilla/index.cgi`, `sslbase` doit être défini à `https://www.toto.com/bugzilla/`.

**cookiepath** Définit un chemin, relatif à la racine du serveur Web, auquel seront restreints les cookies de Bugzilla.

Par exemple, si `urlbase` est défini à `http://www.toto.com/bugzilla/`, `cookiepath` devrait être défini à `/bugzilla/`. Définir ceci à `/` permettra à tous les sites servis par ce serveur Web ou cet hôte virtuel de lire les cookies de Bugzilla.

#### 4.1.2 Général

**maintainer** Adresse électronique de la personne responsable de la maintenance de cette installation de Bugzilla. L'adresse n'est pas nécessairement celle d'un compte Bugzilla valide.

**utf8** Détermine l'utilisation de l'encodage UTF-8 (Unicode) pour tout texte dans Bugzilla. Les nouvelles installations devraient définir ce paramètre à `Activé` pour éviter les problèmes d'encodage de caractères. Les bases de données existantes ne devraient définir ceci à `Activé` qu'après que les données aient été converties de l'encodage existant vers UTF-8, en utilisant le script `contrib/recode.pl`.

---

**Note :** Si vous passez ce paramètre de `Désactivé` à `Activé`, vous devez ré-exécuter `checksetup.pl` immédiatement après.

---

**shutdownhtml** S'il y a du texte dans ce champ, cette installation de Bugzilla sera totalement désactivée et ce texte apparaîtra à la place de toutes les pages de Bugzilla pour tous les utilisateurs, y compris les administrateurs. À utiliser dans le cadre d'une maintenance du site ou de problèmes.

**announcehtml** Tout texte dans ce champ sera affiché en haut de chaque page HTML de cette installation de Bugzilla. Ce texte n'est pas encadré dans des balises. Pour de meilleurs résultats, encadrez le texte avec des balises `<div>`. tout attribut de style de la CSS peut être appliqué. Par exemple, pour mettre le texte en vert dans une boîte rouge, ajoutez `id=message` à la balise `<div>`.

**upgrade\_notification** Active ou désactive la notification sur la page d'accueil de cette installation de Bugzilla quand une nouvelle version de Bugzilla est disponible. Cette notification n'est visible que des administrateurs. Choisissez `disabled`, pour désactiver la notification. Sinon, choisissez pour quelles versions de Bugzilla vous voulez être prévenu : `development_snapshot` est la dernière version du tronc ; `latest_stable_release` est la dernière version disponible sur la branch stable la plus récente ; `stable_branch_release` est la version la plus récente de la branche sur laquelle est basée cette installation.

### 4.1.3 Politiques d'administration

Cette page contient les paramètres pour les fonctions administratives de base. Les options comprennent l'autorisation de la suppression de bogues et d'utilisateurs et l'autorisation pour les utilisateurs de modifier leur adresse électronique.

**allowbugdeletion** Les pages pour modifier les produits et les composants peuvent supprimer tous les bogues associés lors de la suppression d'un produit ou d'un composant. Cela peut être dangereux, c'est pourquoi vous devez explicitement activer cette option ici pour que de telles suppressions surviennent.

**allowemailchange** Les utilisateurs peuvent changer leur propre adresse électronique dans leurs préférences. Notez que le changement est validé en envoyant un courriel aux deux adresses. Activer cette option empêchera les utilisateurs d'utiliser des adresses invalides.

**allowuserdeletion** Les pages d'édition des utilisateurs permettent de supprimer des comptes d'utilisateurs. Bugzilla affichera un avertissement en cas d'incohérences lors de cette action, cependant, de telles suppressions demeurent dangereuses. C'est pourquoi vous devez activer cette option ici avant de pouvoir réaliser de telles suppressions.

**last\_visit\_keep\_days** Cette option permet de définir le nombre de jours pendant lequel Bugzilla conservera les visites des utilisateurs sur des bogues spécifiques.

### 4.1.4 Authentification utilisateur

Cette page contient les paramètres qui contrôlent la façon dont cette installation de Bugzilla fera l'authentification. Choisissez le mécanisme d'authentification à utiliser (la base de données de Bugzilla, ou une source externe comme un serveur LDAP), et définissez les paramètres de base. Par exemple, choisissez si les utilisateurs doivent s'authentifier pour parcourir les bogues, la gestion des cookies d'authentification, et les expressions régulières utilisées pour valider les adresses électroniques. Certains paramètres sont soulignés ci-dessous.

**auth\_env\_id** Variable d'environnement utilisé par un système d'authentification externe pour stocker un identifiant unique à chaque utilisateur. Laisser vide s'il n'y en a pas ou si cette méthode d'authentification n'est pas utilisée.

- auth\_env\_email** Variable d'environnement utilisé par un système d'authentification externe pour stocker l'adresse électronique de chaque utilisateur. C'est un champ obligatoire pour l'authentification par l'environnement. Laisser vide si vous n'utilisez pas cette fonctionnalité.
- auth\_env\_realname** Variable d'environnement utilisé par un système d'authentification externe pour stocker le nom réel de chaque utilisateur. Laisser vide s'il n'y en a pas ou si cette méthode d'authentification n'est pas utilisée.
- user\_info\_class** Mécanisme(s) à utiliser pour rassembler les informations de connexion d'un utilisateur. Plusieurs peuvent être sélectionnés. si le premier ne renvoie rien, le second est essayé, ainsi de suite. Les types sont :
- CGI : demande le nom d'utilisateur et le mot de passe via le formulaire CGI de l'interface.
  - Env : les informations pour un utilisateur pré-authentifié est passé dans les variables de l'environnement système.
- user\_verify\_class** Mécanisme(s) à utiliser pour valider (authentifier) les informations rassemblées par `user_info_class`. Plusieurs peuvent être sélectionnés. Si le premier ne peut trouver l'utilisateur, le second est essayé, ainsi de suite. Les types sont :
- DB : authentification intégrée de Bugzilla. C'est le choix le plus courant.
  - RADIUS : authentification RADIUS utilisant un serveur RADIUS. Utiliser cette méthode nécessite des paramètres supplémentaires. Veuillez consulter [RADIUS](#) pour plus de détails.
  - LDAP : authentification LDAP utilisant un serveur LDAP. Utiliser cette méthode nécessite des paramètres supplémentaires. Veuillez consulter [LDAP](#) pour plus de détails.
- rememberlogin** Contrôle la gestion des cookies de session.
- on - Les cookies de session n'expirent jamais (l'utilisateur doit se connecter une fois pour chaque navigateur).
  - off - Les cookies de session expirent à la fin de la session de l'utilisateur (l'utilisateur devra se connecter pour chaque nouvelle session de navigateur).
  - defaulton/defaultoff - Le comportement par défaut comme décrit ci-dessus, mais l'utilisateur peut choisir si Bugzilla retiendra ses identifiants de connexion ou pas.
- requirelogin** Si cette option est définie, tous les accès autres que celui de la page d'accueil nécessiteront une connexion. Les utilisateurs anonymes ne seront pas autorisés.
- webservice\_email\_filter** Filtre des adresses électroniques renvoyé par l'API de webservice, dépendant de la connexion ou non connexion de l'utilisateur. Ceci fonctionne de manière similaire à l'interface Web utilisateur qui filtre les adresses électroniques. Si « requirelogin » est activé, alors ce paramètre n'aura aucun effet car les utilisateurs doivent être connectés pour utiliser Bugzilla de toutes façons.
- emailregexp** Définit l'expression régulière utilisée pour valider les adresses électroniques utilisées pour les noms de connexion. Par défaut, une correspondance totale est recherchée (par ex. `utilisateur@exemple.com`) d'une façon légèrement plus restrictive que ce qui est autorisé dans la RFC 2822. Certaines installations de Bugzilla n'autorisent que des noms d'utilisateurs locaux (par ex. `utilisateur` au lieu de `utilisateur@exemple.com`). Dans ce cas, ce paramètre doit être utilisé pour définir le domaine des adresses électroniques.
- emailregexpdesc** Cette description est affichée à l'utilisateur pour expliquer quelles adresses électroniques sont autorisées par le paramètre `emailregexp`.
- emailsuffix** Cette chaîne est ajoutée aux noms de connexion lors de l'envoi d'un courriel à un utilisateur. Par exemple, si **emailregexp** a été défini pour permettre les noms d'utilisateurs locaux, alors ce paramètre doit contenir le domaine des adresses électroniques pour tous les utilisateurs (par ex. `@exemple.com`).
- createemailregexp** Ceci définit l'expression régulière (sensible à la casse) à utiliser pour les adresses électroniques autorisées à s'auto-enregistrer. L'expression régulière par défaut `(.*)` permet à tout compte correspondant à « emailregexp » d'être créée. Si ce paramètre est laissé vide, aucun utilisateur ne sera autorisé à créer son compte et tous les comptes devront être créés par un administrateur.
- password\_complexity** Définit la complexité requise pour les mots de passe. Dans tous les cas, les mots de passe devront être constitués d'au moins six caractères.
- no\_constraints - Aucune complexité n'est requise.
  - mixed\_letters - Les mots de passe doivent contenir au moins une lettre en MAJUSCULE et une lettre en minuscule.

- `letters_numbers` - Les mots de passe doivent contenir au moins une lettre en MAJUSCULE, une lettre en minuscule et un chiffre.
- `letters_numbers_specialchars` - Les mots de passe doivent contenir au moins une lettre, un chiffre et un caractère spécial.

**password\_check\_on\_login** Si activé, Bugzilla vérifiera que le mot de passe satisfait aux exigences de complexité et de longueur lors de la connexion de l'utilisateur à l'interface Web de Bugzilla. Si ce n'est pas le cas, l'utilisateur ne pourra pas se connecter et recevra un courriel pour réinitialiser son mot de passe.

### 4.1.5 Fichiers joints

Cette page permet la définition des restrictions et d'autres paramètres concernant les fichiers joints aux bogues. Par exemple, le contrôle des limitations de taille et l'autorisation de pointer vers des fichiers externes via une URI.

**allow\_attachment\_display** Si cette option est activée, les utilisateurs seront capables d'afficher les fichiers joints dans leur navigateur, si celui-ci gère le type MIME du fichier joint. Si cette option est désactivée, les utilisateurs seront forcés de télécharger les fichiers joints, même si le navigateur est capable de les afficher.

Si vous n'avez pas confiance en vos utilisateurs (par ex. si votre installation Bugzilla est publique), vous devez soit laisser cette option désactivée soit configurer et définir le paramètre `attachment_base` (voir ci-dessous). Des utilisateurs mal intentionnés peuvent téléverser des fichiers joints qui peuvent potentiellement être destructeurs s'ils sont visualiser directement dans le navigateur.

**attachment\_base** Quand le paramètre `allow_attachment_display` est activé, il est possible pour des fichiers joints malveillants de dérober vos cookies ou de réaliser une attaque sur Bugzilla en utilisant vos identifiants.

Si vous voulez une sécurité supplémentaire sur vos fichiers joints pour éviter ceci, définissez le paramètre avec une URL alternative pour votre installation Bugzilla qui ne soit pas la même que `urlbase` ou `sslbase`. C'est-à-dire, un nom de domaine différent qui redirige exactement sur cette installation Bugzilla.

Veillez noter que si vous avez défini le `cookiedomain` parameter, vous devez définir `attachment_base` pour utiliser un domaine qui ne corresponde pas à `cookiedomain`.

Pour une sécurité accrue, vous pouvez insérer `%bugid%` dans l'URL, qui sera remplacé par le numéro du bogue auquel le fichier est joint, quand vous accédez à un fichier joint. Ceci limitera les fichiers joints à accéder seulement aux autres fichiers joints du même bogue. Souvenez-vous toutefois que tous les domaines possibles (tels que `1234.your.domain.com`) doivent pointer vers cette même instance de Bugzilla. Pour cela, vous devrez considérer l'utilisation de DNS wildcard.

**allow\_attachment\_deletion** Si cette option est activée, les administrateurs seront capables de supprimer les fichiers joints.

**maxattachmentsize** La taille maximale (en kilooctets) des fichiers joints à stocker dans la base de données. Si un fichier plus gros que cette taille est joint à un bogue, Bugzilla consultera le `maxlocalattachment` pour déterminer si le fichier peut être stocké localement sur le serveur Web. Si la taille du fichier dépasse les deux limites, le fichier joint est alors rejeté. Définir les deux paramètres à 0 empêchera de joindre des fichiers aux bogues.

Certaines bases de données ont des limites par défaut qui empêchent de stocker de plus gros fichiers en base de données. Par exemple, MySQL a un paramètre appelé `max_allowed_packet`, dont la valeur par défaut varie selon la distribution. Définir la valeur de `maxattachmentsize` plus élevée que le paramètre courant pour cette valeur produira une erreur.

**maxlocalattachment** La taille maximale (en mégaoctets) des fichiers joints à stocker localement sur le serveur Web. Si cette valeur est inférieure à celle du paramètre `maxattachmentsize`, les fichiers joints ne seront jamais conservés sur le système de fichiers local.

L'utilisation de cette fonctionnalité dépend de votre environnement. Les raisons de stocker tout ou partie des fichiers peuvent être de piètres performances de la base de données pour les gros objets binaires (blobs), la facilité de sauvegarde/restauration/navigation ou même une déduplication au niveau du système de fichiers. Cependant, vous devez être conscient des limites de stockage de votre serveur Web. en cas de doute, laissez cette valeur à 0.

Veillez noter que la modification de cette valeur n'aura aucun effet sur les fichiers déjà soumis.

### 4.1.6 Politique de modification des bogues

Définit la politique sur le comportement par défaut des événements de modification de bogues. Par exemple, choisir l'état dans lequel mettre un bogue quand celui-ci est marqué comme doublon, et choisir d'autoriser si les rapporteurs de bogues peuvent définir la priorité ou le jalon cible. Permet aussi la configuration des changements qui nécessitent un commentaire de la part des utilisateurs, décrit ci-dessous.

**duplicate\_or\_move\_bug\_status** Quand un bogue est marqué comme doublon d'un autre ou est déplacé vers une autre installation, utiliser cet état de bogue.

**letsubmitterchoosepriority** Si activé, les personnes soumettant des bogues peuvent choisir la priorité initiale. Si ce paramètre est désactivé, alors tous les bogues auront par défaut la priorité sélectionnée ci-dessous.

**letsubmitterchoosemilestone** Si activé, les personnes soumettant des bogues peuvent choisir le jalon cible pour les bogues. Si ce paramètre est désactivé, alors tous les bogues auront le jalon qui a été défini par défaut pour le produit concerné.

**musthavemilestoneonaccept** Si vous utilisez « Jalon », voulez-vous que le jalon soit défini pour qu'un utilisateur puisse définir l'état d'un bogue à IN\_PROGRESS ?

**commenton\*** Tous ces champs vous permettent de définir quels changements peuvent être faits sans commentaire, et ceux qui doivent avoir un commentaire de la personne qui fait les changements. Souvent, les administrateurs autoriseront les utilisateurs à s'ajouter à la liste Copie à, à accepter les bogues ou à modifier le Tableau blanc sans ajouter de commentaire pour justifier les changements, et demanderont que la plupart des autres changements soit justifiés. Définissez les options `commenton` selon la politique de votre site. Il est sage de demander des commentaires quand les utilisateurs résolvent, réassignent ou rouvrent des bogues, au minimum.

---

**Note :** Il est généralement bien mieux de demander un commentaire au développeur lors de la résolution des bogues. Il y a peu de choses plus ennuyeuses pour les utilisateurs d'une base de données de bogues, que d'avoir un développeur marquant un bogue CORRIGÉ sans commentaire sur le correctif (ou même si cela a vraiment été corrigé !)

---

**noresolveonopenblockers** Cette option empêchera les utilisateurs de résoudre les bogues en CORRIGÉ s'il y a des dépendances non résolues. Seule la résolution CORRIGÉ est affectée. Les utilisateurs seront encore capables de résoudre les bogues avec des résolutions autres que CORRIGÉ s'il reste des bogues dépendants non résolus.

### 4.1.7 Champs des bogues

Les paramètres dans cette section déterminent le choix par défaut de plusieurs champs de Bugzilla pour les nouveaux bogues et contrôlent aussi si certains champs sont utilisés ou pas. Par exemple, l'utilisation des champs Jalon cible ou Tableau blanc.

**useclassification** Si activé, Bugzilla associera chaque produit à une catégorie spécifique. Mais vous devez avoir les permissions `editclassification` activées pour pouvoir modifier les catégories.

**usetargetmilestone** Voulez-vous utiliser le champ Target Milestone ?

**useqacontact** Ceci permet de définir une adresse électronique pour chaque composant, en plus de celle du responsable par défaut, à laquelle seront envoyées les copies de courriel de bogues.

**usestatuswhiteboard** Ceci définit si vous souhaitez avoir un champ de formulaire libre et modifiable associé à chaque bogue. L'avantage de ce tableau blanc est qu'il peut être effacé ou modifié facilement, et qu'il fournit un champ de recherche facile pour indexer des bogues qui ont des traits communs.

**use\_see\_also** Voulez-vous utiliser le champ Consulter aussi ? Il vous permet de référer à des bogues dans d'autres installations. Même si vous désactivez ce champ, les relations vers les bogues (URL) déjà définies sur les bogues apparaîtront encore et pourront être supprimées.

**defaultpriority** Priorité par défaut définie pour les nouveaux bogues.

**defaultseverity** Gravité par défaut définie pour les nouveaux bogues.

**defaultplatform** Plateforme qui est pré-sélectionnée dans le formulaire de soumission de bogue. Vous pouvez laisser ce champ vide : Bugzilla utilisera alors la plateforme indiquée par le navigateur.

**defaulttopsys** Système d'exploitation pré-sélectionné dans le formulaire de soumission de bogue. Vous pouvez laisser ce champ vide : Bugzilla utilisera alors le système d'exploitation indiqué par le navigateur

**collapsed\_comment\_tags** Une liste de mots-clés séparés par des virgules, qui, lorsqu'ils sont appliqués aux commentaires, feront masquer ceux-ci par défaut.

### 4.1.8 Graphiques de dépendance

Bugzilla peut générer des graphiques de relations de dépendance des bogues en utilisant un outil appelé `dot` (du [Projet GraphViz](#)) ou un service Web appelé Web Dot. Cette page permet de définir l'emplacement de l'exécutable ou du service Web. Si aucun exécutable ou serveur Web Dot n'est indiqué, alors les graphiques de dépendances seront désactivés.

**webdotbase** Vous pouvez définir ce paramètre pour les options suivantes :

- Un chemin d'accès de fichier complet jusqu'à `dot` (partie de GraphViz), qui générera les graphiques localement.
- Un préfixe d'URL pointant vers une installation du paquet Web Dot, qui générera les graphiques à distance.
- Une valeur vide désactivera les graphiques de dépendance.

La valeur par défaut est vide. Nous recommandons une installation locale de `dot`. Si vous changez cette valeur pour un service Web, assurez-vous que le serveur Web Dot puisse lire les fichiers de votre répertoire Web Dot. Sur Apache, vous faites cela en modifiant le fichier `.htaccess` ; pour les autres systèmes, les mesures à adopter peuvent varier. Vous pouvez exécuter **checksetup.pl** pour recréer le fichier `.htaccess` si celui-ci n'existe plus.

**font\_file** Vous pouvez indiquer le chemin absolu vers un fichier de fonte TrueType qui sera utilisé pour afficher du texte (libellés, légendes, ...) dans les tableaux et rapports graphiques. Pour gérer le plus de langues possibles, nous recommandons d'utiliser une fonte TrueType telle que Unifont qui gère tous les caractères imprimables dans le Plan multilingue de base. Si vous laissez ce paramètre vide, une fonte par défaut sera utilisée, mais celle-ci se limite à l'affichage des caractères anglais seulement et par conséquent, les autres caractères seront mal affichés.

### 4.1.9 Restrictions de groupe

Bugzilla permet la création de différents groupes, avec la possibilité de restreindre la visibilité des bogues dans un groupe à un ensemble d'utilisateurs spécifiques. Des produits spécifiques peuvent aussi être associés à des groupes, et des utilisateurs restreints à ne voir les produits que dans leurs groupes. Plusieurs paramètres sont décrits plus en détail ci-dessous. La plupart de la configuration des groupes et leurs relations aux produits est faite dans les pages [Groupes](#) et [Produit](#) dans la zone [Administration](#). Les options sur cette page contrôlent les comportements globaux par défaut. Pour plus d'informations sur les Groupes et les restrictions de groupes, consulter [Groupes et restrictions de groupes](#).

**makeproductgroups** Si ceci est activé, Bugzilla associera un groupe de bogue avec chaque produit dans la base de données et l'utilisera pour faire les recherches de bogues.

**chartgroup** Le nom du groupe d'utilisateurs qui peuvent utiliser la fonctionnalité « Nouveaux graphiques ». Les administrateurs doivent s'assurer que les catégories publiques et les définitions des collections ne divulguent pas d'informations confidentielles avant d'activer ceci pour une population non sûre. Si ceci est laissé vide, aucun utilisateur ne pourra utiliser la fonctionnalité « Nouveaux graphiques ».

**insidergroup** Le nom du groupe d'utilisateurs qui peuvent voir ou modifier les commentaires et les fichiers joints privés.

**timetrackinggroup** Le nom du groupe d'utilisateurs qui peuvent voir ou modifier les informations d'horodatage.

**querysharegroup** Le nom du groupe d'utilisateurs qui peuvent partager leurs recherches enregistrées avec d'autres. Pour plus de détails sur l'utilisation des recherches enregistrées, consulter [Recherches enregistrées](#).



**comment\_taggers\_group** Le nom du groupe d'utilisateurs qui peuvent appliquer des mots-clés sur les commentaires. Laisser ce paramètre vide désactive les mots-clés sur les commentaires.

**debug\_group** Le nom du groupe d'utilisateurs qui peuvent voir la requête SQL générée courante lors de l'affichage des listes et des rapports de bogues.

**usevisibilitygroups** Si activé, la visibilité des utilisateurs sera restreinte aux membres des groupes, sélectionné dans les paramètres de configuration de groupes. Chaque utilisateur d'un groupe peut être autorisé à voir les membres des groupes sélectionnés. Pour plus de détails sur la configuration des groupes (y compris les restrictions de visibilité), consulter *Modification de groupes et affectation de restrictions*.

**or\_groups** Définit la visibilité d'un bogue appartenant à plusieurs groupes the. Si ce paramètre est activé (ce qui est recommandé), un utilisateur doit seulement être membre d'un des groupes du bogue pour voir celui-ci. Si ce paramètre est désactivé, un utilisateur doit être membre de tous les groupes du bogue. Veuillez noter que dans chacun des cas, un rôle de l'utilisateur dans le bogue (par ex. rapporteur), peut aussi affecter ses permissions.

#### 4.1.10 LDAP

L'authentification LDAP est un module pour l'architecture de plugin d'authentification de Bugzilla. Cette page contient tous les paramètres nécessaires pour configurer Bugzilla en utilisant l'authentification LDAP.

Le schéma d'authentification existant de Bugzilla utilise les adresses électroniques comme identifiant primaire de l'utilisateur et un mot de passe pour authentifier cet utilisateur. Tous les endroits dans Bugzilla qui nécessitent un identifiant d'utilisateur (par ex. pour assigner un bogue) utilisent l'adresse électronique. L'authentification LDAP se situe au-dessus de ce schéma et ne le remplace pas. La connexion initiale est faite avec un nom d'utilisateur et un mot de passe pour l'annuaire LDAP. Bugzilla essaie de se lier à LDAP en utilisant les crédeniels et, s'il réussit, essaie d'associer ce compte à un compte Bugzilla. Si un attribut LDAP d'adresse électronique est défini, la valeur de cet attribut est utilisé, sinon, le paramètre `emailsuffix` est ajouté au nom d'utilisateur LDAP pour former l'adresse électronique complète. Si un compte pour cette adresse existe déjà dans l'installation de Bugzilla, il se connectera avec ce compte. Si aucun compte pour cette adresse électronique n'existe, il en sera créé un au moment de la connexion. (Dans ce cas, Bugzilla essaiera d'utiliser l'attribut `displayName` ou `cn` pour déterminer le nom complet de l'utilisateur). Après l'authentification, toutes les tâches liées à l'utilisateur seront toujours manipulées par l'adresse électronique et pas par le nom d'utilisateur LDAP. Par exemple, les bogues sont encore assignés par l'adresse électronique et les utilisateurs recherchés par leur adresse électronique.

**Avertissement :** Parce qu'un compte Bugzilla n'est pas créé jusqu'à ce que l'utilisateur se connecte pour la première fois, un utilisateur qui ne s'est pas encore connecté est inconnu de Bugzilla. Ceci signifie qu'il ne peut pas être utilisé comme Responsable ou Contact QA (par défaut ou non), ajouté à la liste `Copie` à ou toute autre opération de ce type. Un contournement possible est le script `bugzilla_ldapsync.rb` dans le répertoire `contrib`. Une autre solution est de résoudre le [bogue 201069](#).

Paramètres nécessaires pour utiliser l'authentification LDAP :

**user\_verify\_class (in the Authentication section)** Si vous voulez lister LDAP ici, assurez-vous d'avoir défini les autres paramètres listés ci-dessous. À moins d'avoir d'autres méthodes d'authentification (qui fonctionnent) listées aussi, vous ne pourrez pas vous reconnecter à Bugzilla une fois déconnecté. Si cela vous arrive, vous devrez modifier manuellement `data/params.json` et définir `user_verify_class` à DB.

LDAPserver

Le nom (et éventuellement le port) de votre serveur LDAP (par ex. `ldap.societe.com` ou `ldap.societe.com:numero_de_port`). La syntaxe URI peut également être utilisée, comme par exemple `ldaps://ldap.societe.com` (pour une connexion sécurisée) ou `ldapi://%2fvar%2flib%2fldap_sock` (pour une connexion locale par socket). Plusieurs nom d'hôtes ou URI peuvent être indiqués, séparés par des virgules. Chacun d'eux sera essayé jusqu'à établissement d'une connexion.

**Note :** Afin d'utiliser SSL avec LDAP, indiquez une URI avec `ldaps://` Ceci forcera l'utilisation de SSL sur le port 636. Par exemple, pour LDAP non sécurisé : `ldap://ldap.societe.com`, pour

LDAP avec SSL : `ldaps://ldap.societe.com` ou pour LDAP sur un socket de domaine Unix :  
`ldapi://%2fvar%2flib%2fldap_sock`.

---

**LDAPstarttls** Définit l'activation d'une communication chiffrée lors d'une connexion LDAP à ce serveur.

**LDAPbinddn [Optionnel]** Certains serveurs LDAP une liaison anonyme pour faire des recherches dans l'annuaire. Si c'est le cas pour votre configuration, vous devrez définir le paramètre `LDAPbinddn` pour le compte utilisateur que Bugzilla doit utiliser à la place de la liaison anonyme. Ex. :  
`cn=default,cn=utilisateur:mot_de_passe`

**LDAPBaseDN** Le paramètre `LDAPBaseDN` doit être défini pour indiquer l'emplacement dans votre arbre LDAP où vous voulez faire la recherche des adresses électroniques. Les `uid` doivent être uniques sous le DN indiqué ici. Ex. : `ou=Personne,o=Societe`

**LDAPuidattribute** Le nom de l'attribut contenant le nom de connexion de l'utilisateur. Ex. `uid`

**LDAPmailattribute** Le nom de l'attribut utilisateur de votre annuaire qui contient l'adresse électronique qui sera utilisée comme compte utilisateur dans Bugzilla. Si ce paramètre est vide, Bugzilla utilisera le nom d'utilisateur LDAP comme compte utilisateur de Bugzilla. Dans ce cas, vous voudrez peut-être aussi définir le paramètre `emailsuffix`. Ex. `mail`

**LDAPfilter** Le filtre LDAP pour « AND » avec `LDAPuidattribute` pour filtrer la liste des utilisateurs valides.

### 4.1.11 RADIUS

L'authentification RADIUS est un module pour l'architecture de plugin d'authentification de Bugzilla. Cette page contient tous les paramètres nécessaires pour configurer Bugzilla en utilisant l'authentification RADIUS.

---

**Note :** La plupart des avertissements concernant l'authentification LDAP s'applique aussi à l'authentification RADIUS. Consulter [LDAP](#) pour des détails.

---

Paramètres requis pour utiliser l'authentification RADIUS :

**user\_verify\_class (dans la section Authentification)** Si vous voulez lister RADIUS ici, assurez-vous d'avoir défini les autres paramètres listés ci-dessous. À moins d'avoir d'autres méthodes d'authentification (en fonction) listées aussi, vous pourriez ne pas être en mesure de vous reconnecter à Bugzilla après déconnexion. Si cela se produisait, vous devriez modifier manuellement le fichier `data/params.json` et définir le paramètre `user_verify_class` à DB.

**RADIUS\_server** Ce paramètre doit être renseigné avec le nom (et optionnellement le port) de votre serveur RADIUS.

**RADIUS\_secret** Ce paramètre doit être renseigné avec le secret du serveur RADIUS.

**RADIUS\_NAS\_IP** L'attribut « NAS-IP-Address » à utiliser pour échanger des données avec votre serveur RADIUS. Si non-spécifié, 127.0.0.1 sera utilisé. Utile seulement si le paramètre `user_verify_class` contient RADIUS.

**RADIUS\_email\_suffix** Bugzilla a besoin d'une adresse électronique pour chaque compte utilisateur. Par conséquent, il a besoin de déterminer l'adresse électronique correspondant à un utilisateur RADIUS. Bugzilla ne propose qu'un simple moyen de faire cela : il concatène un suffixe au nom d'utilisateur RADIUS pour le convertir en adresse électronique. Vous pouvez indiquer ce suffixe dans le paramètre `:paramval:RADIUS_email_suffix`. Si cette solution ne fonctionne pas pour vous, vous devrez certainement modifier `Bugzilla/Auth/Verify/RADIUS.pm` pour que cela corresponde à vos besoins.

### 4.1.12 Courriel

Cette page contient tous les paramètres pour configurer la façon dont Bugzilla traite les notifications par courriel qu'il envoie. Voir ci-dessous pour un résumé des options importantes.

**mail\_delivery\_method** Ce paramètre est utilisé pour indiquer comment sont envoyés les courriels ou s'il ne faut pas les envoyer. Il y a plusieurs options pour les différents MTA, avec deux options supplémentaires qui désactivent l'envoi de courriels. Test n'envoie pas les courriels mais les enregistre dans `data/mailler.testfile` pour qu'ils soient consultés plus tard. None désactive totalement l'envoi de courriels.

**mailfrom** C'est l'adresse électronique qui apparaîtra dans le champ De pour tous les courriels envoyés par cette installation de Bugzilla. Certains serveurs de messagerie nécessitent une adresse électronique valide ; par conséquent, il est recommandé de choisir une adresse électronique valide ici.

**use\_mailer\_queue** Dans de grosses installations de Bugzilla, la mise à jour des bogues peut être très lente, car Bugzilla envoie tous les courriels en une fois. Si vous activez ce paramètre, Bugzilla mettra en file d'attente tous les courriels et les enverra en tâche de fond. Ceci nécessite d'avoir installé certains modules Perl (indiqué par `c :file :checksetup.pl` pour cette fonctionnalité), et que le démon `jobqueue.pl` soit exécuté (sans quoi vos courriels ne seront pas envoyés). Ceci affecte tous les courriels envoyés par Bugzilla, et pas seulement les mises à jour de bogue.

**smtpserver** C'est l'adresse du serveur SMTP, si le paramètre `mail_delivery_method` est défini pour SMTP. Utiliser `localhost` si vous utilisez un MTA local, ou le nom du serveur SMTP distant. Ajouter `:` et le numéro de port s'il ne s'agit pas du numéro de port par défaut.

**smtp\_username** Nom d'utilisateur à utiliser pour l'authentification SASL sur le serveur SMTP. Laisser ce paramètre vide si le serveur ne nécessite pas d'authentification.

**smtp\_password** Mot de passe à utiliser pour l'authentification SASL sur le serveur SMTP. Ce paramètre sera ignoré si le paramètre `smtp_username` est laissé vide.

**smtp\_ssl** Active la gestion de SSL pour la connexion au serveur SMTP.

**smtp\_debug** Ce paramètre permet d'activer le débogage détaillé. Les messages sont indiqués dans le journal d'erreur du serveur Web.

**whinedays** Ceci indique le nombre de jours pendant lequel les bogues sont dans l'état CONFIRMÉ avant de notifier les personnes qu'elles ont de nouveaux bogues qui n'ont pas été touchés. Si vous ne comptez pas utiliser cette fonctionnalité, ne définissez pas la tâche de notification *whining cron job* décrite dans les instructions d'installation ou définissez cette valeur à 0 (ne jamais notifier).

**globalwatchers** Ceci permet de définir des utilisateurs spécifiques qui recevront une notification chaque fois qu'un nouveau bogue est saisi ou lors de changements sur un bogue existant, en fonction des permissions de l'ensemble des groupes. Cela peut-être utile pour envoyer les notifications sur une liste de diffusion par exemple.

#### 4.1.13 Options par défaut des requêtes

Cette page contrôle le comportement par défaut de Bugzilla concernant plusieurs aspects des requêtes sur les bogues. Les options comprennent ce que sont les options de requête par défaut, ce que renvoie la page `Mes bogues`, si les utilisateurs peuvent ajouter librement des bogues à la liste de citations, et le nombre de doublons de bogues nécessaire pour ajouter un bogue à la liste des bogues les plus fréquemment rapportés.

**quip\_list\_entry\_control** Contrôle la facilité avec laquelle les utilisateurs peuvent ajouter des entrées à la liste des citations.

- open - Les utilisateurs peuvent librement ajouter des entrées à la liste des citations et leurs entrées seront immédiatement visibles à l'affichage.
- moderated - Les citations peuvent être saisies mais nécessitent d'être approuvées par un modérateur avant d'être affichées.
- closed - Aucun nouvel ajout à la liste des citations n'est autorisé.

**mybugstemplate** This is the URL to use to bring up a simple 'all of my bugs' list for a user. `%userid%` will get replaced with the login name of a user. Special characters must be URL encoded.

**defaultquery** C'est la requête par défaut qui est utilisée initialement quand vous accédez à la page de recherche avancée. C'est dans un format de paramètre URL.

**search\_allow\_no\_criteria** À moins que le code n'autorise explicitement tous les bogues à être renvoyés, ce paramètre permet de bloquer l'exécution de requêtes sans critère. Quand il est désactivé, une requête doit avoir des critères indiqués pour limiter le nombre de bogues renvoyés à l'utilisateur. Quand il est activé, un utilisateur est autorisé à exécuter des requêtes sans critère et à obtenir tous les bogues qu'il peut voir dans sa liste. Activer ce paramètre est déconseillé sur de grosses installations.

**default\_search\_limit** Par défaut, Bugzilla limite les recherches faites dans l'interface Web en ne renvoyant qu'une partie des nombreux résultats, pour des raisons de performance. (Ceci n'affecte que les résultats de recherche au format HTML—Les formats CSV, XML et les autres ne sont pas concernés). Les utilisateurs peuvent cliquer sur un lien dans la page des résultats de recherche pour voir tous les résultats.

Normalement, vous n'avez pas à changer ceci—la valeur par défaut devrait être acceptable pour la plupart des installations.

**max\_search\_results** Le nombre maximum de bogues qu'une recherche puisse jamais renvoyer. Rapports graphiques et tabulaires en sont exemptés cependant.

### 4.1.14 Base de données esclave

Cette page contrôle si une base de données esclave est utilisée et tous les paramètres associés à cette base de données

La configuration d'une grosse base de données standard se compose d'un serveur maître et d'un ensemble de bases de données esclaves en lecture seule (que Bugzilla nomme « shadowdb »). Les requêtes n'effectuant pas de mises à jour de données peuvent être dirigées sur ces bases esclaves, enlevant ainsi la charge et le verrouillage sur la base de données maître lui permettant ainsi de consacrer ses ressources pour les écritures. Bugzilla dirigera les connexions vers les bases esclaves quand il sait qu'il n'y a pas besoin de mettre à jour la base de données (par ex. pour les recherches ou les affichages de bogues pour un utilisateur non connecté).

Bugzilla ne gère pas la synchronisation des données entre la base maître et les bases esclaves. Vous devrez donc configurer la réplication des bases sur votre serveur de base de données.

Si votre base esclave est sur un serveur différent, indiquez shadowdbhost et shadowdbport. Si elle est sur la même machine, indiquez shadowdbsock.

**shadowdbhost** L'hôte sur lequel se trouve la base de données esclave.

**shadowdbport** Le port sur lequel écoute la base de données esclave.

**shadowdbsock** Le socket utilisé pour se connecter à la base de données esclave si l'hôte est la machine locale.

**shadowdb** Le nom de la base de données esclave.

### 4.1.15 Memcached

**memcached\_servers** Si cette option est activée, Bugzilla intégrera Memcached. Indiquer un ou plusieurs serveurs, séparés par des espaces, en utilisant la notation hôte :port (par exemple : 127.0.0.1 :11211).

**memcached\_namespace** Indiquer une chaîne pour préfixer chaque clé dans Memcached.

### 4.1.16 Correspondance d'utilisateur

Les paramètres de cette page contrôlent la façon dont les utilisateurs sont sélectionnés et recherchés lors de l'ajout d'un utilisateur à un bogue. Par exemple, les utilisateurs doivent être sélectionnés lors du choix d'un responsable de bogue, de l'ajout à la liste `Copie` à ou lors de la sélection du `Contact QA`. Avec le paramètre `usemenuforusers`, il est possible de configurer Bugzilla pour afficher une liste des utilisateurs dans les champs plutôt qu'un champ de texte vide. Ceci ne doit être utilisé que pour des installations de Bugzilla ayant un petit nombre d'utilisateurs. Si les utilisateurs sont sélectionnés via une boîte de texte, cette page contient aussi les paramètres sur la façon dont les utilisateurs sont recherchés et le mode de correspondance lors de la saisie.

**usemenuforusers** Si cette option est définie, Bugzilla proposera une liste déroulante (au lieu d'un champ de saisie texte) où un utilisateur pourra être sélectionné. Cette option ne doit pas être sélectionnée sur les sites où il y a beaucoup d'utilisateurs.

**ajax\_user\_autocompletion** Si cette option est activée, saisir des caractères dans certains champs utilisateur fera apparaître une liste de correspondances dans laquelle vous pourrez sélectionner le bon nom.

**maxusermatches** La recherche est limitée à ce nombre de correspondances. Si ceci est défini à « 1 », aucun utilisateur ne sera affiché pour des correspondances ambiguës. Ceci est utile à des fins de respect de la vie privée des utilisateurs. Une valeur zéro signifie aucune limite.

**confirmuniqueusermatch** Définit si un écran de confirmation est affiché lorsqu'un seul utilisateur correspond à la chaîne saisie pour la recherche.

#### 4.1.17 Avancés

**cookiedomain** Le domaine pour les cookies de Bugzilla. Normalement laissé vide. Si votre site Web est `https://www.toto.com`, définir ceci avec `.toto.com` autorisera également `titi.toto.com` à accéder aux cookies de Bugzilla. Ceci est utile si vous avez plus d'un nom d'hôte pointant sur le même serveur Web et que vous voulez qu'ils partagent les cookies de Bugzilla.

**inbound\_proxies** Quand le trafic interne de Bugzilla passe par un proxy, Bugzilla pense que l'adresse IP de chaque utilisateur est l'adresse IP du proxy. Si vous saisissez une liste d'adresses IP séparées par des virgules dans ce paramètre, alors Bugzilla fera confiance à tous les en-têtes `X-Forwarded-For` envoyés pour ces adresses IP, et utilisera la valeur de cet en-tête comme l'adresse IP de l'utilisateur.

**proxy\_url** Bugzilla peut avoir accès au Web pour obtenir des notifications sur les nouvelles versions, voir le paramètre `upgrade_notification`. Si le serveur est derrière un proxy, il peut être nécessaire de saisir l'URL du serveur proxy si le groupe du serveur Web ne peut pas accéder à la variable d'environnement `HTTP_PROXY`. Si vous devez vous authentifier, utilisez la syntaxe `http://utilisateur:mot_de_passe@url_proxy/`.

**strict\_transport\_security** Active l'envoi de l'en-tête `Strict-Transport-Security` avec les réponses HTTP sur les connexions SSL. Ceci améliore grandement la sécurité de vos connexions SSL en forçant le navigateur à toujours accéder à votre domaine avec SSL et à ne jamais accepter de certificat invalide. Cependant, ceci ne doit être utilisé que si vous avez le paramètre `ssl_redirect` activé, que Bugzilla est la seule application s'exécutant sur son domaine (c-à-d., votre urlbase est de la forme `http://bugzilla.exemple.com/`), et que vous projetez ne de jamais désactiver le paramètre `ssl_redirect`.

- `off` - N'envoie pas les en-têtes `Strict-Transport-Security` avec les requêtes.
- `this_domain_only` - Envoie les en-têtes `Strict-Transport-Security` avec toutes les requêtes, mais ne le gère que pour le domaine en cours.
- `include_subdomains` - Envoie les en-têtes `Strict-Transport-Security` avec la balise `includeSubDomains`, qui appliquera le changement de sécurité à tous les sous-domaines. Ceci est particulièrement utile quand il est combiné avec un paramètre `attachment_base` qui existe en tant que sous-domaine du domaine Bugzilla principal.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 4.2 Préférences par défaut

Chaque utilisateur de Bugzilla peut définir un certain nombre de préférences sur le comportement qu'il désire pour Bugzilla. Vous pouvez indiquer ici si les préférences sont disponibles pour l'utilisateur et le cas échéant, la valeur par défaut.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 4.3 Utilisateurs

### 4.3.1 Créer un utilisateur administrateur

À la première exécution du script `checksetup.pl` après l'installation de Bugzilla, il vous demandera le nom de l'utilisateur d'administration (adresse électronique) et le mot de passe pour ce `super` utilisateur. Si pour une raison ou pour une autre vous supprimez le compte `super` utilisateur, la ré-exécution du script `checksetup.pl` vous permettra d'indiquer à nouveau ce nom d'utilisateur et son mot de passe.

Si vous souhaitez ajouter plus d'utilisateurs d'administration, ajoutez-les au groupe `admin` et optionnellement modifiez les groupes `tweakparams`, `editusers`, `creategroups`, `editcomponents` et `editkeywords` pour y ajouter le groupe `admin` (ce qui est le cas par défaut).

### 4.3.2 Rechercher les utilisateurs

Si vous avez le privilège `editusers` ou si vous êtes autorisé à donner des privilèges sur certains groupes, le lien `Utilisateurs` apparaîtra dans la page administration.

Le premier écran que vous obtenez est un formulaire de recherche des comptes utilisateurs existants. Vous pouvez lancer des recherches basées sur le numéro, le nom réel ou le nom de connexion (c-à-d. l'adresse électronique dans la plupart des cas, ou seulement la première partie de l'adresse si le paramètre `emailsuffix` est défini). Vous pouvez faire des recherches de différentes manières à l'aide de la liste déroulante à droite de la boîte de texte. Vous pouvez faire une recherche de correspondances de sous-chaînes insensibles à la casse (par défaut), avec une expression régulière, avec l'*inverse* d'une expression régulière, (qui trouve chaque nom d'utilisateur qui ne correspond PAS à l'expression régulière), ou la chaîne exacte si vous savez qui vous cherchez. La recherche peut être restreinte à des utilisateurs se trouvant dans un groupe spécifique. Par défaut, la restriction est désactivée.

La recherche renvoie une liste des utilisateurs correspondant à vos critères. Les propriétés de l'utilisateur peuvent être modifiées en cliquant sur le nom de connexion. L'historique du compte d'un utilisateur peut être lu en cliquant sur le lien `Afficher` dans la colonne `Journal du compte utilisateur`. Le journal du compte affiche les changements qui ont été faits sur le compte de l'utilisateur, la date et l'heure du changement et l'utilisateur qui a effectué les modifications. Par exemple, le journal du compte affichera les détails sur la suppression ou l'ajout d'un utilisateur à un groupe.

### 4.3.3 Modifier les utilisateurs

Quand vous avez trouvé votre utilisateur, vous pouvez changer les champs suivants :

- *Nom de connexion* : Ceci est généralement l'adresse électronique complète de l'utilisateur. Cependant, si vous utilisez le paramètre `emailsuffix`, ceci peut être juste le nom de connexion de l'utilisateur. Notez que les utilisateurs peuvent à présent changer leur nom de connexion eux-mêmes (pour une adresse électronique valide).
- *Nom réel* : Le nom réel de l'utilisateur. Notez que Bugzilla n'a pas besoin de ceci pour créer un compte.
- *Mot de passe* : Vous pouvez changer le mot de passe de l'utilisateur ici. Les utilisateurs peuvent automatiquement demander un nouveau mot de passe ; vous ne devriez donc pas avoir besoin de faire cela souvent. Si vous voulez désactiver un compte, voir `Texte de désactivation` ci-dessous.
- *Courriel de bogues désactivé* : Cochez cette case pour désactiver les courriels de bogues et de notification totalement pour ce compte. Cette case à cocher remplace le fichier `data/nomail` existant dans les versions précédentes de Bugzilla.
- *Texte de désactivation* : Si vous saisissez quoi que ce soit dans cette boîte, y compris juste une espace, l'utilisateur ne pourra pas se connecter ou faire des modifications de bogues via l'interface Web. Le code HTML que vous saisissez dans cette boîte sera affiché à l'utilisateur quand il essaiera de faire ces actions et doit expliquer pourquoi le compte a été désactivé. Les utilisateurs ayant leur compte désactivé continueront à recevoir des courriels de Bugzilla ; de plus, ils ne pourront pas se connecter eux-mêmes pour changer leurs préférences et

arrêter l'envoi de courriels. Si vous voulez qu'un compte (désactivé ou activé) arrête de recevoir les courriels, cochez la case `Courriel de bogues désactivé` ci-dessus.

**Note :** Même les utilisateurs ayant leur compte désactivé peuvent encore soumettre des bogues par l'intermédiaire de la passerelle de courriels, si elle existe. La passerelle de courriels ne devrait *pas* être activée pour les installations sécurisées de Bugzilla.

**Avertissement :** Ne désactivez pas tous les comptes administrateurs !

- `<nomgroupe>` : Si vous avez créé des groupes, par ex. `securitesensible`, alors des cases à cocher apparaîtront ici pour vous permettre d'ajouter ou de supprimer des utilisateurs de ces groupes. La première case à cocher donne à l'utilisateur la possibilité d'ajouter ou de supprimer d'autres utilisateurs comme membres de ce groupe. La seconde ajoute l'utilisateur lui-même en tant que membre du groupe.
- `canconfirm` : Ce champ est seulement utilisé si vous avez activé l'état `NON CONFIRMÉ`. Si vous activez ceci pour un utilisateur, cet utilisateur peut changer les bogues dans l'état `NON CONFIRMÉ` en `CONFIRMÉ` (par ex. : état `NOUVEAU`).
- `creategroups` : Cette option permet à un utilisateur de créer et supprimer des groupes dans Bugzilla.
- `editbugs` : À moins qu'un utilisateur n'ait cette option définie, il ne peut que modifier que les bogues pour lesquels il est responsable ou rapporteur. Même si cette option n'est pas cochée, les utilisateurs peuvent encore ajouter des commentaires aux bogues.
- `editcomponents` : Cette option permet aux utilisateurs de créer de nouveaux produits et composants et de modifier ou supprimer ceux pour lesquels aucun bogue n'est associé. Si un produit ou un composant a des bogues associés, ces bogues doivent être déplacés dans un produit ou un composant différent avant que Bugzilla n'autorise leur suppression.
- `editkeywords` : Si vous utilisez la fonctionnalité des mots-clés de Bugzilla, activer ceci permet à un utilisateur de créer et supprimer des mots-clés. Comme d'habitude, les bogues existants contenant le mot-clé que l'utilisateur veut supprimer doivent être modifiés avant que Bugzilla autorise la suppression du mot-clé.
- `editusers` : Cette option permet à un utilisateur de faire ce que vous êtes en train de faire : modifier d'autres utilisateurs. Ceci permettra à l'utilisateur ayant cette option d'ajouter ou de retirer les privilèges administrateur à d'autres utilisateurs. À activer avec précaution.
- `tweakparams` : Cette option permet à un utilisateur de modifier les paramètres de Bugzilla (en utilisant `editparams.cgi`.)
- `<nomproduit>` : Ceci autorise un administrateur à indiquer les produits pour lesquels un utilisateur peut voir les bogues. Si vous activez le paramètre `makeproductgroups` dans le panneau `Restrictions de groupe` dans la page `Paramètres`, Bugzilla crée alors un groupe par produit (au moment de la création du produit), et ce groupe a exactement le même nom que le produit. Veuillez noter que pour les produits existants, lorsque le paramètre est activé, les groupes correspondants ne seront pas créés. L'utilisateur doit encore avoir le privilège `editbugs` pour modifier les bogues dans ces produits.

### 4.3.4 Créer de nouveaux utilisateurs

#### Auto-enregistrement

Par défaut, les utilisateurs peuvent créer leur propre compte en cliquant sur le lien `Nouveau compte` au bas de chaque page (en supposant qu'ils ne soient pas déjà connectés avec un autre compte). Si vous voulez désactiver cet enregistrement automatique ou si vous voulez limiter ceux qui peuvent créer leur compte, vous devrez modifier le paramètre `createemailregexp` dans la page `Configuration`, voir [Paramètres](#).

#### Comptes créés par un administrateur

Les utilisateurs ayant le privilège `editusers`, tels que les administrateurs, peuvent créer des comptes pour d'autres utilisateurs :

1. Après la connexion, cliquez sur le lien `Utilisateurs` dans le pied de la page de requête, puis cliquez sur `Ajouter un nouvel utilisateur`.
2. Remplissez le formulaire présenté. Cette page est explicite. Quand c'est terminé, cliquez sur `Soumettre`.

---

**Note :** Ajouter un utilisateur de cette façon n'enverra *pas* un courriel l'informant de son nom d'utilisateur et de son mot de passe. Alors que c'est utile pour créer des comptes génériques (des scrutateurs qui redirigent les courriels vers un autre système par exemple ou des adresses électroniques qui sont des listes de diffusion), en général, il est préférable de se déconnecter et d'utiliser le bouton `Nouveau compte` pour créer des utilisateurs car cela pré-remplira tous les champs requis et notifiera également l'utilisateur de son nom de compte et de son mot de passe.

---

### 4.3.5 Supprimer des utilisateurs

Si le paramètre `allowuserdeletion` est activé, voir *Paramètres*, vous pouvez alors aussi supprimer des comptes utilisateurs. Notez que ce n'est la plupart du temps pas la meilleure chose à faire. Si seul un avertissement dans une boîte jaune est affiché, alors la suppression est sûre. Si un avertissement est également affiché dans une boîte rouge, alors vous ne devriez PAS essayer de supprimer le compte utilisateur, sinon vous rencontrerez des problèmes d'intégrité référentielle dans votre base de données, qui peuvent mener à un comportement inattendu, tels que des bogues n'apparaissant plus dans les listes de bogues ou des données ou des données incorrectement affichées. Vous avez été prévenu !

### 4.3.6 Se substituer à des utilisateurs

Il y a des fois où un administrateur veut accomplir des actions en tant qu'un autre utilisateur. La fonctionnalité `sudo` peut être utilisée pour faire cela.

---

**Note :** Pour utiliser la fonctionnalité `sudo`, vous devez être dans le groupe `bz_sudoers`. Par défaut, tous les administrateurs sont dans ce groupe.

---

Si vous voulez utiliser cette fonctionnalité, vous devez démarrer une session en allant sur la page de modification des utilisateurs, rechercher un utilisateur et cliquer sur son nom de connexion. Vous devriez voir un lien sous son nom de connexion intitulé `Prendre la place de cet utilisateur`. Cliquez sur le lien. Ceci vous amènera sur une page décrivant la fonctionnalité et les instructions pour l'utiliser. Après avoir lu le texte, saisissez le nom de connexion de l'utilisateur dont vous voulez usurper l'identité, fournissez un court message expliquant pourquoi vous faites cela, et appuyez sur le bouton.

Tant que vous utiliserez cette fonctionnalité, tout ce que vous ferez sera vu comme si vous étiez connecté avec le compte utilisateur dont vous usurpez l'identité.

|  |
|--|
| <p><b>Avertissement :</b> L'utilisateur dont vous usurpez l'identité ne recevra pas de compte-rendu de ce que vous faites. Si vous effectuez des actions qui engendrent l'envoi de courriels, ces courriels apparaîtront comme envoyés par l'utilisateur dont vous usurpez l'identité. Vous devez être extrêmement prudent lorsque vous utilisez cette fonctionnalité.</p> |
|--|

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).



## 4.4 Catégories, produits, composants, versions et jalons

Bugs in Bugzilla are classified into one of a set of admin-defined Components. Components are themselves each part of a single Product. Optionally, Products can be part of a single Classification, adding a third level to the hierarchy.

### 4.4.1 Catégories

Les catégories sont utilisées pour regrouper plusieurs produits en une entité distincte.

Par exemple, si une société réalise des jeux vidéos, elle pourrait avoir une catégorie `Jeux` et un produit séparé pour chaque jeu. Cette société pourrait aussi avoir une catégorie `Commun`, contenant des produit représentant des unités technologiques utilisées dans plusieurs jeux et peut-être une catégorie `Autre` contenant des produits spéciaux ne représentant pas des produits commerciaux (par exemple : « Site web » ou « Administration »).

Le niveau `Catégories` est désactivé par défaut; il peut être activé ou désactivé en utilisant le paramètre `useclassification`, dans la section *Champs de bogue* dans la page de modification des paramètres.

L'accès à l'administration des catégories est contrôlé par l'utilisation du groupe `groupe système` *editclassifications*, qui définit les privilèges de création, suppression et modification des catégories.

Lorsqu'elles sont activées, les catégories introduisent une étape supplémentaire lors de la création des bogues (se manifestant par la sélection d'une catégorie); elles apparaissent aussi dans le formulaire de recherche avancée.

### 4.4.2 Produits

Les Produits représentent typiquement les produits que vous délivrez réellement. Les produits peuvent être classés en *Catégories*. Par exemple, si une société conçoit des jeux pour ordinateur, elle pourrait avoir une catégorie `Jeux` et un produit différent pour chaque jeu. Cette société pourrait également avoir un produit `Commun` pour les unités technologiques utilisées dans plusieurs jeux, et peut-être aussi quelques produit spéciaux qui représentent des éléments ne faisant pas partie des produits (par exemple, `Site Web` ou `Administration`).

Beaucoup de paramètres de Bugzilla sont configurables par produit. Le nombre de `votes` disponibles pour les utilisateurs est défini par produit, tout comme le nombre de votes requis pour faire passer automatiquement un bogue de l'état NON CONFIRMÉ à l'état CONFIRMÉ.

Lors de la création ou de la modification des produits, les options suivantes sont disponibles :

**Produit** Le nom du produit

**Description** Une brève description du produit

**Jalon par défaut** Sélectionner le jalon par défaut pour ce produit.

**Fermé pour la saisie de bogues** Sélectionner cette case à cocher pour empêcher la saisie de nouveaux bogues pour ce produit.

**Votes maximum par personne** Le nombre maximum de votes autorisé pour un utilisateur pour ce produit.

**Votes maximum qu'une personne peut affecter à un bogue** Le nombre de votes maximum autorisé pour un utilisateur dans ce produit pour un seul bogue.

**Seuil de confirmation** Le nombre de votes nécessaire pour passer automatiquement tout bogue dans ce produit de l'état NON CONFIRMÉ à NOUVEAU.

**Versión** Indique quelle version sera affichée pour les bogues de ce produit.

**Créer des graphiques pour ce produit** Cocher cette case pour permettre la création de graphiques pour ce produit.

Lors de la modification d'un produit, il y a également un lien pour modifier les restrictions de groupes, voir *Affecter des restrictions de groupes à des produits*.

### Créer de nouveaux produits

Pour créer un nouveau produit :

1. Sélectionnez `Produits` dans le pied de page
2. Sélectionnez le lien `Ajouter` au bas de la page à droite
3. Saisissez le nom du produit et sa description. Le champ `Description` peut contenir du code HTML.
4. Quand le produit est créé, Bugzilla affiche un message indiquant qu'un composant doit être créé avant de pouvoir rapporter des bogues pour ce nouveau produit. Suivre le lien pour créer un nouveau composant. Voir [Composants](#) pour plus de détails.

### Modifier des produits

Pour modifier un produit existant, cliquez sur le lien `Produits` dans la page `Administration`. si le paramètre `useclassification` est activé, un tableau des catégories existantes est affiché, y compris la catégorie `Unclassified`. Le tableau indique le nombre de produits dans chaque catégorie. Cliquez sur le nom de la catégorie pour voir ses produits. Si le paramètre `useclassification` n'est pas activé, le tableau liste tous les produits directement. Le tableau du produit résume les informations sur celui-ci fournie lors de sa création. Cliquez sur le nom du produit pour modifier ses propriétés et accéder aux liens vers les autres attributs tels que les composants, les versions, les jalons et les restrictions de groupe.

### Ajouter ou modifier les composants, versions et jalons cibles

Pour modifier ou ajouter de nouveaux composants, versions ou jalons cibles à un produit, cliquez sur les liens `Modifier les composants`, `Modifier les versions` ou `Modifier les jalons` dans la page `Modifier le produit`. Un tableau des composants, versions et jalons existants est affiché. Cliquez sur un nom d'élément pour modifier ses propriétés. Sous le tableau se trouvent un lien pour ajouter un nouveau composant, version ou jalon.

Pour plus d'informations sur les composants, consultez [Composants](#).

Pour plus d'informations sur les versions, consultez [Versions](#).

Pour plus d'informations sur les jalons, consultez [Milestones](#).

### Affecter des restrictions de groupes à des produits

Sur la page `Modifier le produit`, il y a un lien appelé `Modifier les restrictions de groupes`. Les paramètres de cette page contrôlent les relations des groupes au produit édité.

Les restrictions de groupe sont un aspect important de l'utilisation des groupes pour isoler et restreindre les accès aux bogues saisis pour ces produits. Pour plus d'informations sur les groupes, y compris la création, la modification, l'ajout d'utilisateurs et la modifications des permissions, consultez [Groupes et restrictions de groupes](#).

Après avoir cliqué sur le lien `Modifier les restrictions de groupes` dans la page `Modifier le produit`, un tableau contenant tous les groupes d'utilisateurs de cette installation de Bugzilla est affiché. Les groupes système qui sont créés lors de l'installation de Bugzilla is installed ne sont pas utilisables pour les restrictions de groupes. Une description de la signification de chacun de ces champs est indiquée ci-dessous.

Les groupes peuvent être applicable, défaut, et obligatoire. Les groupes peuvent aussi contrôler les droits de saisie pour un produit donné ou être utilisés pour que les bogues dans le produit soient en lecture seule à moins que les restrictions de groupes ne soient remplies. La meilleure façon comprendre ces relations est de voir des exemples. Consultez de [Applications courantes des restrictions de groupe](#) pour des exemples de relations entre produit et groupe.

**Note :** Les produits et les groupes ne sont pas limités à une relation un pour un. Plusieurs groupes peuvent être associés

au même produit et les groupes peuvent être associés à plus d'un produit.

Si un groupe a *Entry* sélectionné, alors le produit restreindra la saisie de bogues aux seuls utilisateurs membres de groupes ayant *entry* sélectionné.

Si un groupe a *canedit* sélectionné, alors le produit sera en lecture seule pour tous les utilisateurs qui ne sont pas membres de tous les groupes ayant *canedit* sélectionné. *Seuls* les utilisateurs qui sont membres de tous les groupes ayant *canedit* sélectionné seront capables de modifier. C'est une restriction additionnelle qui limite encore plus ce qui peut être modifié par un utilisateur.

Les paramètres suivants vous permettent de choisir les privilèges *par produit*. C'est un moyen pratique pour donner des privilèges à certains utilisateurs pour certains produits seulement, sans avoir à leur donner de privilèges globaux qui affecteraient tous les produits.

Les groupes ayant *editcomponents* sélectionné permettent aux utilisateurs membres de ces groupes de modifier tous les aspects de ce produit, y compris les composants, les jalons et les versions.

Les groupes ayant *canconfirm* sélectionné permettent aux utilisateurs membres de ces groupes de confirmer les bogues dans ce produit.

Les groupes ayant *editbugs* sélectionné permettent aux utilisateurs membres de ces groupes de modifier tous les champs de bogue dans ce produit.

Les champs *MemberControl* et *OtherControl* sont utilisés en tandem pour déterminer quels bogues seront placés dans ce groupe. Les seules combinaisons autorisées de ces deux paramètres sont listées dans un tableau dans la page *Modifier les restrictions de de groupe*. Consultez ce tableau pour des détails sur la façon d'utiliser ces champs. Des exemples de différentes utilisations sont décrits ci-dessous.

### Applications courantes des restrictions de groupe

L'utilisation des groupes est mieux expliquée avec des exemples illustrant des configurations utilisées couramment. Les exemples suivent une syntaxe commune : *Group* : *Entry*, *MemberControl*, *OtherControl*, *CanEdit*, *EditComponents*, *CanConfirm*, *EditBugs*. Où *Group* est le nom du groupe modifié pour ce produit. Les autres champs correspondent tous au tableau dans la page *Modifier les restrictions de groupes*. Si une de ces options n'est pas listée, cela signifie qu'elle n'a pas été cochée.

#### Restriction produit/groupe basique

Supposons qu'il y ait un produit appelé *Toto*. Le produit *Toto* ne peut contenir que des bogues saisis par les utilisateurs du groupe *Titi*. De plus, les bogues saisis pour le produit *Toto* doivent être toujours restreints aux utilisateurs du groupe *Titi*. De plus, seuls les membres du groupe *Titi* peuvent modifier les bogues saisis pour le produit *Toto*, même si d'autres utilisateurs peuvent voir le bogue. Ces conditions seraient réalisées ainsi :

```
Produit Toto:
Titi: ENTRY, OBLIGATOIRE/OBLIGATOIRE, CANEDIT
```

Peut-être que des restrictions si strictes ne sont pas nécessaires pour le produit *Toto*. Un façon moins stricte de configurer le produit *Toto* et le groupe *Titi* serait :

```
Produit Toto:
titi: ENTRY, AFFICHÉ/AFFICHÉ, EDITCOMPONENTS, CANCONFIRM, EDITBUGS
```

Les lignes ci-dessus indiquent que, pour le produit *Toto*, les membres du groupe *Titi* peuvent saisir des bogues. Toute personne ayant la permission de modifier un bogue du produit *Toto* peut mettre le bogue dans le groupe *Titi*, même s'ils n'appartiennent pas eux-mêmes au groupe *Titi*. Tout utilisateur du groupe *Titi* peut modifier tous les aspects des composants du produit *Toto*, peut confirmer les bogues pour le produit *Toto* et peut modifier tous les champs de tous les bogues du produit *Toto*.

### Accès utilisateur général avec groupe de sécurité

Pour permettre à tout utilisateur de saisir un bogue pour le produit A et de soumettre ces bogues au groupe appelé Security :

```
Produit A:  
Security: AFFICHÉ/AFFICHÉ
```

### Accès utilisateur général et produit Sécurité

Pour permettre à tout utilisateur de saisir des bogues pour le produit Sécurité tout en empêchant ces bogues d'être visibles à quiconque en dehors du groupe SecurityWorkers (à moins qu'un membre de ce groupe n'enlève cette restriction) :

```
Produit Sécurité:  
securityworkers: DÉFAUT/OBLIGATOIRE
```

### Isolation de produit avec un groupe commun

Pour permettre aux utilisateurs du produit A d'accéder aux bogues du produit A, aux utilisateurs du produit B d'accéder aux bogues du produit B et au personnel du support, membres du groupe Support, d'accéder aux deux produits, trois groupes sont nécessaires :

1. Le groupe Support : contient les membres du personnel du support.
2. Le groupe AccessA : contient les utilisateurs du produit A et du groupe Support.
3. Le groupe AccessB : contient les utilisateurs du produit B et du groupe Support.

Quand ces trois groupes ont été définis, les restrictions de groupes peuvent être définies ainsi :

```
Produit A:  
AccessA: ENTRY, OBLIGATOIRE/OBLIGATOIRE  
Produit B:  
AccessB: ENTRY, OBLIGATOIRE/OBLIGATOIRE
```

Peut-être que le groupe Support veut plus de droits. Par exemple, le groupe Support pourrait être autorisé à rendre les bogues inaccessible aux utilisateurs des groupes AccessA et AccessB. Alors, le groupe Support pourrait être autorisé à publier les bogues appropriés à tous les utilisateurs dans un troisième produit (appelons-le Commun) qui est en lecture seule pour quiconque n'appartient pas au groupe Support. De cette façon, le groupe Support pourrait contrôler les bogues qui peuvent être vus par les deux groupes à la fois. Cette configuration serait :

```
Produit A:  
AccessA: ENTRY, OBLIGATOIRE/OBLIGATOIRE  
Support: AFFICHÉ/NON APPLICABLE  
Produit B:  
AccessB: ENTRY, OBLIGATOIRE/OBLIGATOIRE  
Support: AFFICHÉ/NON APPLICABLE  
Produit Commun:  
Support: ENTRY, DÉFAUT/OBLIGATOIRE, CANEDIT
```

## Mettre un produit en lecture seule

Quelquefois, un produit est retiré et par conséquent, on ne devrait pas pouvoir saisir de bogues pour celui-ci (par exemple, une ancienne version d'un logiciel qui n'est plus supportée). Un produit peut être mis en lecture seule en créant un groupe appelé `LectureSeule` et en ajoutant les produits dedans quand c'est nécessaire :

```
Produit A:  
LectureSeule: ENTRY, NON APPLICABLE/NON APPLICABLE, CANEDIT
```

---

**Note :** Pour plus d'informations sur les groupes (en dehors de leurs relations avec les produits), consultez *Groupes et restrictions de groupes*.

---

### 4.4.3 Composants

Les composants sont des sous-sections d'un produit. Par exemple, le jeu vidéo que vous concevez peut avoir les composants `UI`, `API`, `Système audio` et `Plugins`, chacun d'eux étant supervisé par un programmeur différent. Il est souvent souhaitable de diviser les composants dans Bugzilla en fonction des divisions naturelles des responsabilités au sein de votre produit ou de votre société.

Chaque composant a un responsable par défaut et, si vous l'activez dans la page des paramètres, un contact QA par défaut. Le responsable par défaut doit être la première personne qui corrige les bogues dans ce composant. Le contact QA doit être la personne qui s'assure que les bogues sont totalement corrigés. Le responsable, le contact QA et le rapporteur recevront un courriel quand de nouveaux bogues sont créés dans ce composant et quand il y a des changements sur les bogues. Les champs `Responsable par défaut` et `Contact QA par défaut` ne concernent que les *assignations par défaut* ; ils peuvent être changés lors de la soumission du bogue ou plus tard dans le cycle de vie du bogue.

Pour créer un nouveau composant :

1. Cliquez sur le lien `Modifier les composants dans la page Modifier le produit`
2. Cliquez sur le lien `Ajouter` au bas de la page à droite.
3. Remplissez le champ `Composant`, saisissez une courte `Description`, le `Responsable par défaut`, la `En copie par défaut` et le `Contact QA par défaut` (si activé). Le champ `Description` du composant peut contenir un nombre limité de balises HTML. Les champs `Composant` et `Description` peuvent contenir du code HTML ; le champ `Responsable par défaut` doit être un nom de connexion déjà existant dans la base de données de Bugzilla.

### 4.4.4 Versions

Les versions sont les révisions du produit, comme par exemple, `Flinders 3.1`, `Flinders 95` et `Flinders 2000`. Le champ `Version` n'est pas un champ à sélections multiples ; la pratique habituelle est de sélectionner la version la plus ancienne connue ayant le bogue.

Pour créer et modifier les versions :

1. Dans la page `Modifier le produit`, sélectionnez `Modifier les versions`
2. Vous remarquerez que le produit a déjà la version par défaut `undefined`. Cliquez sur le lien `Ajouter` au bas de la page à droite.
3. Saisissez le nom de la version. Ce champ ne prend que du texte. Puis, cliquez sur le bouton `Ajouter`.

## 4.4.5 Milestones

Les jalons sont des objectifs pour lesquels vous projetez de corriger un bogue. Par exemple, vous avez un bogue que vous prévoyez de corriger pour la version 3.0, on doit donc lui assigner le jalon 3.0.

---

**Note :** Les options de jalons n'apparaissent pour un produit que si vous avez activé le paramètre `usetargetmilestone` dans l'onglet `Champs des bogues` dans la page `Paramètres`.

---

Pour créer de nouveaux jalons et définir des jalons par défaut :

1. Sélectionnez `Modifier les jalons` dans la page `Modifier le produit`.
  2. Sélectionnez `Ajouter` dans le coin inférieur droit.
  3. Saisissez le nom du jalon dans le champ `Jalon`. Vous pouvez optionnellement définir la `Position`, qui est un nombre positif ou négatif (-32768 to 32767) qui définit où dans la liste apparaît ce jalon. Ceci est utile car les jalons ne suivent pas l'ordre alphanumérique. Par exemple, `Futur` devrait se trouver après `Version 1.2`. Cliquez sur `Ajouter`.
- 

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 4.5 Étiquettes

Si vous avez le privilège `editcomponents`, vous pouvez modifier les `Types d'étiquettes` dans la page principale d'administration. Cliquez sur le lien `Étiquettes` vous amènera sur la page `Gérer les types d'étiquettes`. Vous pouvez sélectionner ici si vous voulez créer (ou modifier) une étiquette de bogue ou de fichier joint.

Peu importe ce que vous choisissez, l'interface est la même, donc nous ne l'aborderons qu'une fois.

### 4.5.1 Propriétés des étiquettes

**Nom** C'est le nom de l'étiquette. Il sera affiché aux utilisateurs de Bugzilla qui consulteront ou définiront les étiquettes. Le nom peut contenir tout caractère Unicode valide sauf les virgules et les espaces.

**Description** Explique de façon détaillée l'étiquette. Elle est visible dans une info-bulle quand le curseur de la souris passe au-dessus du nom de l'étiquette dans les pages `Afficher le bogue` et `Modifier le fichier joint`. Ce champ peut être aussi long que vous le voulez et contenir tous les caractères que vous voulez.

**Categorie** Vous pouvez définir la visibilité d'une étiquette sur toute combinaison de produits et composants.

Le comportement par défaut pour une nouvelle étiquette est d'apparaître sur tous les produits et composants, c'est pourquoi `__Tous__` : `__Tous__` est déjà saisi dans la boîte `Inclusions`. Si ce n'est pas ce que vous voulez, vous pouvez soit définir des exclusions (pour les produits pour lesquels vous ne voulez pas voir apparaître l'étiquette), soit supprimer `__Tous__` : `__Tous__` de la boîte `Inclusions` et définir spécifiquement les produits/composants pour cette étiquette.

Pour créer une inclusion, sélectionnez un produit dans la liste déroulante. Vous pouvez également sélectionner un composant spécifique dans la liste déroulante. (Définir `__Tous__` pour `Produit` se traduit par tous les produits dans cette installation de Bugzilla. Sélectionner `__Tous__` dans le champ `Composant` signifie tous les composants du produit sélectionné.) quand les sélections sont faites, cliquez sur `Inclure` et les paires `Produit/Composant` apparaîtront dans la boîte `Inclusions` à droite.

Pour créer une exclusion, le processus est le même ; sélectionnez un produit dans la liste déroulante, sélectionnez un composant spécifique si vous en voulez un, et cliquez sur `Exclure`. Les paires `produit/composant` apparaîtront dans la boîte `Exclusions` à droite.

Cette étiquette *sera et pourra* être définie pour tous les produits/composants apparaissant dans la boîte `Inclusions` (ou qui tombent dans la règle `__Tous__` appropriée). Cette étiquette *n'apparaîtra pas* (et par conséquent ne pourra être définie) pour tout produit apparaissant dans la boîte `Exclusions`. **IMPORTANT :** *Les exclusions surpassent les inclusions.*

Vous pouvez sélectionner un produit sans sélectionner de composant spécifique, mais vous ne pouvez pas sélectionner de composant sans produit ou sélectionner de composant qui n'appartienne à aucun produit. Si vous faites cela, Bugzilla affichera un message d'erreur, même si tous vos produits ont un composant ayant ce nom.

*Exemple* Imaginons que vous ayez un produit intitulé `Avion` qui a des milliers de composants. Vous voulez avoir la possibilité de demander si un problème doit être corrigé pour le prochain modèle d'avion que vous fabriquez. Nous appellerons l'étiquette `corrigerProchainModèle`. Mais il y a un composant dans `Avion` intitulé `Pilote`. Cela n'a pas de sens de fabriquer un nouveau pilote, et donc vous ne voulez pas avoir cette étiquette dans ce composant. Donc vous incluez `Avion: __Tous__` et vous excluez `Avion: Pilote`.

**Position** Les étiquettes s'affichent normalement en ordre alphabétique. Si vous voulez les afficher dans un ordre différent, vous pouvez utiliser ceci pour définir la position de chaque étiquette. Les étiquettes ayant une valeur de position faible apparaîtront avant les étiquettes ayant une valeur de position élevée. Les étiquettes ayant la même valeur de position seront classées alphabétiquement, mais seront encore placées après les étiquettes ayant une plus faible valeur de la position et avant celles ayant une valeur de position plus élevée.

**Active** Parfois, vous pourriez vouloir conserver les informations sur les anciennes étiquettes dans la base de données de Bugzilla, mais empêcher les utilisateurs de continuer à les utiliser. Pour faire cela, décochez `active`. Les étiquettes désactivées continueront à apparaître dans l'interface utilisateur si elles ont les valeurs `?`, `+` ou `-` mais elles ne peuvent être qu'effacées (non définies) et ne peuvent pas prendre de nouvelle valeur. Quand une étiquette désactivée est effacée (non définie), elle disparaîtra complètement d'un bogue/fichier joint et ne pourra pas être redéfinie.

**Demandé** Les nouvelles étiquettes sont par défaut de type `demandé`, ce qui veut dire qu'elles permettent aux utilisateurs de définir les options `?`, `+` et `-`. Pour supprimer l'option `?`, décochez `demandé`.

**Sollicité** Par défaut cette boîte est cochée pour les nouvelles étiquettes, ce qui veut dire que les utilisateurs peuvent faire des demandes d'étiquettes à des personnes en particulier. Décocher cette case enlèvera la boîte de texte à côté de l'étiquette ; si elle est toujours de type `demandé`, alors les requêtes pourront seulement être faites en l'air. Enlever ceci après que des requêtes de type `sollicité` aient été faites ne supprimera pas ces requêtes ; ces informations resteront dans la base de données (bien qu'elles ne s'afficheront plus pour l'utilisateur).

**Multiple** Une étiquette de type `Multiple` (activé par défaut pour les nouvelles étiquettes) peut être définie plus d'une fois. Après avoir été définie une fois, une étiquette du même type apparaîtra au-dessous avec le mot `suppl.` (abrégé pour `supplémentaire`) devant son libellé. Il n'y a pas de limite au nombre de fois qu'une étiquette de type `multiple` puisse être définie sur le même bogue/fichier joint.

**Liste Copie à** Si vous voulez que certains utilisateurs soient notifiés chaque fois que cette étiquette change de valeur (`?`, `-`, `+` ou vide), ajoutez-les ici. C'est une liste d'adresses électroniques séparées par des virgules qui ne sont pas nécessairement des comptes utilisateurs de Bugzilla.

**Groupe des permissions** Quand ce champ est défini avec un groupe donné, seuls les utilisateurs de ce groupe peuvent définir la valeur de l'étiquette à `+` et `-`. Ce champ n'affecte pas ceux qui peuvent demander ou effacer l'étiquette. Pour cela, voir le champ `Groupe des requêtes` ci-dessous. Si ce champ est laissé vide, tous les utilisateurs peuvent définir ou effacer cette étiquette. Ce champ est utile pour limiter les utilisateurs qui peuvent approuver ou rejeter les requêtes.

**Groupe des requêtes** Quand ce champ est défini sur un groupe donné, seuls les utilisateurs de ce groupe peuvent demander ou effacer cette étiquette. Notez que ce champ n'a aucun effet si le champ `Groupe des permissions` est vide. Vous pouvez définir la valeur de ce champ la valeur de ce champ pour un groupe différent, mais les deux champs doivent être renseignés avec un groupe pour que ce champ ait un effet.

## 4.5.2 Supprimer une étiquette

Quand vous êtes sur l'écran `Gérer les types d'étiquettes`, une liste des étiquettes de bogues et de fichiers joints est affichée.

Pour supprimer une étiquette, cliquez sur le lien `Supprimer` à côté de la description de l'étiquette.

**Avertissement :** Une fois supprimée, l'étiquette n'existe *plus* dans Bugzilla. Toutes les données concernant cette étiquette seront supprimées. Partout où cette étiquette était définie, elle disparaîtra, et vous ne pourrez pas revenir en arrière. Si vous voulez conserver les données concernant l'étiquette, mais que personne ne continue à l'utiliser, décochez `active` dans le formulaire de modification des étiquettes.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 4.6 Champs personnalisés

Bugzilla 3.0 a introduit la possibilité de créer des champs personnalisés. Les champs personnalisés sont traités comme tout autre champ : ils peuvent être définis dans les bogues et utilisés dans les requêtes. Les administrateurs doivent garder à l'esprit qu'ajouter trop de champs peut rendre l'interface utilisateur plus compliquée et plus difficile à utiliser. Les champs personnalisés ne devraient être ajoutés que lorsqu'ils sont absolument nécessaires et en y portant une attention particulière.

---

**Note :** Avant d'ajouter un champ personnalisé, assurez-vous que Bugzilla ne peut pas déjà réaliser le comportement escompté. Beaucoup d'options de Bugzilla ne sont pas activées par défaut, et souvent, les administrateurs trouvent qu'activer simplement certaines options existantes suffit. Les administrateurs peuvent gérer les champs personnalisés en utilisant le lien `Champs personnalisés` dans la page d'administration. La page d'administration des champs personnalisés affiche une liste de champs personnalisés, s'il y en a, et un lien `Ajouter un nouveau champ personnalisé`.

---

### 4.6.1 Ajouter des champs personnalisés

Pour ajouter un nouveau champ personnalisé, cliquez sur le lien `Ajouter un nouveau champ personnalisé`. Cette page affiche plusieurs options pour le nouveau champ, comme indiqué ci-dessous.

Les attributs suivants doivent être définis pour chaque nouveau champ personnalisé :

- *Nom* : Le nom du champ utilisé dans la base de données, utilisée en interne. Ce nom DOIT commencer par `cf_` pour éviter toute confusion avec les champs standards. Si vous omettez cette chaîne, elle sera automatiquement ajoutée au nom que vous avez saisi.
- *Description* : Une chaîne courte qui est utilisée comme libellé pour ce champ personnalisé. C'est la chaîne que les utilisateurs verront ; elle doit donc être courte et explicite.
- *Type* : Le type du champ à créer. Il existe différents types disponibles :

**ID de bogue** Un champ où l'on peut saisir l'ID d'un autre bogue de la même installation Bugzilla. Pour indiquer le bogue d'une autre installation de Bugzilla, utiliser le champ `Consulter` aussi.

**Grande boîte de texte** : Une boîte de plusieurs lignes pour saisir du texte.

**Texte libre** : Une boîte d'une seule ligne pour saisir du texte.

**Boîte à sélection multiple** : Une boîte de liste où plusieurs options peuvent être sélectionnées. Après la création de ce champ, vous devez le modifier pour y ajouter les options de sélection. Voir [Voir/Modifier les valeurs autorisées](#) pour des informations sur la modification des valeurs autorisées.



**Liste déroulante :** Une boîte de liste où une seule option peut-être sélectionnée. Après la création de champ, vous devez le modifier pour y ajouter les options de sélection. Voir [Voir/Modifier les valeurs autorisées](#) pour des informations sur la modification des valeurs autorisées.

**Date/Heure :** Un champ de date. Ce champ apparaît avec un widget de calendrier pour choisir une date.

- *Position :* Un nombre entier qui détermine l'ordre dans lequel seront affichés les champs personnalisés dans l'interface utilisateur, notamment lors de la consultation d'un rapport de bogue. Les champs ayant les valeurs les plus faibles seront affichés en premier.
- *Description de relation réciproque :* Quand le champ personnalisé est de type ID de bogue, vous pouvez saisir du texte ici qui sera utilisé comme libellé dans le bogue référencé pour lister les bogues qui pointent vers celui-ci. Ceci permet d'avoir des relations réciproques entre deux bogues.
- *Peut être défini à la création du bogue :* Un booléen qui détermine si ce champ peut être défini lors de la création du bogue. Si ce n'est pas le cas, vous devrez d'abord créer le bogue pour pouvoir définir ce champ. Sinon, vous pourrez définir sa valeur lors de la création du bogue, voir [Rapporter des bogues](#) à propos de la saisie de bogues.
- *Affiché dans les courriels de bogue pour les nouveaux bogues :* Un booléen qui détermine si la valeur définie dans ce champ doit apparaître dans les courriels de bogues quand un bogue est créé. Cet attribut n'a aucun effet si le champ ne peut pas être défini lors de la création du bogue.
- *Est obsolète :* Un booléen qui détermine si le champ doit être affiché. Les champs personnalisés obsolètes sont cachés.
- *Est obligatoire :* Booléen déterminant si ce champ doit être défini. Pour les champs simples et multiples, ceci signifie qu'une valeur (qui n'est pas par défaut) doit être sélectionnée, et pour les champs date et texte, du texte doit être saisi.
- *Le champ apparaît seulement quand :* Un champ personnalisé peut être rendu visible quand certains critères sont remplis. Par exemple, quand le bogue appartient à un produit donné ou quand le bogue à une certaine gravité. Si ce champ est laissé vide, alors le champ personnalisé sera toujours visible, dans tous les bogues.
- *Champ contrôlant les valeurs qui apparaissent dans ce champ :* Quand le champ personnalisé est de type Liste ou Boîte de sélection multiple, vous pouvez restreindre la disponibilité des valeurs du champ personnalisé en fonction de la valeur d'un autre champ. Ce critère est indépendant du critère utilisé dans le paramètre Le champ apparaît seulement quand :. Par exemple, vous pouvez décider qu'une certaine valeur valeurY est seulement disponible quand l'état du bogue est RÉSOLU alors que la valeur valeurX doit toujours être affichée. Une fois sélectionné le champ qui doit contrôler la disponibilité des valeurs de ce champ personnalisé, vous pouvez modifier les valeurs de ce champ personnalisé pour définir le critère, voir [Voir/Modifier les valeurs autorisées](#).

#### 4.6.2 Modifier les champs personnalisés

Dès qu'un champ personnalisé est créé, son nom et son type ne peuvent pas être modifiés. Si ce champ est une liste déroulanteIf, la liste de ses valeurs peut être définie comme indiqué dans [Voir/Modifier les valeurs autorisées](#). Tous les autres attributs peuvent être modifiés comme décrit ci-dessous.

#### 4.6.3 Supprimer des champs personnalisés

Seuls les champs personnalisés marqués comme obsolètes et qui n'ont jamais été utilisés peuvent être supprimés (si non l'intégrité de l'historique du bogue pourrait être compromise). Pour les champs personnalisés marqués comme obsolètes, un lien Supprimer apparaîtra dans la colonne Action. Si le champ personnalisé a déjà été utilisé auparavant, la suppression sera rejetée. Mais marquer le champ comme obsolète suffira à le masquer totalement de l'interface utilisateur.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 4.7 Valeurs autorisées

Les valeurs autorisées pour le système d'exploitation, la plateforme, les priorité et gravité de bogue, les champs personnalisés de type Liste et Boîte de sélection multiple (voir *Champs personnalisés*), ainsi que la liste des états et résolutions peuvent être personnalisés à partir de la même interface. Vous pouvez ajouter, modifier, désactiver et supprimer les valeurs qui peuvent être utilisées pour ces champs.

### 4.7.1 Voir/Modifier les valeurs autorisées

Modifier les valeurs autorisées nécessite le privilège admin. Cliquez sur Valeurs du champ dans la page d'administration. Une liste de tous les champs, standards et personnalisés, pour lesquels les valeurs autorisées peuvent être modifiées, apparaît. Cliquez sur un nom de champ pour modifier ses valeurs autorisées.

Il n'y a pas de limite au nombre de valeurs qu'un champ peut avoir, mais chaque valeur doit être unique pour ce champ. La position est importante pour afficher ces valeurs dans l'ordre désiré.

Quand la disponibilité des valeurs d'un champ personnalisé est contrôlée par un autre champ, vous pouvez sélectionner ici quelle valeur de l'autre champ doit être définie pour que la valeur du champ personnalisé apparaisse.

### 4.7.2 Supprimer des valeurs autorisées

Les valeurs autorisées de champs personnalisés peuvent être supprimées, mais seulement si les deux conditions suivantes sont respectées :

1. La valeur n'est pas celle utilisée par défaut par le champ.
2. Aucun bogue n'utilise cette valeur.

Si une de ces conditions n'est pas respectée, la valeur ne peut pas être effacée. La seule façon de supprimer ces valeurs consiste à réassigner les bogues pour une autre valeur et de définir une autre valeur par défaut pour le champ.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 4.8 Workflow des états de bogue

Le workflow des états de bogue—les transitions entre les différents états des bogues—peut être personnalisé.

Vous devez commencer par définir les états et les résolutions que vous voulez utiliser (voir *Valeurs autorisées*). Par convention, les états et résolution sont écrits en lettres capitales.

Un seul état, NON CONFIRMÉ, ne peut être renommé ou supprimé. Cependant, il peut être désactivé individuellement pour chaque produit (voir *Catégories, produits, composants, versions et jalons*). L'état indiqué dans le paramètre `duplicate_or_move_bug_status`, s'il est défini, ne peut être supprimé non plus. Pour qu'il le soit, définissez la valeur du paramètre avec un autre état.

À part la valeur vide, deux résolutions, DOUBLONS et CORRIGÉ, ne peuvent être renommés ou supprimés. (CORRIGÉ pourra l'être quand nous aurons corrigé le [bugue 1007605](#).)

Une fois les états à utiliser définis, vous pouvez configurer le cycle des transitions d'états d'un bogue. La page de configuration du workflow affiche tous les états existants deux fois : une fois à gauche pour l'état de départ, et au-dessus pour l'état cibles de la transition. Si la case est cochée, alors la transition de la gauche vers le haut est valide ; sinon, la transition est interdite.

L'état utilisé dans le paramètre `duplicate_or_move_bug_status` (normalement RÉSOLU ou son équivalent) est requis pour être une transition valide à partir de n'importe quel autre état de bogue et est par conséquent appliqué sur la page.

---

Le lien « Voir les commentaires nécessaires pour les transitions d'état » sous le tableau vous permet de définir quelles transitions nécessitent un commentaire de la part de l'utilisateur.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 4.9 Groupes et restrictions de groupes

Les groupes permettent de séparer les bogues en divisions logiques. Les groupes sont typiquement utilisés pour isoler les bogues qui ne devraient être vus que par certaines personnes. Par exemple, une société pourrait créer un groupe différent pour chacun de ses clients ou de ses partenaires. Les permissions de groupe pourraient être définies de sorte que chaque partenaire ou client ne puisse accéder qu'à ses propres bogues. Ou encore, les groupes pourraient être utilisés pour créer des contrôles d'accès variables pour différents départements à l'intérieur d'une organisation. Un autre usage courant des groupes est d'associer des groupes à des produits, créant ainsi une isolation et un contrôle d'accès par produit.

Les groupes et les comportements de groupe sont contrôlés dans plusieurs endroits :

1. La page de configuration des groupes. Pour voir ou modifier des groupes existants, ou pour en créer de nouveaux, cliquez sur le lien [Groupes](#) dans la page [Administration](#). Cette section du manuel traite principalement des aspects des restrictions de groupes accédés sur cette page.
2. Paramètres de configuration globaux. Bugzilla a plusieurs paramètres qui contrôlent le comportement des groupes par défaut et les niveaux de restrictions globaux. Pour plus d'informations sur les paramètres qui contrôlent le comportement global des groupes, consultez [Restrictions de groupe](#).
3. Association de produits avec des groupes. La plupart des fonctionnalités des groupes et de leur sécurité est contrôlée au niveau du produit. Certains aspects des restrictions de groupe pour les produits sont traités dans cette section, mais pour plus de détails, consultez [Affecter des restrictions de groupes à des produits](#).
4. Contrôles d'accès aux groupes pour les utilisateurs. Consultez [Affecter des utilisateurs aux groupes](#) pour des détails sur la façon d'affecter des restrictions de groupe pour les utilisateurs.

Les restrictions de groupe sont tels que, que seuls les membres d'un groupe peuvent voir le bogue. Si un bogue est dans plus d'un groupe, seuls les membres de *tous* les groupes auxquels appartient le bogue, peuvent voir le bogue. Pour des informations pour autoriser un accès en lecture seule à certains utilisateurs et un accès en modification complet à d'autres, consultez [Affecter des restrictions de groupes à des produits](#).

---

**Note :** Par défaut, les bogues peuvent être également vus par le responsable, le rapporteur et par toutes les personnes dans la liste [Copie à](#), sans tenir compte des droits qu'ils auraient normalement pour l'affichage des bogues. La visibilité pour le rapporteur et les personnes de la liste [Copie à](#) peut être être dépassé (bogue par bogue) en modifiant le bogue, en y cherchant la section commençant par [Les utilisateurs des rôles sélectionnés ci-dessous . . .](#) et en retirant la coche de la case située à côté de [Rapporteur](#) ou de [Copie à](#) (ou les deux).

---

### 4.9.1 Créer des groupes

Pour créer un nouveau groupe, réalisez les étapes suivantes :

1. Cliquez sur le lien [Administration](#) dans le pied de page, puis sur le lien [Groupes](#) dans la page d'administration.
2. Un tableau de tous les groupes existants est affiché. Sous le tableau se trouve une description de tous les champs. Pour créer un nouveau groupe, cliquez sur le lien [Ajouter un groupe](#) sous le tableau des groupes existants.

- Il y a cinq champs à remplir. Ces champs sont documentés sous le formulaire. Choisissez un nom et une description pour le groupe. Décidez ensuite si ce groupe doit être utilisé pour les bogues (selon toute vraisemblance, ceci doit être sélectionné). Vous pouvez optionnellement choisir une expression régulière qui ajoutera automatiquement les utilisateurs qui correspondent au groupe et choisir une icône qui aidera à identifier les commentaires d'utilisateurs pour le groupe. L'expression régulière peut être utile, par exemple pour ajouter automatiquement tous les utilisateurs d'une même société dans un groupe (si le groupe est destiné à un client ou un partenaire spécifique).

**Note :** Si le champ Nouvelle expression régulière d'utilisateur est rempli, tous les utilisateurs dont l'adresse électronique correspond à l'expression régulière seront automatiquement membre du groupe tant que leurs adresses correspondent à l'expression régulière. Si leurs adresses changent et ne correspondent plus à l'expression régulière, ils seront retirés du groupe. Les versions 2.16 et précédentes ne retireraient pas automatiquement les utilisateurs dont l'adresse électronique ne correspondait plus à l'expression régulière.

**Avertissement :** Si vous spécifiez un domaine dans l'expression régulière, assurez-vous de terminer l'expression régulière par \$. Sans quoi, en autorisant l'accès à @masociete\\.com, vous autoriserez aussi l'accès à pirate@masociete.com.cracker.net. Vous devrez plutôt utiliser @masociete\\.com\$ comme expression régulière.

- Après la création du groupe, vous pouvez le modifier pour définir des options supplémentaires. La page Modification du groupe permet de spécifier d'autres groupes qui pourraient être inclus dans celui-ci et les groupes autorisés à ajouter ou retirer des utilisateurs de ce groupe. Pour plus de détails, consultez *Modification de groupes et affectation de restrictions*.

### 4.9.2 Modification de groupes et affectation de restrictions

Pour accéder à la page Modification du groupe, cliquez sur le lien Administration dans le pied de page puis cliquez sur le lien Groupes dans la page d'administration. Un tableau de tous les groupes existants est affiché. Cliquez sur le nom du groupe que vous voulez modifier.

La page Modification du groupe contient les mêmes cinq champs présents lors de la création d'un nouveau groupe. Elle contient deux sections supplémentaires Permissions de groupe et Suppression de masse. L'option Suppression de masse supprime simplement tous les utilisateurs qui correspondent à l'expression régulière saisie du groupe. La section Permissions de groupe nécessite plus d'explications.

La section Permissions de groupe dans la page Modification du groupe contient quatre ensembles de permissions qui contrôlent les relations de ce groupe aux autres groupes. Si le paramètre usevisibilitygroups est utilisé (voir *Paramètres*) deux ensembles de permissions supplémentaires sont affichés. Chacun consiste en deux boîtes de sélection. Sur la gauche, une boîte de sélection avec une liste des groupes existants. Sur la droite, une boîte de sélection listant tous les groupes actuellement sélectionnés pour cette permission (cette boîte sera vide pour les nouveaux groupes). La façon dont ces permissions permettent aux groupes d'être en relation avec d'autres est appelée *héritage*. Chacune des six permissions est décrite ci-dessous.

**Groupes qui sont membres de ce groupe** Les membres de tous les groupes sélectionnés ici seront automatiquement membres de ce groupe. En d'autres termes, les membres de tout groupe sélectionné hériteront de l'appartenance à ce groupe.

**Groupes dont ce groupe est membre** Les membres de ce groupe hériteront de l'appartenance à tout groupe sélectionné ici. Par exemple, supposons que le groupe modifié est Admin. S'il y a deux produits (Produit1 et Produit2) et que chaque produit a son propre groupe (Groupe1 et Groupe2), et que le groupe Admin doit avoir accès aux deux produits, sélectionnez simplement Groupe1 et Groupe2 ici.

**Groupes pouvant donner l'appartenance à ce groupe** Les membres de tout groupe sélectionné ici pourront ajouter des utilisateurs à ce groupe, même s'ils ne sont pas membres eux-mêmes de ce groupe.

**Groupes pour lesquels ce groupe peut donner l'appartenance** Les membres de ce groupe peuvent ajouter des utilisateurs à tout groupe sélectionné ici, même s'ils ne sont pas membres eux-mêmes des groupes sélectionnés.

**Groupes pouvant voir ce groupe** Les membres de tout groupe sélectionné peuvent voir les utilisateurs dans ce groupe. Ce paramètre n'est visible que si le paramètre `usevisibilitygroups` est activé dans la page de configuration de Bugzilla. Consultez [Paramètres](#) pour des informations sur la configuration de Bugzilla.

**Groupes que ce groupe peut voir** Les membres de ce groupe peuvent voir les membres de tous les groupes sélectionnés. Ce paramètre n'est visible que si le paramètre `usevisibilitygroups` est activé dans la page de configuration de Bugzilla. Consultez [Paramètres](#) pour des informations sur la configuration de Bugzilla.

### 4.9.3 Affecter des utilisateurs aux groupes

Les utilisateurs peuvent devenir membres d'un groupe de plusieurs manières :

1. L'utilisateur peut être explicitement placé dans le groupe par la modification de son profil. Ceci peut être effectué en accédant à la page [Utilisateurs](#) dans la page [Administration](#). Utilisez le formulaire de recherche pour trouver l'utilisateur dont vous voulez modifier l'appartenance à un groupe, et cliquez sur son adresse électronique dans les résultats de recherche pour modifier son profil. La page du profil liste tous les groupes et indique si l'utilisateur est membre du groupe directement ou indirectement. Vous trouverez plus d'informations sur l'appartenance indirecte à un groupe ci-dessous. Pour plus de détails sur l'administration des utilisateur, consultez [Utilisateurs](#).
2. Le groupe peut inclure un autre groupe dont l'utilisateur est membre. Ceci est indiqué par des crochets autour de la case à cocher à côté du nom du groupe dans le profil de l'utilisateur. Consultez [Modification de groupes et affectation de restrictions](#) pour plus de détails sur l'héritage de groupe.
3. L'adresse électronique de l'utilisateur peut correspondre à une expression régulière spécifiée dans le groupe et qui autorise automatiquement l'appartenance au groupe. Ceci est indiqué par des \* encadrant la case à cocher à côté du nom du groupe dans le profil de l'utilisateur. Consultez [Créer des groupes](#) pour des détails sur l'option d'expression régulière lors de la création de groupe.

### 4.9.4 Affecter des restrictions de groupes à des produits

La fonctionnalité principale des groupes est dérivée des relations des groupes aux produits. Les concepts sur la ségrégation de l'accès aux bogues en utilisant les restrictions de groupes sur les produits peuvent être déroutants. Pour des détails et des exemples sur ce sujet, consultez [Affecter des restrictions de groupes à des produits](#).

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 4.10 Mots-clés

L'administrateur peut définir des mots-clés qui peuvent être utilisés pour étiqueter et catégoriser les bogues. Par exemple, le mot-clé `régression` est utilisé couramment. Une société pourrait avoir comme politique de corriger toutes les régressions dans la version suivante ; ce mot-clé peut permettre alors de tracer ces bogues plus facilement.

Les mots-clés peuvent être créés, modifiés ou supprimés en cliquant sur le lien [Mots-clés](#) dans la page d'administration. Il y a deux champs pour chaque mot-clé : le mot-clé lui-même et une brève description. Une fois créé, les mots-clés peuvent être sélectionnés et appliqués à chaque bogue individuellement dans la section [Détails](#) de chaque bogue.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 4.11 Notifications

Les notifications sont une fonctionnalité de Bugzilla destinée à ennuyer les utilisateurs à des moments spécifiques. En utilisant cette fonctionnalité, les utilisateurs peuvent exécuter des recherches enregistrées à des moments spécifiques (par ex. le 15 du mois à minuit) ou à intervalles réguliers (par ex. toutes les 15 minutes le dimanche). Les résultats des recherches sont envoyés aux utilisateurs sous forme d'un seul courriel ou d'un courriel par bogue, accompagné d'un texte descriptif.

**Avertissement :** Dans cette section, nous supposons que tous les utilisateurs sont membres du groupe `bz_canusewhines`. Cette appartenance est nécessaire pour utiliser le système de notifications. Vous pouvez facilement rendre tous les utilisateurs membres du groupe `bz_canusewhines` en définissant l'expression régulière d'utilisateur suivante : « `.*` » (sans les guillemets).

Le groupe `bz_canusewhineatothers` est également intéressant. Les membres de ce groupe peuvent créer des notifications pour tout utilisateur ou groupe de Bugzilla en utilisant un formulaire avancé dans l'interface de notifications. Les fonctionnalités uniquement disponibles pour les membres du groupe `bz_canusewhineatothers` sont notées aux endroits appropriés.

---

**Note :** Pour que les notifications fonctionnent, un script Perl spécifique doit être exécuté à intervalles réguliers. Plus de détails sont disponibles dans *Notifications*.

---

**Note :** Cette section ne couvre pas l'utilisation du script `whineatnews.pl`. Consulter *Notifications pour les bogues non triés* pour plus de détails sur la programmation des notifications.

---

### 4.11.1 Événement

Le système de notifications définit un « événement » comme une ou plusieurs requêtes exécutées à intervalles réguliers, dont les résultats (s'il y en a) sont envoyées par courriel à l'utilisateur. Les événements sont créés en cliquant sur le bouton « Ajouter un événement ».

Quand le nouvel événement est créé, la première chose à faire est de définir la « Ligne de sujet du courriel ». Le contenu de ce champ sera utilisé dans le sujet de tous les courriels générés par cet événement. En plus du paramétrage de la ligne du sujet, un emplacement est prévu pour saisir une description qui sera incluse au début de chaque message (pour indiquer pourquoi vous recevez ce courriel).

L'étape suivante consiste à indiquer quand l'événement doit être exécuté (« Programmation ») et quelles recherches doivent être faites (« Requetes »).

### 4.11.2 Programmation des notifications

Chaque événement de notification est associé à zéro ou plus programmations. Une programmation est utilisée pour indiquer quand la requête (spécifiée dessous) doit être exécutée. Un nouvel événement est créé sans programmation (ce qui signifie qu'il ne sera jamais exécuté, car il n'est pas programmé). Pour ajouter une programmation, cliquez sur le bouton « Ajouter une nouvelle programmation ».

Chaque programmation utilise un intervalle que vous utilisez pour dire à Bugzilla quand l'événement doit être exécuté. Un événement peut être exécuté certains jours de la semaine, certains jours du mois, pendant les jours ouvrables (du lundi au vendredi) ou chaque jour.

**Avertissement :** Faites attention si vous définissez votre événement pour qu'il soit exécuté le 29, le 30 ou le 31 du mois, car il pourrait ne pas s'exécuter exactement comme prévu. Si vous voulez que votre événement s'exécute le dernier jour du mois, sélectionnez l'intervalle « Le dernier jour du mois ».

Quand vous avez spécifié le ou les jours pendant lesquels l'événement doit être exécuté, vous devez alors indiquer l'heure de l'exécution de l'événement. Vous pouvez exécuter l'événement à une certaine heure du ou des jours spécifiés, ou chaque heure, demie-heure ou quart d'heure du ou des jours spécifiés.

Si une programmation ne s'exécute pas aussi souvent que vous le voudriez, vous pouvez créer une autre programmation pour le même événement. Par exemple, si vous voulez exécuter un événement les jours dont les dates sont divisible par sept, vous devrez ajouter quatre programmations à l'événement, déclenchées les 7, 14, 21 et 28 du mois (un jour par programmation) à l'heure ou à chaque intervalle de temps choisi.

---

**Note :** Si vous êtes membre du groupe `bz_causewhineatothers`, vous aurez alors une option supplémentaire : « Envoyer à ». En utilisant ceci, vous pouvez contrôler qui recevra les courriels générés par cet événement. Vous pouvez choisir d'envoyer les courriels à un seul utilisateur (identifié par l'adresse électronique) ou à un groupe (identifié par le nom du groupe). Pour envoyer les courriels à plusieurs utilisateurs ou groupes, créez une nouvelle programmation pour chaque utilisateur ou groupe supplémentaire.

---

### 4.11.3 Requêtes de notifications

Chaque notification est associée à zéro ou plus requêtes. Une requête est toute recherche enregistrée à exécuter dans la programmation spécifiée (voir plus haut). Un nouvel événement est créé sans requête associée (ce qui signifie que l'événement ne sera jamais exécuté car il n'y aura jamais de résultats à renvoyer). Pour ajouter une requête, cliquez sur le bouton « Ajouter une nouvelle requête ».

Le premier champ à examiner dans votre nouvelle requête ajoutée est le champ « Ordre ». Les requêtes sont exécutées et les résultats inclus dans l'ordre indiqué par le champ « Ordre ». Les requêtes ayant des valeurs faibles pour le champ « Ordre » seront exécutées avant celles ayant des valeurs élevées pour ce champ.

Le champ suivant à examiner est le champ « Requête ». C'est l'endroit où vous choisissez la requête à exécuter. Plutôt que de définir les paramètres de recherche ici, il vous est demandé de choisir dans la liste des recherches enregistrées (la même liste apparaît dans le pied de chaque page de Bugzilla). Vous n'êtes autorisé à choisir que les recherches que vous avez enregistré vous-même (le recherche enregistrée par défaut, « Mes bogues » n'est pas un choix valide). Si vous n'avez pas de recherches enregistrées, profitez de cette opportunité pour en créer une (voir `:ref'list'`).

---

**Note :** Lors de l'exécution des requêtes, le système de notifications agit comme si vous étiez l'utilisateur exécutant la requête. Ce qui signifie que le système de notifications ignorera les bogues correspondant à votre requête pour lesquels vous n'avez pas d'accès.

---

Quand vous avez choisi la recherche enregistrée à exécuter, donnez à la requête un titre descriptif. Ce titre apparaîtra dans le courriel, au-dessus des résultats de la requête. Si vous choisissez « Un message par bogue », le titre de la requête apparaîtra en haut de chaque courriel qui contient un bogue correspondant à votre requête.

Enfin, décidez si les résultats de votre requête doivent être envoyés dans un seul courriel ou si chaque bogue doit apparaître dans son propre courriel.

**Avertissement :** Réfléchissez soigneusement avant de cocher la case « Un message par bogue ». Si vous créez une requête qui correspond à des milliers de bogues, vous recevrez des milliers de courriels !

### 4.11.4 Enregistrement des modifications

Quand vous avez terminé de définir une programmation et de créer au moins une recherche enregistrée, cliquez sur le bouton « Mettre à jour/Appliquer ». Ceci enregistrera votre événement et le rendra disponible pour une exécution immédiate.

---

**Note :** Si vous voulez supprimer un événement, vous pouvez le faire en utilisant le bouton « Supprimer l'événement » dans le coin supérieur droit de chaque événement. Vous pouvez aussi modifier un événement existant en cliquant sur le bouton « Mettre à jour/Appliquer » après avoir terminé vos modifications.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 4.12 Citations

Les citations sont de petits messages qui peuvent être configurés pour apparaître avec les résultats de recherche. Une installation de Bugzilla peut avoir ses propres citations. Quand une citation doit être affichée, une sélection aléatoire est faite sur l'ensemble des citations déjà présentes.

La soumission de citation est contrôlée par le paramètre *quip\_list\_entry\_control*. Plusieurs valeurs sont possibles : open, moderated ou closed. Pour activer l'approbation de citations, vous devez définir ce paramètre à *moderated*. De cette façon, les utilisateurs seront libres de soumettre des citations mais un administrateur doit explicitement les approuver avant qu'elles ne soient effectivement utilisées.

Pour voir les citations dans l'interface utilisateurs, il suffit de cliquer sur une citation lorsqu'elle est affichée avec les résultats de recherche. Ou elles peuvent être atteintes directement dans le navigateur en visitant l'URL `quips.cgi` (préfixée par l'emplacement Web habituel de votre installation Bugzilla). Quand l'interface pour les citations a été activée, il suffit de cliquer sur `Voir et modifier la liste complète des citations` pour voir la page d'administration. Une page recensant toutes les citations disponibles de la base de données sera affichée.

Une case à cocher est située à côté de chaque citation, dans la colonne *Approuvée*. Les citations ayant cette case cochée sont déjà approuvées et apparaîtront avec les résultats de recherche. Celles qui n'ont pas cette case cochée sont encore dans la base de données mais n'apparaîtront pas dans les résultats de recherche. Les citations soumises par les utilisateurs n'ont pas cette case cochée.

Il y a également un lien de suppression à côté de chaque citation qui peut être utilisé pour supprimer définitivement une citation.

L'affichage des citations est contrôlé par la préférence utilisateur *display\_quips*. Les valeurs possibles sont *Activé* et *Désactivé*.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 4.13 Extensions installées

Bugzilla peut être amélioré en utilisant des extensions (voir *Extensions*). Si une extension contient de la documentation dans le format approprié et que vous avez compilé votre propre copie de la documentation de Bugzilla en utilisant `makedocs.pl`, alors la documentation de vos extensions installées apparaîtront [ici](#).

Votre installation Bugzilla dispose des extensions suivantes (lors de votre dernière compilation de la documentation) :

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

---



---

## Guide d'intégration et de personnalisation

---

Vous pouvez trouver que Bugzilla fait déjà ce dont vous avez besoin, il suffit de le configurer correctement. Lisez attentivement les section *Guide d'administration* pour voir si c'est le cas pour vous. Dans le cas contraire, ce chapitre explique comment utiliser les mécanismes disponibles pour l'intégration et la personnalisation.

### 5.1 FAQ sur la personnalisation

Comment puis-je...

- ... **ajouter un nouveau champ à un bogue ?** Utilisez *Champs personnalisés* ou, si vous voulez seulement des nouveaux champs de formulaire lors de la saisie d'un bogue but don't need Bugzilla to track the field seperately thereafter, you can use a *custom bug entry form*.
- ... **changer le nom d'un champ de bogue ?** *Modifiez* la valeur correspondante dans le template `template/en/default/global/field-descs.none.tmpl`.
- ... **utiliser un autre mot que « bogue » pour décrire des bogues ?** *Modifiez* les valeurs correspondantes dans le template `template/en/default/global/variables.none.tmpl`.
- ... **appeler le système autrement que « Bugzilla » ?** *Modifiez* la valeur correspondante dans le template `template/en/default/global/variables.none.tmpl`.
- ... **modifier qui peut changer quoi dans un bogue ?** Consultez *Modifier qui peut changer quoi*.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 5.2 Langues

Les templates de Bugzilla peuvent être traduits, bien que ce soit un [gros travail](#). Si vous disposez d'une jeu de templates traduits pour votre version de Bugzilla, Bugzilla peut gérer plusieurs langues à la fois. Dans ce cas, Bugzilla se base sur l'en-tête HTTP `Accept-Language` du navigateur de l'utilisateur pour décider quelle langue utiliser. Si plusieurs langues sont installées, un menu est affiché sur la page d'accueil en haut à droite permettant à l'utilisateur de sélectionner manuellement une langue différente. S'il le fait, son choix ignorera alors l'en-tête HTTP `Accept-Language`.

Beaucoup de langues peuvent être obtenues dans [la section localisation du site Web de Bugzilla](#). Les instructions pour soumettre de nouvelles langues sont aussi disponibles à cet emplacement. Il y a également une [liste des équipes de localisation](#) ; vous pourriez vouloir contacter quelqu'un pour savoir où en est l'avancement de la traduction des templates dans votre langue.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 5.3 Thèmes

Bugzilla sait gérer les thèmes pour changer l'apparence de l'interface utilisateur. Il en existe deux : « Classic » et « Dusk ». Vous pouvez en trouver d'autres indiqués [sur le wiki](#), et ici deux autres qui font partie de [bugzilla.mozilla.org](http://bugzilla.mozilla.org). Cependant, dans chaque cas, vous devrez vérifier que le thème est supporté dans votre version de Bugzilla.

Pour créer un nouveau thème, créez un répertoire qui contient tous les fichiers CSS contenus dans `skins/standard/`, et ajoutez votre répertoire dans `skins/contrib/`. Ensuite, modifiez les CSS comme vous l'entendez.

Après avoir ajouté votre répertoire, assurez-vous d'exécuter `checksetup.pl` pour que les bonnes uatorisations soient mises sur les fichiers.

Après avoir installé le nouveau thème, il s'affichera comme une option dans les *Préférences* de l'utilisateur, dans l'onglet *Général*. Si vous voulez forcer un thème particulier pour tous les utilisateurs, sélectionnez-le dans *Préférences par défaut* dans l'interface utilisateur *Administration* et désélectionnez « Activé » pour la préférence pour que les utilisateurs ne puissent pas la changer.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 5.4 Templates

Bugzilla utilise un système de templates pour définir son interface utilisateur. Les templates standards peuvent être modifiés, remplacés ou écrasés. Vous pouvez aussi utiliser des crochets de template dans une *extension* pour ajouter ou modifier le comportement des templates en utilisant une interface stable.

### 5.4.1 Structure du répertoire Template

La structure du répertoire de templates débute par le répertoire `template`, qui contient un répertoire pour chaque localisation (langue) installée. Bugzilla fournit par défaut les templates en anglais, le répertoire s'appelle donc `en` et nous parlerons donc du répertoire `template/en` tout au long de cette documentation. Sous le répertoire `template/en` se trouve le répertoire `default` qui contient tous les templates standards fournis par Bugzilla.

**Avertissement :** Il existe aussi un répertoire `data/template`. C'est ici que Template Toolkit place les versions compilées (c'est-à-dire le code Perl) des templates. *Ne modifiez pas* les fichiers dans ce répertoire ou tous vos changements seront perdus lors de la prochaine compilation des templates par Template Toolkit.

### 5.4.2 Choisir une méthode de personnalisation

Si vous voulez modifier les templates de Bugzilla, la première décision à prendre est la méthode pour le faire. Il y a trois possibilités, dépendant principalement de la portée de vos modifications et de la méthode que vous utilisez pour mettre à jour Bugzilla.

1. Vous pouvez éditer directement les templates dans `template/en/default`.
2. Vous pouvez copier les templates à modifier dans un répertoire miroir ayant la même arborescence dans `template/en/custom`. Les templates dans ce répertoire écraseront automatiquement tout template de même nom et situé au même endroit de l'arborescence du répertoire `template/en/default`. (Le répertoire `custom` n'existe pas par défaut et doit être créé si vous voulez l'utiliser).

3. Vous pouvez utiliser les crochets présents dans de nombreux templates pour ajouter des éléments ou modifier l'interface utilisateur à l'aide d'une *extension*. Les crochets ne sont généralement pas supprimés d'une version à l'autre et bénéficient d'une interface stable.

La troisième méthode est la meilleure s'il existe des crochets dans les endroits appropriés and the change you want to do is possible using hooks. It's not very easy et que le changement que vous voulez faire est possible avec les crochets. C'est ce qui est le plus souvent utilisé pour faire des ajouts. Vous pouvez faire des modifications si vous ajoutez du code JS qui fera les modifications lors du chargement de la page. Vous pouvez retirer des éléments de l'interface utilisateur en ajoutant du code CSS pour les masquer.

Contrairement aux crochets de code, il n'est pas nécessaire de documenter les crochets de templates, il suffit d'ouvrir le template et de rechercher `Hook.process`.

S'il n'y a pas de crochet disponible, alors la deuxième méthode de personnalisation doit être utilisée si vous voulez faire des changements majeurs, car il est garanti que le contenu du répertoire `custom` ne sera pas touché pendant une mise à jour, et vous pouvez décider de revenir aux templates standards ou continuer à utiliser les vôtres, ou faire les modifications nécessaires pour être compatible avec la nouvelle version. C'est aussi une bonne méthode pour des fichiers totalement nouveaux ou pour quelques fichiers comme `bug/create/user-message.html.tpl` qui sont conçus pour être entièrement remplacés.

En utilisant la deuxième méthode, votre interface utilisateur peut être corrompue si des changements incompatibles sont faits dans l'interface des templates. Les templates changent régulièrement et par conséquent, ils ne sont pas tous individuellement documentés. Vous devrez alors trouver ce qui a changé et adapter vos templates en conséquence.

Pour des changements mineurs, la commodité de la première méthode est difficile à battre. Lors de la mise à jour, **git** fusionnera vos modifications dans la nouvelle version pour vous. Le revers de la médaille, si la fusion échoue, c'est que Bugzilla ne fonctionnera pas correctement jusqu'à ce que vous ayez corrigé le problème et réintégré votre code.

Vous pouvez aussi voir ce que vous avez changé en utilisant **git diff**, ce qui n'est pas possible si vous avez placé le fichier dans le répertoire `custom`.

### 5.4.3 Comment modifier les templates

---

**Note :** Si vous faites des modifications de template que vous souhaitez soumettre pour être inclus dans les standard de Bugzilla, veuillez lire les sections appropriées du [Guide des développeurs](#).

---

Bugzilla utilise un système de template appelé Template Toolkit. La syntaxe de ce langage dépasse le cadre de ce guide. Il est assez facile à appréhender en consultant les templates de Bugzilla. Vous pouvez aussi consulter le manuel, disponible sur le [site Web de Template Toolkit](#).

Vous devez porter une attention particulière sur la nécessité de filtrer correctement les données HTML passées dans le template. Cela signifie que si des données peuvent contenir un caractère spécial HTML tel que `<` et que les données ne sont pas destinées à être du HTML, elles doivent être converties en entités, par ex. : `&lt;`. Il faut utiliser le filtre `html` de Template Toolkit pour cela (ou le filtre `uri` pour encoder les caractères spéciaux dans les URL). Si vous oubliez de le faire, vous pourriez ouvrir votre installation aux attaques par « cross-site scripting ».

Vous devez exécuter la commande `./checksetup.pl` après avoir modifié des templates. Ne pas le faire peut conduire à ce que vos changements ne soient pas pris en compte ou que les permissions sur les fichiers modifiés ne soient pas correctes et que le serveur Web ne soient pas en mesure de les lire.

### 5.4.4 Format et type de template

Certains CGI sont capables d'utiliser plus d'un template. Par exemple, `buglist.cgi` peut renvoyer deux formats de HTML (complexe et simple). Chacun d'eux dans un template séparé. Le mécanisme qui fournit cette fonctionnalité est extensible : vous pouvez ajouter de nouveaux templates pour ajouter de nouveaux formats.

Vous pourriez utiliser cette fonctionnalité pour par exemple ajouter un formulaire de saisie de bogue personnalisé pour un sous-ensemble particulier d'utilisateurs ou un type de bogue particulier.

Bugzilla peut aussi gérer deux types de sorties. Par exemple, les bogues sont disponibles aux formats HTML et XML, et ce mécanisme est aussi extensible pour ajouter de nouveaux types de contenu. Cependant, au lieu d'utiliser de telles interfaces ou d'étendre Bugzilla pour en ajouter d'autres, il est préférable d'utiliser les *WebService API Reference* pour s'intégrer avec Bugzilla.

Pour voir si un CGI gère plusieurs formats et types de sortie, recherchez dans le fichier CGI `get_format`. Si ce n'est pas présent, ajouter la gestion de plusieurs formats ou types n'est pas très compliqué : regardez dans d'autres fichiers CGI comment c'est mis en œuvre, par exemple : `config.cgi`.

Pour faire un nouveau format de template qui soit géré par un fichier CGI, ouvrez le template pour ce fichier CGI et regardez le commentaire « INTERFACE » (s'il existe). Ce commentaire indique les variables qui sont passées à ce template. S'il n'existe pas, il faudra lire le template et le code pour trouver cette information.

Écrivez votre template avec le langage à balises ou le style de texte approprié.

Vous devez décider du type de contenu que votre template va servir. Les types de contenu sont définis dans le fichier `Bugzilla/Constants.pm` dans la constante `contenttypes`. Si votre type de contenu n'est pas présent, ajoutez-le. Rappelez-vous du marqueur à trois ou quatre lettres affecté à votre type de contenu. Ce marqueur fera partie du nom du filename.

Enregistrez votre nouveau template sous la forme `<stubname>-<formatname>.<contenttypetag>.tmpl`. Essayez votre template en l'appelant avec le fichier CGI : `<cginame>.cgi?format=<formatname>`. Ajoutez `&ctype=<type>` si le type n'est pas HTML.

### 5.4.5 Templates particuliers

Il existe quelques templates qui pourraient vous intéresser pour personnaliser votre installation.

**index.html.tmpl** : Ceci est la page d'accueil de Bugzilla.

**global/header.html.tmpl** : Ceci définit l'en-tête utilisé dans toutes les pages de Bugzilla. L'en-tête comprend la bannière, qui est ce qui est présenté aux utilisateurs et vous voudrez certainement modifier. Cependant, l'en-tête comprend aussi la section « HTML HEAD », et vous pourriez par exemple vouloir y ajouter une feuille de style ou une balise « META » en modifiant l'en-tête.

**global/banner.html.tmpl** : Ceci contient la bannière, la partie de l'en-tête qui apparaît en haut de chaque page de Bugzilla. La bannière par défaut est sobre et vous voudrez peut-être la personnaliser pour donner à votre installation une apparence distinctive. Il est recommandé de conserver le numéro de version de Bugzilla sous une forme ou une autre afin de déterminer la version de Bugzilla exécutée et que les utilisateurs sachent quelle documentation lire.

**global/footer.html.tmpl** : Ceci définit le pied de page de toutes les pages de Bugzilla. Vous pouvez modifier ce fichier pour personnaliser votre installation et lui donner une apparence distinctive.

**global/variables.none.tmpl** : Ceci permet de changer le terme « bogue » pour un autre (par ex. : « ticket ») dans toute l'interface. Il permet aussi de remplacer le nom Bugzilla par un autre (par ex. : « Support utilisateur de Toto Corp. »).

**list/table.html.tmpl** : Ce template contrôle l'apparence des listes de bogues créées par Bugzilla. En modifiant ce template, vous pouvez contrôler pour chaque colonne sa largeur et son titre, la longueur maximale de chaque entrée et le comportement à adopter pour les entrées très longues. Pour les longues listes de bogues, Bugzilla insère un séparateur tous les cent bogues par défaut. Ce comportement est également contrôlé par ce template et la valeur peut être modifiée ici.

**bug/create/user-message.html.tmpl** : C'est le message qui apparaît près du haut de la page de saisie d'un bogue. En le modifiant, vous pouvez indiquer à vos utilisateurs comment il doivent rapporter un bogue.

**bug/process/midair.html.tmpl** : C'est la page utilisée si deux personnes soumettent en même temps une modification sur le même bogue. La deuxième personne soumettant les changements verra cette page lui

indiquant ce que la première personne a modifié et demandera si elle souhaite écraser ces changements ou annuler et revenir sur le bogue. Le titre par défaut et l'en-tête de ce page est : « Collision détectée ! ». Si vous travaillez dans l'industrie aéronautique ou un autre environnement où ce message pourrait paraître inapproprié (oui, on nous a rapporté des histoires vraies à ce sujet) vous voudrez peut-être changer ce message par un autre plus adapté à votre environnement.

**bug/create/create.html.tpl et bug/create/comment.txt.tpl** : Vous ne voulez peut-être pas faire l'effort de créer des champs personnalisés dans Bugzilla, mais vous voulez vous assurer que chaque rapport de bogue contienne un certain nombre d'informations importantes pour lesquelles il n'y a pas de champ spécifique. Le système de saisie de bogue a été conçu de manière extensible pour vous permettre d'ajouter à votre discrétion des widgets HTML, tels que des boîtes de texte ou des listes déroulantes dans la page de saisie de bogues, et avoir leurs valeurs affichées dans le commentaire initial.

Un exemple de ceci est dans le [formulaire de saisie assistée](#). Ce code est fourni dans Bugzilla comme exemple que vous pouvez copier. Il se trouve dans les fichiers `create-guided.html.tpl` et `comment-guided.html.tpl`.

Un champ masqué indiquant le format doit être ajouté dans le formulaire afin de rendre le template fonctionnel. Sa valeur doit être le suffixe du nom du template. Par exemple, si le fichier s'appelle `create-guided.html.tpl`, alors

```
<input type="hidden" name="format" value="guided">
```

est utilisé à l'intérieur du formulaire.

Pour utiliser cette fonctionnalité, créez un template personnalisé pour le fichier `enter_bug.cgi`. Le template par défaut sur lequel vous pourriez vous baser est `default/bug/create/create.html.tpl`. Nommez-le `custom/bug/create/create-<formatname>.html.tpl`, et à l'intérieur ajoutez les champs de saisie du formulaire pour chaque information désirée, comme par exemple, le numéro de version ou les étapes à reproduire.

Ensuite, créez un template basé sur `default/bug/create/comment.txt.tpl` et nommez-le `custom/bug/create/comment-<formatname>.txt.tpl`. Il nécessite deux lignes à placer au début du fichier comme suit :

```
[% USE Bugzilla %]
[% cgi = Bugzilla.cgi %]
```

Ensuite, ce template peut référencer les champs de formulaire que vous avez créés en utilisant la syntaxe `[% cgi.param("field_name") %]`. Quand un rapport de bogue est soumis, le commentaire initial attaché au rapport de bogue sera formaté selon la disposition de ce template.

Par exemple, si votre template personnalisé `enter_bug` a un champ

```
<input type="text" name="buildid" size="30">
```

et que votre fichier `comment.txt.tpl` a

```
[% USE Bugzilla %]
[% cgi = Bugzilla.cgi %]
Build Identifier: [%+ cgi.param("buildid") %]
```

alors quelque chose comme ceci

```
Build Identifier: 20140303
```

apparaîtra dans le commentaire initial.

Le système permet de collecter des données structurées dans les rapports de bogue sans la complexité d'une interface utilisateur contenant un grand nombre de champs personnalisés.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 5.5 Extensions

La meilleure façon de personnaliser Bugzilla est d'utiliser une extension Bugzilla. Les extensions peuvent modifier le code et l'interface utilisateur de Bugzilla et être portées dans les futures versions de Bugzilla avec peu d'effort. Nous maintenons une [liste des extensions disponibles](#) écrites par d'autres développeurs sur notre wiki. Vous devrez vous assurer que l'extension en question fonctionne avec votre version de Bugzilla.

Ou vous pouvez aussi écrire votre propre extension. Consulter la [documentation des extensions de Bugzilla](#) pour l'essentiel des informations sur la façon de les faire. Il est utile aussi de lire la section sur les *Templates*. Il existe une extension d'exemple dans `$BUGZILLA_HOME/extensions/Example/` qui montre comment utiliser tous les crochets de code.

Cette section explique comment réaliser quelques tâches courantes en utilisant les API Extension.

### 5.5.1 Ajouter une nouvelle page à Bugzilla

Il peut exister des situations où il est utile d'ajouter une nouvelle page à Bugzilla qui n'a que très peu ou pas de relation avec les autres pages, ou qui peut-être n'utilise que très peu de données Bugzilla. Une page d'aide ou un rapport personnalisé par exemple. Le meilleur moyen d'y parvenir est d'utiliser `page.cgi` et le crochet `page_before_template`.

### 5.5.2 Modifier les données sur une page existante

Le crochet `template_before_process` peut être utilisé pour personnaliser les données affichées sur une page particulière si vous connaissez le template utilisé. Ce crochet peut accéder à toutes les variables de template avant qu'elles soient envoyées au moteur de template.

### 5.5.3 Ajouter de nouveaux champs aux bogues

Pour ajouter de nouveaux champs sur des bogues, vous devrez faire ce qui suit :

- Ajouter un crochet `install_update_db` pour ajouter les champs en appelant `Bugzilla::Field->create` (seulement si le champ n'existe pas déjà). Cela ressemble à ce qui suit pour un champ simple :

```
my $field = new Bugzilla::Field({ name => $name });
return if $field;

$field = Bugzilla::Field->create({
    name          => $name,
    description   => $description,
    type          => $type,           # À partir de la liste dans Constants.pm
    enter_bug     => 0,
    buglist       => 0,
    custom        => 1,
});
```

- Poussez le nom du champ dans le tableau approprié dans les crochets `bug_columns` et `bug_fields`.
- Si vous voulez des accesseurs directs ou d'autres fonctions sur l'objet, vous aurez besoin d'ajouter un bloc `BEGIN` dans votre fichier `Extension.pm` :

```
BEGIN {
    *Bugzilla::Bug::is_foopy = \&_bug_is_foopy;
}
```

```
...
sub _bug_is_foopy {
    return $_[0]->{'is_foopy'};
}
```

- Vous n’avez pas besoin de modifier `Bugzilla/DB/Schema.pm`.
- Vous pouvez utiliser `bug_end_of_create`, `bug_end_of_create_validators` et `bug_end_of_update` pour créer ou mettre à jour les valeurs pour le nouveau champ.

## 5.5.4 Ajouter de nouveaux champs pour d’autres choses

Si vous ajoutez de nouveaux champs pour un objet autre qu’un bogue, vous devrez descendre un niveau plus bas en vous référant aux instructions ci-dessous :

- Dans `install_update_db`, utilisez `bz_add_column`
- Poussez les colonnes dans `object_columns` et `object_update_columns` au lieu de `bug_columns`.
- Ajoutez les validateurs pour les valeurs dans `object_validators`

Le processus pour ajouter les fonctions accesseurs est le même.

Vous pouvez utiliser les crochets `object_end_of_create`, `object_end_of_create_validators`, `object_end_of_set_all` et `object_end_of_update` pour créer ou mettre à jour les valeurs des champs du nouvel objet que vous avez ajouté. Dans les crochets, vous pouvez vérifier le type de l’objet manipulé et passer outre les objets dont vous n’avez pas besoin. Par exemple, si vous ajoutez un nouveau champ à la table `products` :

```
sub object_end_of_create {
    my ($self, $args) = @_;
    my $class = $args->{'class'};
    my $object = $args->{'object'};
    if ($class->isa('Bugzilla::Product') {
        [...]
    }
}
```

Vous devrez faire ce filtrage pour la plupart des crochets dont les noms commencent par `object_`.

## 5.5.5 Ajouter des panneaux de configuration d’administration

Si vous ajoutez de nouvelles fonctionnalités à Bugzilla, elles peuvent avoir des options ou des paramètres configurables. Vous devez alors ajouter un panneau de configuration pour permettre à un administrateur de définir ces paramètres.

En plus du crochet `config_add_panels`, vous aurez besoin d’un template pour définir les chaînes de l’interface utilisateur pour le panneau. Consultez les templates dans `template/en/default/admin/params` pour des exemples et ajoutez le vôtre dans `template/en/default/admin/params` dans le répertoire de votre extension.

Vous pouvez accéder aux valeurs des paramètres des templates en utilisant :

```
[% Param('param_name') %]
```

et dans votre code, en utilisant :

```
Bugzilla->params->{'param_name'}
```

### 5.5.6 Ajouter des préférences utilisateur

Pour ajouter une nouvelle préférence utilisateur :

- Appelez `add_setting('setting_name', ['some_option', 'another_option'], 'some_option')` dans le crochet `install_before_final_checks`. (Le dernier paramètre est le nom de l'option par défaut).
- Ajoutez des descriptions pour les identifiants pour vos paramètres et choix (`setting_name`, `some_option` etc.) dans le hash défini dans `global/setting-descs.none.tmpl`. Faites ceci dans un template de crochet : `hook/global/setting-descs-settings.none.tmpl`. Votre code peut voir la variable de hash ; ajoutez plus de membres dans celui-ci.
- Pour changer le comportement basé sur votre paramètre, référez-le dans des templates en utilisant `[% user.settings.setting_name.value %]`. Référez-le dans votre code en utilisant `$user->settings->{'setting_name'}->{'value'}`. La valeur sera un des noms de l'option (par ex. : `some_option`).

### 5.5.7 Modifier qui peut changer quoi

Les sociétés ont souvent des règles sur les employés ou les catégories d'employés autorisés à modifier certaines choses dans le système de bogues. Par exemple, seul le responsable QA est autorisé à marquer le bogue comme VÉRIFIÉ. Bugzilla a été conçu pour rendre cela facile en écrivant vos propres règles personnalisées pour définir qui est autorisé à faire certains types de transitions sur les bogues.

Par défaut, les responsables, les responsables QA et les utilisateurs avec les privilèges *editbugs* peuvent modifier tous les champs des bogues, sauf restrictions de groupes (à moins qu'ils ne soient membres des groupes qu'ils essaient de modifier). Les rapporteurs de bogues ont aussi la possibilité de modifier certains champs, mais de manière plus restrictive. Les autres utilisateurs, sans privilèges *editbugs*, ne peuvent pas modifier les bogues, sauf pour ajouter un commentaire ou s'ajouter à la liste « Copie à ».

Parce que ce type de modification est une demande récurrente, nous avons ajouté un crochet spécifique que les *Extensions* peuvent appeler. Ce crochet s'appelle `bug_check_can_change_field` et est documenté dans la [documentation sur les crochets](#).

### 5.5.8 Vérifier la syntaxe

Il n'est pas évident de voir comment vérifier la syntaxe des modules Perl de votre extension, s'il elle en possède. Exécuter la commande `checksetup.pl` peut détecter certaines erreurs, mais les informations données ne sont pas nécessairement compréhensibles.

```
perl -Mlib=lib -MBugzilla -e 'BEGIN { Bugzilla->extensions; } use Bugzilla : :Extension : :Extension-Name : :Class;'
```

(exécuté à partir de `$BUGZILLA_HOME`) est ce que vous voulez pour vérifier la syntaxe.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 5.6 API

Bugzilla dispose de nombreuses API que vous pouvez appeler à partir de votre code pour extraire des informations ou en injecter dans Bugzilla. Certaines sont obsolètes et seront bientôt retirées. Lesquelles utiliser ? La réponse courte : *REST WebService API v1* doit être utilisé pour toutes les nouvelles intégrations, mais gardez un œil sur la version 2 qui sera bientôt publiée.

Les API actuellement disponibles :



### 5.6.1 API ad hoc

Diverses pages de Bugzilla sont disponibles dans des formats classiques comme le HTML. Par exemple, les bogues peuvent être téléchargés sous forme XML et les listes de bogues sous forme CSV. Le format CSV est utile pour les imports de feuille de tableur. Il existe des liens sur les pages HTML pour d'autres formats alternatifs quand ils sont disponibles.

### 5.6.2 XML-RPC

Bugzilla contient une [API XML-RPC](#). Cette API ne sera plus mise à jour et sera retirée dans une future version de Bugzilla.

Point d'entrée : `/xmlrpc.cgi`

### 5.6.3 JSON-RPC

Bugzilla dispose d'une [API JSON-RPC](#). Cette API ne sera plus mise à jour et sera retirée dans une future version de Bugzilla.

Point d'entrée : `/jsonrpc.cgi`

### 5.6.4 REST

Bugzilla dispose d'une [API REST](#) qui est l'API actuellement recommandée pour l'intégration avec Bugzilla. La version actuelle de l'API REST est la version 1. Elle est stable et les mises à jour préserveront la compatibilité descendante.

**C'est l'API actuellement recommandée pour les nouveaux développements.**

Endpoint : `/rest`

### 5.6.5 BzAPI et BzAPI compatible avec REST

La première API REST pour Bugzilla a été mise en œuvre en utilisant un proxy externe appelé [BzAPI](#). Elle est devenue suffisamment populaire pour qu'une API compatible BzAPI soit ajoutée par dessus l'API REST (native) écrite ensuite, pour permettre au code utilisant l'API BzAPI de tirer avantage des améliorations de performances de l'intégration directe sans avoir besoin de tout ré-écrire. Cette extension peut être nécessaire pour votre installation Bugzilla.

Les API BzAPI et BzAPI-compatible API ne seront plus mises à jour et ne doivent pas être utilisées pour de nouveaux développements.

### 5.6.6 REST v2

La future version est la version 2, qui reprendra le meilleur de l'API REST actuelle et de l'API BzAPI. Elle est encore en développement.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

---



---

## WebService API Reference

---

This Bugzilla installation has the following WebService APIs available (as of the last time you compiled the documentation) :

### 6.1 Core API v1

#### 6.1.1 General

This is the standard REST API for external programs that want to interact with Bugzilla. It provides a REST interface to various Bugzilla functions.

#### Basic Information

##### Browsing

If the `Accept` header of a request is set to `text/html` (as it is by an ordinary web browser) then the API will return the JSON data as a HTML page which the browser can display. In other words, you can play with the API using just your browser to see results in a human-readable form. This is a good way to try out the various GET calls, even if you can't use it for POST or PUT.

##### Data Format

The REST API only supports JSON input, and either JSON or JSONP output. So objects sent and received must be in JSON format.

On every request, you must set both the `Accept` and `Content-Type` HTTP headers to the MIME type of the data format you are using to communicate with the API. `Content-Type` tells the API how to interpret your request, and `Accept` tells it how you want your data back. `Content-Type` must be `application/json`. `Accept` can be either that, or `application/javascript` for JSONP. In the latter case, add a `callback` parameter to name your callback.

Parameters may also be passed in as part of the query string for non-GET requests and will override any matching parameters in the request body.

Example request which returns the current version of Bugzilla :

```
GET /rest/version HTTP/1.1
Host: bugzilla.example.com
Accept: application/json
```

Example response :

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{
  "version" : "4.2.9+"
}
```

### Errors

When an error occurs over REST, an object is returned with the key `error` set to `true`.

The error contents look similar to :

```
{
  "error": true,
  "message": "Some message here",
  "code": 123
}
```

### Common Data Types

The Bugzilla API uses the following various types of parameters :

| type      | description   |
|-----------|---|
| int       | Integer.  |
| double    | A floating-point number.  |
| string    | A string.   |
| email     | A string representing an email address. This value, when returned, may be filtered based on if the user is logged in or not.  |
| date      | A specific date. Example format : YYYY-MM-DD.   |
| date-time | A date/time. Timezone should be in UTC unless otherwise noted. Example format : YYYY-MM-DDTHH24:MI:SSZ.   |
| boolean   | true or false.  |
| base64    | A base64-encoded string. This is the only way to transfer binary data via the API.  |
| array     | An array. There may be mixed types in an array. [ and ] are used to represent the beginning and end of arrays.  |
| object    | A mapping of keys to values. Called a “hash”, “dict”, or “map” in some other programming languages. The keys are strings, and the values can be any type. { and } are used to represent the beginning and end of objects. |

Parameters that are required will be displayed in **bold** in the parameters table for each API method.

### Authentication

Some methods do not require you to log in. An example of this is *Get Bug*. However, authenticating yourself allows you to see non-public information, for example, a bug that is not publicly visible.

There are two ways to authenticate yourself :

#### API Keys

You can specify `Bugzilla_api_key` or simply `api_key` as an argument to any call, and you will be logged in as that user if the key is correct and has not been revoked. You can set up an API key by using the ‘API Key’ tab in the Preferences pages.

## Login and Password

You can specify `Bugzilla_login` and `Bugzilla_password` or simply `login` and `password` respectively, as arguments to any call, and you will be logged in as that user if your credentials are correct.

| name                                | type    | description  |
|-------------------------------------|---------|--|
| <b>Bugzilla_login</b>               | string  | A user's login name.   |
| <b>Bugzilla_password</b>            | string  | That user's password.  |
| <code>Bugzilla_restrictlogin</code> | boolean | If true, then your login will only be valid for your IP address. |

The `Bugzilla_restrictlogin` option is only used when you have also specified `Bugzilla_login` and `Bugzilla_password`.

There is also a deprecated method of authentication described below that will be removed in the version after Bugzilla 5.0.

## Bugzilla Tokens

You can use *Login* to log in as a Bugzilla user. This issues a token that you must then use in future calls. Just use the value for `token` and pass as either `Bugzilla_token` or simply `token` as arguments to an API call.

| name                  | type   | description   |
|-----------------------|--------|---|
| <b>Bugzilla_token</b> | string | You can specify this as argument to any call, and you will be logged in as that user if the token is correct. This is the token returned when calling <i>Login</i> mentioned above. |

An error is thrown if you pass an invalid token ; you will need to log in again to get a new token.

Also starting with Bugzilla 5.0, login cookies are no longer returned by *Login* due to security concerns.

## Useful Parameters

Many calls take common arguments. These are documented below and linked from the individual calls where these parameters are used.

### Including Fields

Many calls return an array of objects with various fields in the objects. (For example, *Get Bug* returns a list of bugs that have fields like `id`, `summary`, `creation_time`, etc.)

These parameters allow you to limit what fields are present in the objects, to improve performance or save some bandwidth.

`include_fields` : The (case-sensitive) names of fields in the response data. Only the fields specified in the object will be returned, the rest will not be included. Fields should be comma delimited.

Invalid field names are ignored.

Example request to *Get User* :

```
GET /rest/user/1?include_fields=id,name
```

would return something like :

```
{
  "users" : [
    {
      "id" : 1,
      "name" : "user@domain.com"
    }
  ]
}
```

### Excluding Fields

`exclude_fields` : The (case-sensitive) names of fields in the return value. The fields specified will not be included in the returned hashes. Fields should be comma delimited.

Invalid field names are ignored.

Specifying fields here overrides `include_fields`, so if you specify a field in both, it will be excluded, not included.

Example request to *Get User* :

```
GET /rest/user/1?exclude_fields=name
```

would return something like :

```
{
  "users" : [
    {
      "id" : 1,
      "real_name" : "John Smith"
    }
  ]
}
```

Some calls support specifying “subfields”. If a call states that it supports “subfield” restrictions, you can restrict what information is returned within the first field. For example, if you call *Get Product* with an `include_fields` of `components.name`, then only the component name would be returned (and nothing else). You can include the main field, and exclude a subfield.

There are several shortcut identifiers to ask for only certain groups of fields to be returned or excluded :

| va-lue                 | description   |
|------------------------|---|
| <code>_all</code>      | All possible fields are returned if this is specified in <code>include_fields</code> .  |
| <code>_de-fault</code> | Default fields are returned if <code>include_fields</code> is empty or this is specified. This is useful if you want the default fields in addition to a field that is not normally returned.   |
| <code>_ex-tra</code>   | Extra fields are not returned by default and need to be manually specified in <code>include_fields</code> either by exact field name, or adding <code>_extra</code> .   |
| <code>_cus-tom</code>  | Custom fields are normally returned by default unless this is added to <code>exclude_fields</code> . Also you can use it in <code>include_fields</code> if for example you want specific field names plus all custom fields. Custom fields are normally only relevant to bug objects. |

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 6.1.2 Attachments

The Bugzilla API for creating, changing, and getting the details of attachments.

### Get Attachment

This allows you to get data about attachments, given a list of bugs and/or attachment IDs. Private attachments will only be returned if you are in the appropriate group or if you are the submitter of the attachment.

#### Request

To get all current attachments for a bug :

GET /rest/bug/(bug\_id)/attachment

To get a specific attachment based on attachment ID :

GET /rest/bug/attachment/(attachment\_id)

One of the below must be specified.

| name                 | type | description            |
|----------------------|------|------------------------|
| <b>bug_id</b>        | int  | Integer bug ID.        |
| <b>attachment_id</b> | int  | Integer attachment ID. |

### Response

```
{
  "bugs" : {
    "1345" : [
      { (attachment) },
      { (attachment) }
    ],
    "9874" : [
      { (attachment) },
      { (attachment) }
    ],
  ],
  "attachments" : {
    "234" : { (attachment) },
    "123" : { (attachment) },
  }
}
```

An object containing two elements : `bugs` and `attachments`.

The attachments for the bug that you specified in the `bug_id` argument in input are returned in `bugs` on output. `bugs` is a object that has integer bug IDs for keys and the values are arrays of objects as attachments. (Fields for attachments are described below.)

For the attachment that you specified directly in `attachment_id`, they are returned in `attachments` on output. This is a object where the attachment ids point directly to objects describing the individual attachment.

The fields for each attachment (where it says `(attachment)` in the sample response above) are :

| name             | type      | description   |
|------------------|-----------|---|
| data             | base64    | The raw data of the attachment, encoded as Base64.  |
| size             | int       | The length (in bytes) of the attachment.  |
| creation_time    | date-time | The time the attachment was created.  |
| last_change_time | date-time | The last time the attachment was modified.  |
| id               | int       | The numeric ID of the attachment.   |
| bug_id           | int       | The numeric ID of the bug that the attachment is attached to.   |
| file_name        | string    | The file name of the attachment.  |
| summary          | string    | A short string describing the attachment.   |
| content_type     | string    | The MIME type of the attachment.  |
| is_private       | boolean   | true if the attachment is private (only visible to a certain group called the “insidergroup”, false otherwise.  |
| is_obsolete      | boolean   | true if the attachment is obsolete, false otherwise.  |
| is_patch         | boolean   | true if the attachment is a patch, false otherwise.   |
| creator          | string    | The login name of the user that created the attachment.   |
| flags            | array     | Array of objects, each containing the information about the flag currently set for each attachment. Each flag object contains items described in the Flag object below. |

Flag object :

| name              | type      | description   |
|-------------------|-----------|---|
| id                | int       | The ID of the flag.   |
| name              | string    | The name of the flag.   |
| type_id           | int       | The type ID of the flag.  |
| creation_date     | date-time | The timestamp when this flag was originally created.  |
| modification_date | date-time | The timestamp when the flag was last modified.  |
| status            | string    | The current status of the flag such as ?, +, or -.  |
| setter            | string    | The login name of the user who created or last modified the flag.   |
| requestee         | string    | The login name of the user this flag has been requested to be granted or denied. Note, this field is only returned if a requestee is set. |

## Create Attachment

This allows you to add an attachment to a bug in Bugzilla.

### Request

To create attachment on a current bug :

```
POST /rest/bug/(bug_id)/attachment
```

```
{
  "ids" : [ 35 ],
  "is_patch" : true,
  "comment" : "This is a new attachment comment",
  "summary" : "Test Attachment",
  "content_type" : "text/plain",
```



```

"data" : "(Some patch content)",
"file_name" : "test_attachment.patch",
"obsoletes" : [],
"is_private" : false,
"flags" : [
  {
    "name" : "review",
    "status" : "?",
    "requestee" : "user@bugzilla.org",
    "new" : true
  }
]
}

```

The params to include in the POST body, as well as the returned data format, are the same as below. The `bug_id` param will be overridden as it is pulled from the URL path.

| name                | type    | description   |
|---------------------|---------|---|
| <b>ids</b>          | array   | The IDs or aliases of bugs that you want to add this attachment to. The same attachment and comment will be added to all these bugs.  |
| <b>data</b>         | string  | The content of the attachment. If the content of the attachment is not ASCII text such as <code>application/octet-stream</code> you must encode it in base64 using an appropriate client library such as <code>MIME::Base64</code> for Perl.  |
| <b>file_name</b>    | string  | The “file name” that will be displayed in the UI for this attachment and also downloaded copies will be given.  |
| <b>summary</b>      | string  | A short string describing the attachment.   |
| <b>content_type</b> | string  | The MIME type of the attachment, like <code>text/plain</code> or <code>image/png</code> .   |
| <b>comment</b>      | string  | A comment to add along with this attachment.  |
| <b>is_patch</b>     | boolean | <code>true</code> if Bugzilla should treat this attachment as a patch. If you specify this, you do not need to specify a <code>content_type</code> . The <code>content_type</code> of the attachment will be forced to <code>text/plain</code> . Defaults to <code>false</code> if not specified. |
| <b>is_private</b>   | boolean | <code>true</code> if the attachment should be private (restricted to the “insidergroup”), <code>false</code> if the attachment should be public. Defaults to <code>false</code> if not specified.   |
| <b>flags</b>        | array   | Flags objects to add to the attachment. The object format is described in the Flag object below.  |

Flag object :

To create a flag, at least the `status` and the `type_id` or `name` must be provided. An optional `requestee` can be passed if the flag type is requestable to a specific user.

| name             | type   | description  |
|------------------|--------|--|
| <b>name</b>      | string | The name of the flag type.   |
| <b>type_id</b>   | int    | The internal flag type ID.   |
| <b>status</b>    | string | The flags new status (i.e. “?”, “+”, “-” or “X” to clear a flag).              |
| <b>requestee</b> | string | The login of the requestee if the flag type is requestable to a specific user. |

### Response

```

{
  "ids" : [
    "2797"
  ]
}

```

| name | type  | description             |
|------|-------|-------------------------|
| ids  | array | Attachment IDs created. |

## Update Attachment

This allows you to update attachment metadata in Bugzilla.

### Request

To update attachment metadata on a current attachment :

PUT /rest/bug/attachment/(attachment\_id)

```
{
  "ids" : [ 2796 ],
  "summary" : "Test XML file",
  "comment" : "Changed this from a patch to a XML file",
  "content_type" : "text/xml",
  "is_patch" : 0
}
```

| name          | type  | description                                    |
|---------------|-------|--|
| attachment_id | int   | Integer attachment ID.                         |
| ids           | array | The IDs of the attachments you want to update. |

| name         | type    | description  |
|--------------|---------|--|
| file_name    | string  | The “file name” that will be displayed in the UI for this attachment.  |
| summary      | string  | A short string describing the attachment.  |
| comment      | string  | An optional comment to add to the attachment’s bug.  |
| content_type | string  | The MIME type of the attachment, like <code>text/plain</code> or <code>image/png</code> .  |
| is_patch     | boolean | <code>true</code> if Bugzilla should treat this attachment as a patch. If you specify this, you do not need to specify a <code>content_type</code> . The <code>content_type</code> of the attachment will be forced to <code>text/plain</code> . |
| is_private   | boolean | <code>true</code> if the attachment should be private (restricted to the “insidergroup”), <code>false</code> if the attachment should be public.   |
| is_obsolete  | boolean | <code>true</code> if the attachment is obsolete, <code>false</code> otherwise.   |
| flags        | array   | An array of Flag objects with changes to the flags. The object format is described in the Flag object below.   |

Flag object :

The following values can be specified. At least the `status` and one of `type_id`, `id`, or `name` must be specified. If a `type_id` or `name` matches a single currently set flag, the flag will be updated unless `new` is specified.

| name                | type    | description  |
|---------------------|---------|--|
| name                | string  | The name of the flag that will be created or updated.  |
| type_id             | int     | The internal flag type ID that will be created or updated. You will need to specify the <code>type_id</code> if more than one flag type of the same name exists. |
| status              | string  | The flags new status (i.e. "?" , "+" , "-" or "X" to clear a flag).  |
| re-<br>ques-<br>tee | string  | The login of the requestee if the flag type is requestable to a specific user.   |
| id                  | int     | Use ID to specify the flag to be updated. You will need to specify the <code>id</code> if more than one flag is set of the same name.                            |
| new                 | boolean | Set to true if you specifically want a new flag to be created.   |

**Response**

```
{
  "attachments" : [
    {
      "changes" : {
        "content_type" : {
          "added" : "text/xml",
          "removed" : "text/plain"
        },
        "is_patch" : {
          "added" : "0",
          "removed" : "1"
        },
        "summary" : {
          "added" : "Test XML file",
          "removed" : "test patch"
        }
      },
      "id" : 2796,
      "last_change_time" : "2014-09-29T14:41:53Z"
    }
  ]
}
```

`attachments` (array) Change objects with the following items :

| name             | type     | description   |
|------------------|----------|---|
| id               | int      | The ID of the attachment that was updated.  |
| last_change_time | datetime | The exact time that this update was done at, for this attachment. If no update was done (that is, no fields had their values changed and no comment was added) then this will instead be the last time the attachment was updated.  |
| changes          | object   | The changes that were actually done on this attachment. The keys are the names of the fields that were changed, and the values are an object with two items : <ul style="list-style-type: none"> <li>— added : (string) The values that were added to this field. Possibly a comma-and-space-separated list if multiple values were added.</li> <li>— removed : (string) The values that were removed from this field.</li> </ul> |

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 6.1.3 Bugs

The REST API for creating, changing, and getting the details of bugs.

This part of the Bugzilla REST API allows you to file new bugs in Bugzilla and to get information about existing bugs.

#### Get Bug

Gets information about particular bugs in the database.

##### Request

To get information about a particular bug using its ID or alias :

```
GET /rest/bug/(id_or_alias)
```

You can also use *Search Bugs* to return more than one bug at a time by specifying bug IDs as the search terms.

```
GET /rest/bug?id=12434,43421
```

| name        | type  | description                              |
|-------------|-------|--|
| id_or_alias | mixed | An integer bug ID or a bug alias string. |

##### Response

```

{
  "faults": [],
  "bugs": [
    {
      "assigned_to_detail": {
        "id": 2,
        "real_name": "Test User",
        "name": "user@bugzilla.org",
        "email": "user@bugzilla.org"
      },
      "flags": [
        {
          "type_id": 11,
          "modification_date": "2014-09-28T21:03:47Z",
          "name": "blocker",
          "status": "?",
          "id": 2906,
          "setter": "user@bugzilla.org",
          "creation_date": "2014-09-28T21:03:47Z"
        }
      ],
      "resolution": "INVALID",
      "id": 35,
      "qa_contact": "",
      "version": "1.0",
      "status": "RESOLVED",
      "creator": "user@bugzilla.org",
      "cf_drop_down": "---",
      "summary": "test bug",
      "last_change_time": "2014-09-23T19:12:17Z",
      "platform": "All",
      "url": "",
      "classification": "Unclassified",
      "cc_detail": [
        {
          "id": 786,
          "real_name": "Foo Bar",
          "name": "foo@bar.com",
          "email": "foo@bar.com"
        }
      ],
      "priority": "P1",
      "is_confirmed": true,
      "creation_time": "2000-07-25T13:50:04Z",
      "assigned_to": "user@bugzilla.org",
      "flags": [],
      "alias": [],
      "cf_large_text": "",
      "groups": [],
      "op_sys": "All",
      "cf_bug_id": null,
      "depends_on": [],
      "is_cc_accessible": true,
      "is_open": false,
      "cf_qa_list_4": "---",
      "keywords": [],
      "cc": [
        "foo@bar.com",

```

```

    ],
    "see_also": [],
    "deadline": null,
    "is_creator_accessible": true,
    "whiteboard": "",
    "dupe_of": null,
    "target_milestone": "---",
    "cf_multiple_select": [],
    "component": "SaltSprinkler",
    "severity": "critical",
    "cf_date": null,
    "product": "FoodReplicator",
    "creator_detail": {
      "id": 28,
      "real_name": "hello",
      "name": "user@bugzilla.org",
      "email": "namachi@netscape.com"
    },
    "cf_free_text": "",
    "blocks": []
  }
]
}

```

bugs (array) Each bug object contains information about the bugs with valid ids containing the following items :

These fields are returned by default or by specifying `_default` in `include_fields`.

| name                  | type     | description   |
|-----------------------|----------|---|
| actual_time           | double   | The total number of hours that this bug has taken so far. If you are not in the time-tracking group,              |
| alias                 | array    | The unique aliases of this bug. An empty array will be returned if this bug has no aliases.                       |
| assigned_to           | string   | The login name of the user to whom the bug is assigned.   |
| assigned_to_detail    | object   | An object containing detailed user information for the <code>assigned_to</code> . To see the keys included in the |
| blocks                | array    | The IDs of bugs that are “blocked” by this bug.   |
| cc                    | array    | The login names of users on the CC list of this bug.  |
| cc_detail             | array    | Array of objects containing detailed user information for each of the cc list members. To see the k               |
| classification        | string   | The name of the current classification the bug is in.   |
| component             | string   | The name of the current component of this bug.  |
| creation_time         | datetime | When the bug was created.   |
| creator               | string   | The login name of the person who filed this bug (the reporter).   |
| creator_detail        | object   | An object containing detailed user information for the creator. To see the keys included in the user              |
| deadline              | string   | The day that this bug is due to be completed, in the format YYYY-MM-DD.   |
| depends_on            | array    | The IDs of bugs that this bug “depends on”.   |
| dupe_of               | int      | The bug ID of the bug that this bug is a duplicate of. If this bug isn’t a duplicate of any bug, this w           |
| estimated_time        | double   | The number of hours that it was estimated that this bug would take. If you are not in the time-track              |
| flags                 | array    | An array of objects containing the information about flags currently set for the bug. Each flag obje              |
| groups                | array    | The names of all the groups that this bug is in.  |
| id                    | int      | The unique numeric ID of this bug.  |
| is_cc_accessible      | boolean  | If true, this bug can be accessed by members of the CC list, even if they are not in the groups the b             |
| is_confirmed          | boolean  | true if the bug has been confirmed. Usually this means that the bug has at some point been move                   |
| is_open               | boolean  | true if this bug is open, false if it is closed.  |
| is_creator_accessible | boolean  | If true, this bug can be accessed by the creator of the bug, even if they are not a member of the g               |
| keywords              | array    | Each keyword that is on this bug.   |
| last_change_time      | datetime | When the bug was last changed.  |

TABLE 6.1 – Suite de la page précédente

| name              | type   | description  |
|-------------------|--------|--|
| op_sys            | string | The name of the operating system that the bug was filed against.   |
| platform          | string | The name of the platform (hardware) that the bug was filed against.                                      |
| priority          | string | The priority of the bug.   |
| product           | string | The name of the product this bug is in.  |
| qa_contact        | string | The login name of the current QA Contact on the bug.   |
| qa_contact_detail | object | An object containing detailed user information for the qa_contact. To see the keys included in the ti    |
| remaining_time    | double | The number of hours of work remaining until work on this bug is complete. If you are not in the ti       |
| resolution        | string | The current resolution of the bug, or an empty string if the bug is open.                                |
| see_also          | array  | The URLs in the See Also field on the bug.   |
| severity          | string | The current severity of the bug.   |
| status            | string | The current status of the bug.   |
| summary           | string | The summary of this bug.   |
| target_milestone  | string | The milestone that this bug is supposed to be fixed by, or for closed bugs, the milestone that it was    |
| update_token      | string | The token that you would have to pass to the <code>process_bug.cgi</code> page in order to update this b |
| url               | string | A URL that demonstrates the problem described in the bug, or is somehow related to the bug repo          |
| version           | string | The version the bug was reported against.  |
| whiteboard        | string | The value of the “status whiteboard” field on the bug.   |

#### Custom fields :

Every custom field in this installation will also be included in the return value. Most fields are returned as strings. However, some field types have different return values.

Normally custom fields are returned by default similar to normal bug fields or you can specify only custom fields by using `_custom` in `include_fields`.

#### Extra fields :

These fields are returned only by specifying `_extra` or the field name in `include_fields`.

| name | type  | description  |
|------|-------|--|
| tags | array | Each array item is a tag name. Note that tags are personal to the currently logged in user and are not the same as comment tags. |

#### User object :

| name      | type   | description   |
|-----------|--------|---|
| id        | int    | The user ID for this user.  |
| real_name | string | The ‘real’ name for this user, if any.                                  |
| name      | string | The user’s Bugzilla login.  |
| email     | string | The user’s email address. Currently this is the same value as the name. |

#### Flag object :

| name              | type     | description   |
|-------------------|----------|---|
| id                | int      | The ID of the flag.   |
| name              | string   | The name of the flag.   |
| type_id           | int      | The type ID of the flag.  |
| creation_date     | datetime | The timestamp when this flag was originally created.  |
| modification_date | datetime | The timestamp when the flag was last modified.  |
| status            | string   | The current status of the flag.   |
| setter            | string   | The login name of the user who created or last modified the flag.   |
| requestee         | string   | The login name of the user this flag has been requested to be granted or denied. Note, this field is only returned if a requestee is set. |

Custom field object :

You can specify to only return custom fields by specifying `_custom` or the field name in `include_fields`.

- Bug ID Fields : (int)
- Multiple-Selection Fields : (array of strings)
- Date/Time Fields : (datetime)

## Bug History

Gets the history of changes for particular bugs in the database.

### Request

To get the history for a specific bug ID :

```
GET /rest/bug/(id)/history
```

To get the history for a bug since a specific date :

```
GET /rest/bug/(id)/history?new_since=YYYY-MM-DD
```

| name      | type     | description                                      |
|-----------|----------|--|
| id        | mixed    | An integer bug ID or alias.                      |
| new_since | datetime | A datetime timestamp to only show history since. |

### Response

```
{
  "bugs": [
    {
      "alias": [],
      "history": [
        {
          "when": "2014-09-23T19:12:17Z",
          "who": "user@bugzilla.org",
          "changes": [
            {
              "added": "P1",
              "field_name": "priority",
              "removed": "P2"
            },
            {
              "removed": "blocker",
              "field_name": "severity",
              "added": "critical"
            }
          ]
        }
      ]
    }
  ]
}
```





Unless otherwise specified in the description of a parameter, bugs are returned if they match *exactly* the criteria you specify in these parameters. That is, we don't match against substrings—if a bug is in the “Widgets” product and you ask for bugs in the “Widg” product, you won't get anything.

Criteria are joined in a logical AND. That is, you will be returned bugs that match *all* of the criteria, not bugs that match *any* of the criteria.

Each parameter can be either the type it says, or a list of the types it says. If you pass an array, it means “Give me bugs with *any* of these values.” For example, if you wanted bugs that were in either the “Foo” or “Bar” products, you'd pass :

```
GET /rest/bug?product=Foo&product=Bar
```

Some Bugzillas may treat your arguments case-sensitively, depending on what database system they are using. Most commonly, though, Bugzilla is not case-sensitive with the arguments passed (because MySQL is the most-common database to use with Bugzilla, and MySQL is not case sensitive).

In addition to the fields listed below, you may also use criteria that is similar to what is used in the Advanced Search screen of the Bugzilla UI. This includes fields specified by `Search by Change History` and `Custom Search`. The easiest way to determine what the field names are and what format Bugzilla expects is to first construct your query using the Advanced Search UI, execute it and use the query parameters in they URL as your query for the REST call.

| name             | type      | description  |
|------------------|-----------|--|
| alias            | array     | The unique aliases of this bug. An empty array will be returned if this bug has no aliases.  |
| assigned_to      | string    | The login name of a user that a bug is assigned to.  |
| component        | string    | The name of the Component that the bug is in. Note that if there are multiple Components with the same name, and you search for that name, bugs in <i>all</i> those Components will be returned. If you don't want this, be sure to also specify the <code>product</code> argument.  |
| creation_time    | date-time | Searches for bugs that were created at this time or later. May not be an array.  |
| creator          | string    | The login name of the user who created the bug. You can also pass this argument with the name <code>reporter</code> , for backwards compatibility with older Bugzillas.  |
| id               | int       | The numeric ID of the bug.   |
| last_change_time | date-time | Searches for bugs that were modified at this time or later. May not be an array.   |
| limit            | int       | Limit the number of results returned. If the limit is more than zero and higher than the maximum limit set by the administrator, then the maximum limit will be used instead. If you set the limit equal to zero, then all matching results will be returned instead.  |
| offset           | int       | Used in conjunction with the <code>limit</code> argument, <code>offset</code> defines the starting position for the search. For example, given a search that would return 100 bugs, setting <code>limit</code> to 10 and <code>offset</code> to 10 would return bugs 11 through 20 from the set of 100.  |
| op_sys           | string    | The "Operating System" field of a bug.   |
| platform         | string    | The Platform (sometimes called "Hardware") field of a bug.   |
| priority         | string    | The Priority field on a bug.   |
| product          | string    | The name of the Product that the bug is in.  |
| resolution       | string    | The current resolution—only set if a bug is closed. You can find open bugs by searching for bugs with an empty resolution.   |
| severity         | string    | The Severity field on a bug.   |
| status           | string    | The current status of a bug (not including its resolution, if it has one, which is a separate field above).  |
| summary          | string    | Searches for substrings in the single-line Summary field on bugs. If you specify an array, then bugs whose summaries match <i>any</i> of the passed substrings will be returned. Note that unlike searching in the Bugzilla UI, substrings are not split on spaces. So searching for <code>foo bar</code> will match "This is a foo bar" but not "This foo is a bar". <code>['foo', 'bar']</code> , would, however, match the second item. |
| tags             | string    | Searches for a bug with the specified tag. If you specify an array, then any bugs that match <i>any</i> of the tags will be returned. Note that tags are personal to the currently logged in user.   |
| target_milestone | string    | The Target Milestone field of a bug. Note that even if this Bugzilla does not have the Target Milestone field enabled, you can still search for bugs by Target Milestone. However, it is likely that in that case, most bugs will not have a Target Milestone set (it defaults to "—" when the field isn't enabled).   |
| qa_contact       | string    | The login name of the bug's QA Contact. Note that even if this Bugzilla does not have the QA Contact field enabled, you can still search for bugs by QA Contact (though it is likely that no bug will have a QA Contact set, if the field is disabled).  |
| url              | string    | The "URL" field of a bug.  |
| version          | string    | The Version field of a bug.  |
| whiteboard       | string    | Search the "Status Whiteboard" field on bugs for a substring. Works the same as the <code>summary</code> field described above, but searches the Status Whiteboard field.  |
| quick-search     | string    | Search for bugs using quicksearch syntax.  |

## Response

The same as *Get Bug*.

### Create Bug

This allows you to create a new bug in Bugzilla. If you specify any invalid fields, an error will be thrown stating which field is invalid. If you specify any fields you are not allowed to set, they will just be set to their defaults or ignored.

You cannot currently set all the items here that you can set on `enter_bug.cgi`.

The `WebService` interface may allow you to set things other than those listed here, but realize that anything undocumented here may likely change in the future.

### Request

To create a new bug in Bugzilla.

POST `/rest/bug`

```
{
  "product" : "TestProduct",
  "component" : "TestComponent",
  "version" : "unspecified",
  "summary" : "'This is a test bug - please disregard",
  "alias" : "SomeAlias",
  "op_sys" : "All",
  "priority" : "P1",
  "rep_platform" : "All"
}
```

Some params must be set, or an error will be thrown. These params are marked in **bold**.

Some parameters can have defaults set in Bugzilla, by the administrator. If these parameters have defaults set, you can omit them. These parameters are marked (defaulted).

Clients that want to be able to interact uniformly with multiple Bugzillas should always set both the params marked required and those marked (defaulted), because some Bugzillas may not have defaults set for (defaulted) parameters, and then this method will throw an error if you don't specify them.

| name               | type    | description   |
|--------------------|---------|---|
| <b>product</b>     | string  | The name of the product the bug is being filed against.   |
| <b>component</b>   | string  | The name of a component in the product above.   |
| <b>summary</b>     | string  | A brief description of the bug being filed.   |
| <b>version</b>     | string  | A version of the product above ; the version the bug was found in.  |
| description        | string  | (defaulted) The initial description for this bug. Some Bugzilla installations require this to not be blank.   |
| op_sys             | string  | (defaulted) The operating system the bug was discovered on.   |
| platform           | string  | (defaulted) What type of hardware the bug was experienced on.   |
| priority           | string  | (defaulted) What order the bug will be fixed in by the developer, compared to the developer's other bugs.   |
| severity           | string  | (defaulted) How severe the bug is.  |
| alias              | array   | One or more brief aliases for the bug that can be used instead of a bug number when accessing this bug. Must be unique in all of this Bugzilla.   |
| assigned_to        | string  | A user to assign this bug to, if you don't want it to be assigned to the component owner.   |
| cc                 | array   | An array of usernames to CC on this bug.  |
| comment_is_private | boolean | If set to true, the description is private, otherwise it is assumed to be public.   |
| groups             | array   | An array of group names to put this bug into. You can see valid group names on the Permissions tab of the Preferences screen, or, if you are an administrator, in the Groups control panel. If you don't specify this argument, then the bug will be added into all the groups that are set as being "Default" for this product. (If you want to avoid that, you should specify <code>groups</code> as an empty array.) |
| qa_contact         | string  | If this installation has QA Contacts enabled, you can set the QA Contact here if you don't want to use the component's default QA Contact.  |
| status             | string  | The status that this bug should start out as. Note that only certain statuses can be set on bug creation.   |
| resolution         | string  | If you are filing a closed bug, then you will have to specify a resolution. You cannot currently specify a resolution of <code>DUPLICATE</code> for new bugs, though. That must be done with <i>Update Bug</i> .  |
| target_milestone   | string  | A valid target milestone for this product.  |
| flags              | array   | Flags objects to add to the bug. The object format is described in the Flag object below.   |

Flag object :

To create a flag, at least the `status` and the `type_id` or `name` must be provided. An optional requestee can be passed if the flag type is requestable to a specific user.

| name      | type   | description  |
|-----------|--------|--|
| name      | string | The name of the flag type.   |
| type_id   | int    | The internal flag type ID.   |
| status    | string | The flags new status (i.e. "??", "+", "-" or "X" to clear flag).               |
| requestee | string | The login of the requestee if the flag type is requestable to a specific user. |

In addition to the above parameters, if your installation has any custom fields, you can set them just by passing in the name of the field and its value as a string.

### Response

```
{
  "id" : 12345
```

}

| name | type | description                            |
|------|------|--|
| id   | int  | This is the ID of the newly-filed bug. |

## Update Bug

Allows you to update the fields of a bug. Automatically sends emails out about the changes.

### Request

To update the fields of a current bug.

```
PUT /rest/bug/(id_or_alias)
```

```
{
  "ids" : [35],
  "status" : "IN_PROGRESS",
  "keywords" : {
    "add" : ["funny", "stupid"]
  }
}
```

The params to include in the PUT body as well as the returned data format, are the same as below. You can specify the ID or alias of the bug to update either in the URL path and/or in the `ids` param. You can use both and they will be combined so you can edit more than one bug at a time.

| name               | type  | description   |
|--------------------|-------|---|
| <b>id_or_alias</b> | mixed | An integer bug ID or alias.                             |
| <b>ids</b>         | array | The IDs or aliases of the bugs that you want to modify. |

All following fields specify the values you want to set on the bugs you are updating.

| name                       | type   | description   |
|----------------------------|--------|---|
| alias                      | object | <p>These specify the aliases of a bug that can be used instead of a bug number when accessing this bug. To set these, you should pass a hash as the value. The object may contain the following items :</p> <ul style="list-style-type: none"> <li>— <code>add</code> (array) Aliases to add to this field.</li> <li>— <code>remove</code> (array) Aliases to remove from this field. If the aliases are not already in the field, they will be ignored.</li> <li>— <code>set</code> (array) An exact set of aliases to set this field to, overriding the current value. If you specify <code>set</code>, then <code>add</code> and <code>remove</code> will be ignored.</li> </ul> <p>You can only set this if you are modifying a single bug. If there is more than one bug specified in <code>ids</code>, passing in a value for <code>alias</code> will cause an error to be thrown.</p> <p>For backwards compatibility, you can also specify a single string. This will be treated as if you specified the <code>set</code> key above.</p> |
| assigned_to                | string | The full login name of the user this bug is assigned to.  |
| blocks                     | object | (Same as <code>depends_on</code> below)   |
| depends_on                 | object | <p>These specify the bugs that this bug blocks or depends on, respectively. To set these, you should pass an object as the value. The object may contain the following items :</p> <ul style="list-style-type: none"> <li>— <code>add</code> (array) Bug IDs to add to this field.</li> <li>— <code>remove</code> (array) Bug IDs to remove from this field. If the bug IDs are not already in the field, they will be ignored.</li> <li>— <code>set</code> (array of) An exact set of bug IDs to set this field to, overriding the current value. If you specify <code>set</code>, then <code>add</code> and <code>remove</code> will be ignored.</li> </ul>   |
| Suite sur la page suivante |        |   |

TABLE 6.2 – Suite de la page précédente

| name               | type    | description  |
|--------------------|---------|--|
| cc                 | object  | <p>The users on the cc list. To modify this field, pass an object, which may have the following items :</p> <ul style="list-style-type: none"> <li>— <code>add</code> (array) User names to add to the CC list. They must be full user names, and an error will be thrown if you pass in an invalid user name.</li> <li>— <code>remove</code> (array) User names to remove from the CC list. They must be full user names, and an error will be thrown if you pass in an invalid user name.</li> </ul>   |
| is_cc_accessible   | boolean | Whether or not users in the CC list are allowed to access the bug, even if they aren't in a group that can normally access the bug.  |
| comment            | object  | <p>A comment on the change. The object may contain the following items :</p> <ul style="list-style-type: none"> <li>— <code>body</code> (string) The actual text of the comment. For compatibility with the parameters to <i>Create Comments</i>, you can also call this field <code>comment</code>, if you want.</li> <li>— <code>is_private</code> (boolean) Whether the comment is private or not. If you try to make a comment private and you don't have the permission to, an error will be thrown.</li> </ul>   |
| comment_is_private | object  | <p>This is how you update the privacy of comments that are already on a bug. This is a object, where the keys are the <code>int</code> ID of comments (not their count on a bug, like #1, #2, #3, but their globally-unique ID, as returned by <i>Get Comments</i> and the value is a <code>boolean</code> which specifies whether that comment should become private (<code>true</code>) or public (<code>false</code>).</p> <p>The comment IDs must be valid for the bug being updated. Thus, it is not practical to use this while updating multiple bugs at once, as a single comment ID will never be valid on multiple bugs.</p> |

Suite sur la page suivante



TABLE 6.2 – Suite de la page précédente

| name                       | type   | description  |
|----------------------------|--------|--|
| component                  | string | The Component the bug is in.   |
| deadline                   | date   | The Deadline field is a date specifying when the bug must be completed by, in the format YYYY-MM-DD.   |
| dupe_of                    | int    | The bug that this bug is a duplicate of. If you want to mark a bug as a duplicate, the safest thing to do is to set this value and <i>not</i> set the <code>status</code> or <code>resolution</code> fields. They will automatically be set by Bugzilla to the appropriate values for duplicate bugs.  |
| estimated_time             | double | The total estimate of time required to fix the bug, in hours. This is the <i>total</i> estimate, not the amount of time remaining to fix it.   |
| flags                      | array  | An array of Flag change objects. The items needed are described below.   |
| groups                     | object | The groups a bug is in. To modify this field, pass an object, which may have the following items : <ul style="list-style-type: none"> <li>— <code>add</code> (array) The names of groups to add. Passing in an invalid group name or a group that you cannot add to this bug will cause an error to be thrown.</li> <li>— <code>remove</code> (array) The names of groups to remove. Passing in an invalid group name or a group that you cannot remove from this bug will cause an error to be thrown.</li> </ul> |
| Suite sur la page suivante |        |  |

TABLE 6.2 – Suite de la page précédente

| name                       | type   | description   |
|----------------------------|--------|---|
| keywords                   | object | <p>Keywords on the bug. To modify this field, pass an object, which may have the following items :</p> <ul style="list-style-type: none"> <li>— <code>add</code> (array) The names of keywords to add to the field on the bug. Passing something that isn't a valid keyword name will cause an error to be thrown.</li> <li>— <code>remove</code> (array) The names of keywords to remove from the field on the bug. Passing something that isn't a valid keyword name will cause an error to be thrown.</li> <li>— <code>set</code> (array) An exact set of keywords to set the field to, on the bug. Passing something that isn't a valid keyword name will cause an error to be thrown. Specifying <code>set</code> overrides <code>add</code> and <code>remove</code>.</li> </ul> |
| op_sys                     | string | The Operating System (“OS”) field on the bug.   |
| platform                   | string | The Platform or “Hardware” field on the bug.  |
| priority                   | string | The Priority field on the bug.  |
| Suite sur la page suivante |        |   |

TABLE 6.2 – Suite de la page précédente

| name                  | type    | description   |
|-----------------------|---------|---|
| product               | string  | <p>The name of the product that the bug is in. If you change this, you will probably also want to change <code>target_milestone</code>, <code>version</code>, and <code>component</code>, since those have different legal values in every product.</p> <p>If you cannot change the <code>target_milestone</code> field, it will be reset to the default for the product, when you move a bug to a new product.</p> <p>You may also wish to add or remove groups, as which groups are valid on a bug depends on the product. Groups that are not valid in the new product will be automatically removed, and groups which are mandatory in the new product will be automatically added, but no other automatic group changes will be done.</p> <hr/> <p><b>Note :</b> Users can only move a bug into a product if they would normally have permission to file new bugs in that product.</p> <hr/> |
| qa_contact            | string  | The full login name of the bug's QA Contact.  |
| is_creator_accessible | boolean | Whether or not the bug's reporter is allowed to access the bug, even if they aren't in a group that can normally access the bug.  |
| remaining_time        | double  | How much work time is remaining to fix the bug, in hours. If you set <code>work_time</code> but don't explicitly set <code>remaining_time</code> , then the <code>work_time</code> will be deducted from the bug's <code>remaining_time</code> .  |
| reset_assigned_to     | boolean | If true, the <code>assigned_to</code> field will be reset to the default for the component that the bug is in. (If you have set the component at the same time as using this, then the component used will be the new component, not the old one.)  |

Suite sur la page suivante

TABLE 6.2 – Suite de la page précédente

| name             | type    | description   |
|------------------|---------|---|
| reset_qa_contact | boolean | If true, the <code>qa_contact</code> field will be reset to the default for the component that the bug is in. (If you have set the component at the same time as using this, then the component used will be the new component, not the old one.)   |
| resolution       | string  | The current resolution. May only be set if you are closing a bug or if you are modifying an already-closed bug. Attempting to set the resolution to <i>any</i> value (even an empty or null string) on an open bug will cause an error to be thrown.<br><br><b>Note :</b> If you change the <code>status</code> field to an open status, the resolution field will automatically be cleared, so you don't have to clear it manually.            |
| see_also         | object  | The See Also field on a bug, specifying URLs to bugs in other bug trackers. To modify this field, pass an object, which may have the following items :<br><ul style="list-style-type: none"> <li>— <code>add</code> (array) URLs to add to the field. Each URL must be a valid URL to a bug-tracker, or an error will be thrown.</li> <li>— <code>remove</code> (array) URLs to remove from the field. Invalid URLs will be ignored.</li> </ul> |
| severity         | string  | The Severity field of a bug.  |
| status           | string  | The status you want to change the bug to. Note that if a bug is changing from open to closed, you should also specify a <code>resolution</code> .   |
| summary          | string  | The Summary field of the bug.   |
| target_milestone | string  | The bug's Target Milestone.   |
| url              | string  | The "URL" field of a bug.   |
| version          | string  | The bug's Version field.  |
| whiteboard       | string  | The Status Whiteboard field of a bug.   |
| work_time        | double  | The number of hours worked on this bug as part of this change. If you set <code>work_time</code> but don't explicitly set <code>remaining_time</code> , then the <code>work_time</code> will be deducted from the bug's <code>remaining_time</code> .   |

You can also set the value of any custom field by passing its name as a parameter, and the value to set the field to. For multiple-selection fields, the value should be an array of strings.

Flag change object :

The following values can be specified. At least the `status` and one of `type_id`, `id`, or `name` must be specified. If a `type_id` or `name` matches a single currently set flag, the flag will be updated unless `new` is specified.

| name      | type    | description  |
|-----------|---------|--|
| name      | string  | The name of the flag that will be created or updated.  |
| type_id   | int     | The internal flag type ID that will be created or updated. You will need to specify the <code>type_id</code> if more than one flag type of the same name exists. |
| status    | string  | The flags new status (i.e. "?" , "+" , "-" or "X" to clear a flag).  |
| requestee | string  | The login of the requestee if the flag type is requestable to a specific user.   |
| id        | int     | Use ID to specify the flag to be updated. You will need to specify the <code>id</code> if more than one flag is set of the same name.                            |
| new       | boolean | Set to true if you specifically want a new flag to be created.   |

### Response

```
{
  "bugs" : [
    {
      "alias" : [],
      "changes" : {
        "keywords" : {
          "added" : "funny, stupid",
          "removed" : ""
        },
        "status" : {
          "added" : "IN_PROGRESS",
          "removed" : "CONFIRMED"
        }
      },
      "id" : 35,
      "last_change_time" : "2014-09-29T14:25:35Z"
    }
  ]
}
```

bugs (array) This points to an array of objects with the following items :

| name             | type     | description   |
|------------------|----------|---|
| id               | int      | The ID of the bug that was updated.   |
| alias            | array    | The aliases of the bug that was updated, if this bug has any alias.   |
| last_change_time | datetime | The exact time that this update was done at, for this bug. If no update was done (that is, no fields had their values changed and no comment was added) then this will instead be the last time the bug was updated.  |
| changes          | object   | The changes that were actually done on this bug. The keys are the names of the fields that were changed, and the values are an object with two keys : <ul style="list-style-type: none"><li>— <code>added</code> (string) The values that were added to this field, possibly a comma-and-space-separated list if multiple values were added.</li><li>— <code>removed</code> (string) The values that were removed from this field, possibly a comma-and-space-separated list if multiple values were removed.</li></ul> |

Currently, some fields are not tracked in `changes` : `comment`, `comment_is_private`, and `work_time`. This means that they will not show up in the return value even if they were successfully updated. This may change in a future version of Bugzilla.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 6.1.4 Bug User Last Visited

#### Update Last Visited

Update the last-visited time for the specified bug and current user.

#### Request

To update the time for a single bug id :

```
POST /rest/bug_user_last_visit/(id)
```

To update one or more bug ids at once :

```
POST /rest/bug_user_last_visit
```

```
{
  "ids" : [35,36,37]
}
```

| name       | type  | description                    |
|------------|-------|--------------------------------|
| <b>id</b>  | int   | An integer bug id.             |
| <b>ids</b> | array | One or more bug ids to update. |

**Response**

```
[
  {
    "id" : 100,
    "last_visit_ts" : "2014-10-16T17:38:24Z"
  }
]
```

An array of objects containing the items :

| name          | type     | description                                  |
|---------------|----------|--|
| id            | int      | The bug id.                                  |
| last_visit_ts | datetime | The timestamp the user last visited the bug. |

**Get Last Visited****Request**

Get the last-visited timestamp for one or more specified bug ids or get a list of the last 20 visited bugs and their timestamps.

To return the last-visited timestamp for a single bug id :

```
GET /rest/bug_user_last_visit/(id)
```

To return more than one specific bug timestamps :

```
GET /rest/bug_user_last_visit/123?ids=234&ids=456
```

To return just the most recent 20 timestamps :

```
GET /rest/bug_user_last_visit
```

| name       | type  | description                          |
|------------|-------|--------------------------------------|
| <b>id</b>  | int   | An integer bug id.                   |
| <b>ids</b> | array | One or more optional bug ids to get. |

**Response**

```
[
  {
    "id" : 100,
    "last_visit_ts" : "2014-10-16T17:38:24Z"
  }
]
```

An array of objects containing the following items :

| name          | type     | description                                  |
|---------------|----------|--|
| id            | int      | The bug id.                                  |
| last_visit_ts | datetime | The timestamp the user last visited the bug. |

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 6.1.5 Bugzilla Information

These methods are used to get general configuration information about this Bugzilla instance.

### Version

Returns the current version of Bugzilla. Normally in the format of X.X or X.X.X. For example, 4.4 for the initial release of a new branch. Or 4.4.6 for a minor release on the same branch.

### Request

GET /rest/version

### Response

```
{
  "version": "4.5.5+"
}
```

| name    | type   | description                          |
|---------|--------|--------------------------------------|
| version | string | The current version of this Bugzilla |

### Extensions

Gets information about the extensions that are currently installed and enabled in this Bugzilla.

### Request

GET /rest/extensions

### Response

```
{
  "extensions": {
    "Voting": {
      "version": "4.5.5+"
    },
    "BmpConvert": {
      "version": "1.0"
    }
  }
}
```

| name       | type   | description   |
|------------|--------|---|
| extensions | object | An object containing the extensions enabled as keys. Each extension object contains the following keys : <ul style="list-style-type: none"><li>— version (string) The version of the extension.</li></ul> |



## Timezone

Returns the timezone in which Bugzilla expects to receive dates and times on the API. Currently hard-coded to UTC (“+0000”). This is unlikely to change.

### Request

GET /rest/timezone

```
{
  "timezone": "+0000"
}
```

### Response

| name     | type   | description   |
|----------|--------|---|
| timezone | string | The timezone offset as a string in (+/-)XXXX (RFC 2822) format. |

## Time

Gets information about what time the Bugzilla server thinks it is, and what timezone it's running in.

### Request

GET /rest/time

### Response

```
{
  "web_time_utc": "2014-09-26T18:01:30Z",
  "db_time": "2014-09-26T18:01:30Z",
  "web_time": "2014-09-26T18:01:30Z",
  "tz_offset": "+0000",
  "tz_short_name": "UTC",
  "tz_name": "UTC"
}
```

| name          | type   | description  |
|---------------|--------|--|
| db_time       | string | The current time in UTC, according to the Bugzilla database server.<br>Note that Bugzilla assumes that the database and the webserver are running in the same time zone. However, if the web server and the database server aren't synchronized or some reason, <i>this</i> is the time that you should rely on or doing searches and other input to the WebService. |
| web_time      | string | This is the current time in UTC, according to Bugzilla's web server.<br>This might be different by a second from db_time since this comes from a different source. If it's any more different than a second, then there is likely some problem with this Bugzilla instance. In this case you should rely on the db_time, not the web_time.                           |
| web_time      | string | Identical to web_time. (Exists only for backwards-compatibility with versions of Bugzilla before 3.6.)   |
| tz_name       | string | The literal string UTC. (Exists only for backwards-compatibility with versions of Bugzilla before 3.6.)  |
| tz_short_name | string | The literal string UTC. (Exists only for backwards-compatibility with versions of Bugzilla before 3.6.)  |
| tz_offset     | string | The literal string +0000. (Exists only for backwards-compatibility with versions of Bugzilla before 3.6.)  |

### Parameters

Returns parameter values currently used in this Bugzilla.

#### Request

GET /rest/parameters

#### Response

Example response for anonymous user :

```
{
  "parameters" : {
    "maintainer" : "admin@example.com",
    "requirelogin" : "0"
  }
}
```

Example response for authenticated user :

```
{
  "parameters" : {
    "allowemailchange" : "1",
    "attachment_base" : "http://bugzilla.example.com/",
    "commentonchange_resolution" : "0",
    "commentonduplicate" : "0",
    "cookiepath" : "/",
    "createemailregexp" : ".*",
    "defaulttopsys" : "",
    "defaultplatform" : "",
    "defaultpriority" : "--",
    "defaultseverity" : "normal",
    "duplicate_or_move_bug_status" : "RESOLVED",
    "emailregexp" : "^[\\w\\.\\+\\-=']+@[\\w\\.\\+\\-]+\\.([\\w\\-]+)$",
    "emailsuffix" : "",
    "letsubmitterchoosemilestone" : "1",
    "letsubmitterchoosepriority" : "1",
    "mailfrom" : "bugzilla-daemon@example.com",
    "maintainer" : "admin@example.com",
    "maxattachmentsize" : "1000",
    "maxlocalattachment" : "0",
    "musthavemilestoneonaccept" : "0",
    "noresolveonopenblockers" : "0",
    "password_complexity" : "no_constraints",
    "rememberlogin" : "on",
    "requirelogin" : "0",
    "urlbase" : "http://bugzilla.example.com/",
    "use_see_also" : "1",
    "useclassification" : "1",
    "usemenuforusers" : "0",
    "useqacontact" : "1",
    "usestatuswhiteboard" : "1",
    "usetargetmilestone" : "1",
  }
}
```

A logged-out user can only access the maintainer and requirelogin parameters.

A logged-in user can access the following parameters (listed alphabetically) :

- allowemailchange
- attachment\_base
- commentonchange\_resolution
- commentonduplicate
- cookiepath
- defaulttopsys
- defaultplatform
- defaultpriority
- defaultseverity
- duplicate\_or\_move\_bug\_status
- emailregexpdesc
- emailsuffix
- letsubmitterchoosemilestone
- letsubmitterchoosepriority
- mailfrom
- maintainer
- maxattachmentsize
- maxlocalattachment
- musthavemilestoneonaccept
- noresolveonopenblockers
- password\_complexity
- rememberlogin
- requirelogin
- search\_allow\_no\_criteria
- urlbase
- use\_see\_also
- useclassification
- usemenuforusers
- useqacontact
- usestatuswhiteboard
- usetargetmilestone

A user in the tweakparams group can access all existing parameters. New parameters can appear or obsolete parameters can disappear depending on the version of Bugzilla and on extensions being installed. The list of parameters returned by this method is not stable and will never be stable.

## Last Audit Time

Gets the most recent timestamp among all of the events recorded in the audit\_log table.

### Request

To get most recent audit timestamp for all classes :

```
GET /rest/last_audit_time
```

To get the the most recent audit timestamp for the Bugzilla::Product class :

```
GET /rest/last_audit_time?class=Bugzilla::Product
```

| name  | type  | description  |
|-------|-------|--|
| class | array | The class names are defined as Bugzilla::<class_name>" such as Bugzilla:Product" for products. |

### Response

```
{
  "last_audit_time": "2014-09-23T18:03:38Z"
}
```

| name            | type   | description                                    |
|-----------------|--------|--|
| last_audit_time | string | The maximum of the at_time from the audit_log. |

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 6.1.6 Classifications

This part of the Bugzilla API allows you to deal with the available classifications. You will be able to get information about them as well as manipulate them.

#### Get Classification

Returns an object containing information about a set of classifications.

#### Request

To return information on a single classification using the ID or name :

```
GET /rest/classification/(id_or_name)
```

| name       | type  | description                           |
|------------|-------|---------------------------------------|
| id_or_name | mixed | An Integer classification ID or name. |

#### Response

```
{
  "classifications": [
    {
      "sort_key": 0,
      "description": "Unassigned to any classifications",
      "products": [
        {
          "id": 2,
          "name": "FoodReplicator",
          "description": "Software that controls a piece of hardware that will create any food item t
        },
        {
          "description": "Silk, etc.",
          "name": "Spider Secretions",
          "id": 4
        }
      ]
    },
    {
      "id": 1,
      "name": "Unclassified"
    }
  ]
}
```

classifications (array) Each object is a classification that the user is authorized to see and has the following items :

| name        | type   | description   |
|-------------|--------|---|
| id          | int    | The ID of the classification.   |
| name        | string | The name of the classification.   |
| description | string | The description of the classification.  |
| sort_key    | int    | The value which determines the order the classification is sorted.                                      |
| products    | array  | Products the user is authorized to access within the classification. Each hash has the following keys : |

Product object :

| name        | type   | description                     |
|-------------|--------|---------------------------------|
| name        | string | The name of the product.        |
| id          | int    | The ID of the product.          |
| description | string | The description of the product. |

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 6.1.7 Comments

### Get Comments

This allows you to get data about comments, given a bug ID or comment ID.

#### Request

To get all comments for a particular bug using the bug ID or alias :

```
GET /rest/bug/(id_or_alias)/comment
```

To get a specific comment based on the comment ID :

```
GET /rest/bug/comment/(comment_id)
```

| name          | type      | description   |
|---------------|-----------|---|
| id_or_alias   | mixed     | A single integer bug ID or alias.   |
| comment_id    | int       | A single integer comment ID.  |
| new_sincetime | date-time | If specified, the method will only return comments <i>newer</i> than this time. This only affects comments returned from the <code>ids</code> argument. You will always be returned all comments you request in the <code>comment_ids</code> argument, even if they are older than this date. |

#### Response

```
{
  "bugs": {
    "35": {
      "comments": [
        {
          "time": "2000-07-25T13:50:04Z",
          "text": "test bug to fix problem in removing from cc list.",
          "bug_id": 35,
          "count": 0,
          "attachment_id": null,
          "is_private": false,
          "tags": [],

```

```

        "creator": "user@bugzilla.org",
        "creation_time": "2000-07-25T13:50:04Z",
        "id": 75
    }
  ]
},
"comments": {}
}

```

Two items are returned :

**bugs** This is used for bugs specified in `ids`. This is an object, where the keys are the numeric IDs of the bugs, and the value is a object with a single key, `comments`, which is an array of comments. (The format of comments is described below.)

Any individual bug will only be returned once, so if you specify an ID multiple times in `ids`, it will still only be returned once.

**comments** Each individual comment requested in `comment_ids` is returned here, in a object where the numeric comment ID is the key, and the value is the comment. (The format of comments is described below.)

A “comment” as described above is a object that contains the following items :

| name                        | type      | description   |
|-----------------------------|-----------|---|
| <code>id</code>             | int       | The globally unique ID for the comment.   |
| <code>bug_id</code>         | int       | The ID of the bug that this comment is on.  |
| <code>attach-ment_id</code> | int       | If the comment was made on an attachment, this will be the ID of that attachment. Otherwise it will be null.  |
| <code>count</code>          | int       | The number of the comment local to the bug. The Description is 0, comments start with 1.  |
| <code>text</code>           | string    | The actual text of the comment.   |
| <code>creator</code>        | string    | The login name of the comment’s author.   |
| <code>time</code>           | date-time | The time (in Bugzilla’s timezone) that the comment was added.   |
| <code>creation_time</code>  | date-time | This is exactly same as the <code>time</code> key. Use this field instead of <code>time</code> for consistency with other methods including <i>Get Bug</i> and <i>Get Attachment</i> . For compatibility, <code>time</code> is still usable. However, please note that <code>time</code> may be deprecated and removed in a future release. |
| <code>is_private</code>     | boolean   | <code>true</code> if this comment is private (only visible to a certain group called the “insidergroup”), <code>false</code> otherwise.   |

## Create Comments

This allows you to add a comment to a bug in Bugzilla.

### Request

To create a comment on a current bug.

POST `/rest/bug/(id)/comment`

```

{
  "ids" : [123,..],
  "comment" : "This is an additional comment",
  "is_private" : false
}

```

`ids` is optional in the data example above and can be used to specify adding a comment to more than one bug at the same time.

| name                    | type    | description   |
|-------------------------|---------|---|
| <b>id</b>               | int     | The ID or alias of the bug to append a comment to.  |
| <code>ids</code>        | array   | List of integer bug IDs to add the comment to.  |
| <b>comment</b>          | string  | The comment to append to the bug. If this is empty or all whitespace, an error will be thrown saying that you did not set the <code>comment</code> parameter. |
| <code>is_private</code> | boolean | If set to true, the comment is private, otherwise it is assumed to be public.   |
| <code>work_time</code>  | double  | Adds this many hours to the “Hours Worked” on the bug. If you are not in the time tracking group, this value will be ignored.                                 |

### Response

```
{
  "id" : 789
}
```

| name            | type | description                      |
|-----------------|------|----------------------------------|
| <code>id</code> | int  | ID of the newly-created comment. |

### Search Comment Tags

Searches for tags which contain the provided substring.

#### Request

To search for comment tags :

```
GET /rest/bug/comment/tags/(query)
```

Example :

```
GET /rest/bug/comment/tags/spa
```

| name               | type   | description   |
|--------------------|--------|---|
| <b>query</b>       | string | Only tags containing this substring will be returned.                         |
| <code>limit</code> | int    | If provided will return no more than <code>limit</code> tags. Defaults to 10. |

#### Response

```
[
  "spam"
]
```

An array of matching tags.

### Update Comment Tags

Adds or removes tags from a comment.

#### Request

To update the tags comments attached to a comment :

```
PUT /rest/bug/comment/(comment_id)/tags
```

Example :

```
{
  "comment_id" : 75,
  "add" : ["spam", "bad"]
}
```

| name              | type  | description                          |
|-------------------|-------|--------------------------------------|
| <b>comment_id</b> | int   | The ID of the comment to update.     |
| add               | array | The tags to attach to the comment.   |
| remove            | array | The tags to detach from the comment. |

### Response

```
[
  "bad",
  "spam"
]
```

An array of strings containing the comment's updated tags.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 6.1.8 Components

This part of the Bugzilla API allows you to deal with the available product components. You will be able to get information about them as well as manipulate them.

### Create Component

This allows you to create a new component in Bugzilla. You must be authenticated and be in the *editcomponents* group to perform this action.

#### Request

To create a new component :

POST /rest/component

```
{
  "product" : "TestProduct",
  "name" : "New Component",
  "description" : "This is a new component",
  "default_assignee" : "dkl@mozilla.com"
}
```

Some params must be set, or an error will be thrown. These params are shown in **bold**.



| name                    | type    | description  |
|-------------------------|---------|--|
| <b>name</b>             | string  | The name of the new component.   |
| <b>product</b>          | string  | The name of the product that the component must be added to. This product must already exist, and the user have the necessary permissions to edit components for it. |
| <b>description</b>      | string  | The description of the new component.  |
| <b>default_assignee</b> | string  | The login name of the default assignee of the component.   |
| default_cc              | array   | Each string representing one login name of the default CC list.  |
| default_qa_contact      | string  | The login name of the default QA contact for the component.  |
| is_open                 | boolean | 1 if you want to enable the component for bug creations. 0 otherwise. Default is 1.  |

**Response**

```
{
  "id": 27
}
```

| name | type | description                          |
|------|------|--------------------------------------|
| id   | int  | The ID of the newly-added component. |

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 6.1.9 Bug Fields

The Bugzilla API for getting details about bug fields.

**Fields**

Get information about valid bug fields, including the lists of legal values for each field.

**Request**

To get information about all fields :

```
GET /rest/field/bug
```

To get information related to a single field :

```
GET /rest/field/bug/(id_or_name)
```

| name       | type  | description  |
|------------|-------|--|
| id_or_name | mixed | An integer field ID or string representing the field name. |

**Response**

```
{
  "fields": [
    {
      "display_name": "Priority",
      "name": "priority",
      "type": 2,

```

```
"is_mandatory": false,
"value_field": null,
"values": [
  {
    "sortkey": 100,
    "sort_key": 100,
    "visibility_values": [],
    "name": "P1"
  },
  {
    "sort_key": 200,
    "name": "P2",
    "visibility_values": [],
    "sortkey": 200
  },
  {
    "sort_key": 300,
    "visibility_values": [],
    "name": "P3",
    "sortkey": 300
  },
  {
    "sort_key": 400,
    "name": "P4",
    "visibility_values": [],
    "sortkey": 400
  },
  {
    "name": "P5",
    "visibility_values": [],
    "sort_key": 500,
    "sortkey": 500
  }
],
"visibility_values": [],
"visibility_field": null,
"is_on_bug_entry": false,
"is_custom": false,
"id": 13
}
]
```

field (array) Field objects each containing the following items :

| name                    | type    | description   |
|-------------------------|---------|---|
| id                      | int     | An integer ID uniquely identifying this field in this installation only.  |
| type                    | int     | The number of the fieldtype. The following values are defined : <ul style="list-style-type: none"> <li>— 0 Field type unknown</li> <li>— 1 Single-line string field</li> <li>— 2 Single value field</li> <li>— 3 Multiple value field</li> <li>— 4 Multi-line text value</li> <li>— 5 Date field with time</li> <li>— 6 Bug ID field</li> <li>— 7 See Also field</li> <li>— 8 Keywords field</li> <li>— 9 Date field</li> <li>— 10 Integer field</li> </ul> |
| is_custom               | boolean | true when this is a custom field, false otherwise.  |
| name                    | string  | The internal name of this field. This is a unique identifier for this field. If this is not a custom field, then this name will be the same across all Bugzilla installations.  |
| display_name            | string  | The name of the field, as it is shown in the user interface.  |
| is_mandatory            | boolean | true if the field must have a value when filing new bugs. Also, mandatory fields cannot have their value cleared when updating bugs.  |
| is_on_bug_entry         | boolean | For custom fields, this is true if the field is shown when you enter a new bug. For standard fields, this is currently always false, even if the field shows up when entering a bug. (To know whether or not a standard field is valid on bug entry, see <i>Create Bug</i> .)   |
| visibility_field        | string  | The name of a field that controls the visibility of this field in the user interface. This field only appears in the user interface when the named field is equal to one of the values in visibility_values. Can be null.   |
| visibility_values       | array   | This field is only shown when visibility_field matches one of these string values. When visibility_field is null, then this is an empty array.  |
| value_field             | string  | The name of the field that controls whether or not particular values of the field are shown in the user interface. Can be null.   |
| values                  | array   | Objects representing the legal values for select-type (drop-down and multiple-selection) fields. This is also populated for the component, version, target_milestone, and keywords fields, but not for  |
| <b>6.1. Core API v1</b> |         | 133   |

Value object :

| name              | type    | description   |
|-------------------|---------|---|
| name              | string  | The actual value—this is what you would specify for this field in create, etc.  |
| sort_key          | int     | Values, when displayed in a list, are sorted first by this integer and then secondly by their name.   |
| visibility_values | array   | If <code>value_field</code> is defined for this field, then this value is only shown if the <code>value_field</code> is set to one of the values listed in this array. Note that for per-product fields, <code>value_field</code> is set to <code>product</code> and <code>visibility_values</code> will reflect which product(s) this value appears in.  |
| is_active         | boolean | This value is defined only for certain product-specific fields such as <code>version</code> , <code>target_milestone</code> or <code>component</code> . When true, the value is active ; otherwise the value is not active.   |
| description       | string  | The description of the value. This item is only included for the <code>keywords</code> field.   |
| is_open           | boolean | For <code>bug_status</code> values, determines whether this status specifies that the bug is “open” ( <code>true</code> ) or “closed” ( <code>false</code> ). This item is only included for the <code>bug_status</code> field.   |
| can_change_to     | array   | For <code>bug_status</code> values, this is an array of objects that determine which statuses you can transition to from this status. (This item is only included for the <code>bug_status</code> field.) Each object contains the following items : <ul style="list-style-type: none"><li>— <code>name</code> : (string) The name of the new status</li><li>— <code>comment_required</code> : (boolean) <code>true</code> if a comment is required when you change a bug into this status using this transition.</li></ul> |

## Legal Values

**DEPRECATED** Use “Fields” instead.

Tells you what values are allowed for a particular field.

## Request

To get information on the values for a field based on field name :

```
GET /rest/field/bug/(field)/values
```

To get information based on field name and a specific product :

```
GET /rest/field/bug/(field)/(product_id)/values
```

| name       | type   | description   |
|------------|--------|---|
| field      | string | The name of the field you want information about. This should be the same as the name you would use in <i>Create Bug</i> , below. |
| product_id | int    | If you're picking a product-specific field, you have to specify the ID of the product you want the values for.                    |

### Response

```
{
  "values": [
    "P1",
    "P2",
    "P3",
    "P4",
    "P5"
  ]
}
```

| name   | type  | description   |
|--------|-------|---|
| values | array | The legal values for this field. The values will be sorted as they normally would be in Bugzilla. |

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 6.1.10 Flag Types

This part of the Bugzilla API allows you to get and create bug and attachment flags.

### Get Flag Type

Get information about valid flag types that can be set for bugs and attachments.

#### Request

To get information about all flag types for a product :

```
GET /rest/flag_type/(product)
```

To get information about flag\_types for a product and component :

```
GET /rest/flag_type/(product)/(component)
```

```
{
  "bug": [
    {
      "is_multiplicable": false,
      "is_requesteeble": false,
      "values": [
        "X",

```

```
    "?",
    "+",
    "-"
  ],
  "id": 11,
  "type": "bug",
  "is_active": true,
  "description": "Blocks the next release",
  "name": "blocker"
},
{
  "is_requesteeble": false,
  "is_multiplicable": false,
  "is_active": true,
  "description": "Regression found?",
  "name": "regression",
  "id": 10,
  "type": "bug",
  "values": [
    "X",
    "?",
    "+",
    "-"
  ]
},
],
"attachment": [
  {
    "is_requesteeble": true,
    "is_multiplicable": true,
    "name": "review",
    "is_active": true,
    "description": "Review the patch for correctness and applicability to the problem.",
    "type": "attachment",
    "id": 1,
    "values": [
      "X",
      "?",
      "+",
      "-"
    ]
  },
  {
    "name": "approval",
    "description": "Approve the patch for check-in to the tree.",
    "is_active": true,
    "values": [
      "X",
      "?",
      "+",
      "-"
    ]
  },
  "type": "attachment",
  "id": 3,
  "is_multiplicable": false,
  "is_requesteeble": false
}
]
```

```
}

```

You must pass a product name and an optional component name. If the product or component names contains a / character, you will need to url encode it.

| name           | type   | description   |
|----------------|--------|---|
| <b>product</b> | string | The name of a valid product.                                  |
| component      | string | An optional valid component name associated with the product. |

### Response

An object containing two items, `bug` and `attachment`. Each value is an array of objects, containing the following items :

| name             | type    | description   |
|------------------|---------|---|
| id               | int     | An integer ID uniquely identifying this flag type.  |
| name             | string  | The name for the flag type.   |
| type             | string  | The target of the flag type which is either <code>bug</code> or <code>attachment</code> . |
| description      | string  | The description of the flag type.   |
| values           | array   | String values that the user can set on the flag type.                                     |
| is_requesteeble  | boolean | Users can ask specific other users to set flags of this type.                             |
| is_multiplicable | boolean | Multiple flags of this type can be set for the same bug or attachment.                    |

### Create Flag Type

Create a new flag type. You must be authenticated and be in the `editcomponents` group to perform this action.

### Request

```
POST /rest/flag_type
```

```
{
  "name" : "feedback",
  "description" : "This attachment needs feedback",
  "inclusions" : [ "WorldControl "],
  "target_type" : "attachment"
}
```

Some params must be set, or an error will be thrown. The required params are marked in **bold**.

| name                               | type    | description  |
|------------------------------------|---------|--|
| <b>name</b>                        | string  | The name of the new flag type.   |
| <b>description</b>                 | string  | A description for the flag type.   |
| <b>target_type</b>                 | string  | The new flag is either for a <code>bug</code> or an <code>attachment</code> .  |
| <b>inclusions</b>                  | array   | An array of strings or an object containing product names, and optionally component names. If you provide a string, the flag type will be shown on all bugs in that product. If you provide an object, the key represents the product name, and the value is the components of the product to be included. |
| <b>exclusions</b>                  | array   | An array of strings or an object containing product names. This uses the same format as <code>inclusions</code> . This will exclude the flag from all products and components specified.   |
| <b>sortkey</b>                     | int     | A number between 1 and 32767 by which this type will be sorted when displayed to users in a list ; ignore if you don't care what order the types appear in or if you want them to appear in alphabetical order.  |
| <b>is_active</b>                   | boolean | Flag of this type appear in the UI and can be set. Default is <code>true</code> .  |
| <b>is_requestable</b>              | boolean | Users can ask for flags of this type to be set. Default is <code>true</code> .   |
| <b>cc_list</b>                     | array   | If the flag type is requestable, who should receive e-mail notification of requests. This is an array of e-mail addresses which do not need to be Bugzilla logins.   |
| <b>is_specifically_requestable</b> | boolean | Users can ask specific other users to set flags of this type as opposed to just asking the wind. Default is <code>true</code> .  |
| <b>is_multiplicable</b>            | boolean | Multiple flags of this type can be set on the same bug. Default is <code>true</code> .   |
| <b>grant_group</b>                 | string  | The group allowed to grant/deny flags of this type (to allow all users to grant/deny these flags, select no group). Default is no group.   |
| <b>request_group</b>               | string  | If flags of this type are requestable, the group allowed to request them (to allow all users to request these flags, select no group). Note that the request group alone has no effect if the grant group is not defined ! Default is no group.  |

An example for inclusions and/or exclusions :

```
[
  "FooProduct"
]

{
  "BarProduct" : [ "C1", "C3" ],
  "BazProduct" : [ "C7" ]
}
```

This flag will be added to **all** components of `FooProduct`, components `C1` and `C3` of `BarProduct`, and `C7` of `BazProduct`.

### Response

```
{
  "id": 13
}
```

| name           | type | description                                |
|----------------|------|--|
| <b>flag_id</b> | int  | ID of the new FlagType object is returned. |

### Update Flag Type

This allows you to update a flag type in Bugzilla. You must be authenticated and be in the `editcomponents` group to perform this action.



## Request

PUT /rest/flag\_type/(id\_or\_name)

```
{
  "ids" : [13],
  "name" : "feedback-new",
  "is_requestable" : false
}
```

You can edit a single flag type by passing the ID or name of the flag type in the URL. To edit more than one flag type, you can specify addition IDs or flag type names using the `ids` or `names` parameters respectively.

One of the below must be specified.

| name              | type  | description  |
|-------------------|-------|--|
| <b>id_or_name</b> | mixed | Integer flag type ID or name.  |
| <b>ids</b>        | array | Numeric IDs of the flag types that you wish to update.   |
| <b>names</b>      | array | Names of the flag types that you wish to update. If many flag types have the same name, this will change <b>all</b> of them. |

The following parameters specify the new values you want to set for the flag types you are updating.

| name                        | type    | description  |
|-----------------------------|---------|--|
| name                        | string  | A short name identifying this type.  |
| description                 | string  | A comprehensive description of this type.  |
| inclusions                  | array   | An array of strings or an object containing product names, and optionally component names. If you provide a string, the flag type will be shown on all bugs in that product. If you provide an object, the key represents the product name, and the value is the components of the product to be included. |
| exclusions                  | array   | An array of strings or an object containing product names. This uses the same format as <code>inclusions</code> . This will exclude the flag from all products and components specified.   |
| sortkey                     | int     | A number between 1 and 32767 by which this type will be sorted when displayed to users in a list ; ignore if you don't care what order the types appear in or if you want them to appear in alphabetical order.  |
| is_active                   | boolean | Flag of this type appear in the UI and can be set.   |
| is_requestable              | boolean | Users can ask for flags of this type to be set.  |
| cc_list                     | array   | If the flag type is requestable, who should receive e-mail notification of requests. This is an array of e-mail addresses which do not need to be Bugzilla logins.   |
| is_specifically_requestable | boolean | Users can ask specific other users to set flags of this type as opposed to just asking the wind.   |
| is_multiplicable            | boolean | Multiple flags of this type can be set on the same bug.  |
| grant_group                 | string  | The group allowed to grant/deny flags of this type (to allow all users to grant/deny these flags, select no group).  |
| request_group               | string  | If flags of this type are requestable, the group allowed to request them (to allow all users to request these flags, select no group). Note that the request group alone has no effect if the grant group is not defined !   |

An example for inclusions and/or exclusions :

```
[
  "FooProduct",
]
```

```
{
  "BarProduct" : [ "C1", "C3" ],
  "BazProduct" : [ "C7" ]
}
```

This flag will be added to **all** components of FooProduct, components C1 and C3 of BarProduct, and C7 of BazProduct.

### Response

```
{
  "flagtypes": [
    {
      "name": "feedback-new",
      "changes": {
        "is_requestable": {
          "added": "0",
          "removed": "1"
        },
        "name": {
          "removed": "feedback",
          "added": "feedback-new"
        }
      }
    },
    {
      "id": 13
    }
  ]
}
```

flagtypes (array) Flag change objects containing the following items :

| name    | type   | description  |
|---------|--------|--|
| id      | int    | The ID of the flag type that was updated.  |
| name    | string | The name of the flag type that was updated.  |
| changes | object | The changes that were actually done on this flag type. The keys are the names of the fields that were changed, and the values are an object with two items : <ul style="list-style-type: none"> <li>— added : (string) The value that this field was changed to.</li> <li>— removed : (string) The value that was previously set in this field.</li> </ul> |

Booleans changes will be represented with the strings '1' and '0'.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

### 6.1.11 Groups

The API for creating, changing, and getting information about groups.

## Create Group

This allows you to create a new group in Bugzilla. You must be authenticated and be in the *creategroups* group to perform this action.

### Request

POST /rest/group

```
{
  "name" : "secret-group",
  "description" : "Too secret for you!",
  "is_active" : true
}
```

Some params must be set, or an error will be thrown. The required params are marked in **bold**.

| name                           | type    | description  |
|--------------------------------|---------|--|
| <b>name</b>                    | string  | A short name for this group. Must be unique. This is not usually displayed in the user interface, except in a few places.                                      |
| <b>des-<br/>crip-<br/>tion</b> | string  | A human-readable name for this group. Should be relatively short. This is what will normally appear in the UI as the name of the group.                        |
| user_regex                     | string  | A regular expression. Any user whose Bugzilla username matches this regular expression will automatically be granted membership in this group.                 |
| is_active                      | boolean | <code>true</code> if new group can be used for bugs, <code>false</code> if this is a group that will only contain users and no bugs will be restricted to it.  |
| icon_url                       | string  | A URL pointing to a small icon used to identify the group. This icon will show up next to users' names in various parts of Bugzilla if they are in this group. |

### Response

```
{
  "id": 22
}
```

| name | type | description                    |
|------|------|--------------------------------|
| id   | int  | ID of the newly-created group. |

## Update Group

This allows you to update a group in Bugzilla. You must be authenticated and be in the *creategroups* group to perform this action.

### Request

To update a group using the group ID or name :

PUT /rest/group/(id\_or\_name)

```
{
  "name" : "secret-group",
  "description" : "Too secret for you! (updated description)",
  "is_active" : false
}
```

You can edit a single group by passing the ID or name of the group in the URL. To edit more than one group, you can specify addition IDs or group names using the `ids` or `names` parameters respectively.

One of the below must be specified.

| name              | type  | description                |
|-------------------|-------|----------------------------|
| <b>id_or_name</b> | mixed | Integer group or name.     |
| <b>ids</b>        | array | IDs of groups to update.   |
| <b>names</b>      | array | Names of groups to update. |

The following parameters specify the new values you want to set for the group(s) you are updating.

| name                     | type    | description  |
|--------------------------|---------|--|
| <code>name</code>        | string  | A new name for the groups. If you try to set this while updating more than one group, an error will occur, as group names must be unique.                          |
| <code>description</code> | string  | A new description for the groups. This is what will appear in the UI as the name of the groups.  |
| <code>user_regex</code>  | string  | A new regular expression for email. Will automatically grant membership to these groups to anyone with an email address that matches this perl regular expression. |
| <code>is_active</code>   | boolean | Set if groups are active and eligible to be used for bugs. <code>true</code> if bugs can be restricted to this group, <code>false</code> otherwise.                |
| <code>icon_url</code>    | string  | A URL pointing to an icon that will appear next to the name of users who are in this group.  |

### Response

```
{
  "groups": [
    {
      "changes": {
        "description": {
          "added": "Too secret for you! (updated description)",
          "removed": "Too secret for you!"
        },
        "is_active": {
          "removed": "1",
          "added": "0"
        }
      },
      "id": "22"
    }
  ]
}
```

`groups` (array) Group change objects, each containing the following items :

| name    | type   | description   |
|---------|--------|---|
| id      | int    | The ID of the group that was updated.   |
| changes | object | The changes that were actually done on this group. The keys are the names of the fields that were changed, and the values are an object with two items : <ul style="list-style-type: none"> <li>— added : (string) The values that were added to this field, possibly a comma-and-space-separated list if multiple values were added.</li> <li>— removed : (string) The values that were removed from this field, possibly a comma-and-space-separated list if multiple values were removed.</li> </ul> |

## Get Group

Returns information about Bugzilla groups.

### Request

To return information about a specific group ID or name :

```
GET /rest/group/(id_or_name)
```

You can also return information about more than one specific group by using the following in your query string :

```
GET /rest/group?ids=1&ids=2&ids=3
GET /group?names=ProductOne&names=Product2
```

If neither IDs nor names are passed, and you are in the creategroups or editusers group, then all groups will be retrieved. Otherwise, only groups that you have bless privileges for will be returned.

| name       | type    | description  |
|------------|---------|--|
| id_or_name | mixed   | Integer group ID or name.  |
| ids        | array   | Integer IDs of groups.   |
| names      | array   | Names of groups.   |
| membership | boolean | Set to 1 then a list of members of the passed groups names and IDs will be returned. |

### Response

```
{
  "groups": [
    {
      "membership": [
        {
          "real_name": "Bugzilla User",
          "can_login": true,
          "name": "user@bugzilla.org",
          "login_denied_text": "",
          "id": 85,
          "email_enabled": false,
          "email": "user@bugzilla.org"
        }
      ]
    }
  ]
}
```

```

    },
  ],
  "is_active": true,
  "description": "Test Group",
  "user_regexp": "",
  "is_bug_group": true,
  "name": "TestGroup",
  "id": 9
}
]
}

```

If the user is a member of the *creategroups* group they will receive information about all groups or groups matching the criteria that they passed. You have to be in the *creategroups* group unless you're requesting membership information.

If the user is not a member of the *creategroups* group, but they are in the "editusers" group or have *bless* privileges to the groups they require membership information for, the *is\_active*, *is\_bug\_group* and *user\_regexp* values are not supplied.

The return value will be an object containing group names as the keys ; each value will be an object that describes the group and has the following items :

| name         | type   | description  |
|--------------|--------|--|
| id           | int    | The unique integer ID that Bugzilla uses to identify this group. Even if the name of the group changes, this ID will stay the same.  |
| name         | string | The name of the group.   |
| description  | string | The description of the group.  |
| is_bug_group | bool   | Whether this group is to be used for bug reports or is only administrative specific.   |
| user_regexp  | string | A regular expression that allows users to be added to this group if their login matches.   |
| is_active    | int    | Whether this group is currently active or not.   |
| users        | array  | User objects that are members of this group ; only returned if the user sets the <i>membership</i> parameter to 1. Each user object has the items describe in the User object below. |

User object :

| name          | type    | description   |
|---------------|---------|---|
| id            | int     | The ID of the user.   |
| real_name     | string  | The actual name of the user.  |
| email         | string  | The email address of the user.  |
| name          | string  | The login name of the user. Note that in some situations this is different than their email.  |
| can_login     | boolean | A boolean value to indicate if the user can login into bugzilla.  |
| email_enabled | boolean | A boolean value to indicate if bug-related mail will be sent to the user or not.  |
| disabled_text | string  | A text field that holds the reason for disabling a user from logging into Bugzilla. If empty, then the user account is enabled ; otherwise it is disabled/closed. |

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 6.1.12 Products

This part of the Bugzilla API allows you to list the available products and get information about them.

## List Products

Returns a list of the IDs of the products the user can search on.

### Request

To get a list of product IDs a user can select such as for querying bugs :

```
GET /rest/product_selectable
```

To get a list of product IDs a user can enter a bug against :

```
GET /rest/product_enterable
```

To get a list of product IDs a user can search or enter bugs against.

```
GET /rest/product_accessible
```

### Response

```
{
  "ids": [
    "2",
    "3",
    "19",
    "1",
    "4"
  ]
}
```

| name | type  | description                  |
|------|-------|------------------------------|
| ids  | array | List of integer product IDs. |

## Get Product

Returns a list of information about the products passed to it.

### Request

To return information about a specific type of products such as accessible, selectable, or enterable :

```
GET /rest/product?type=accessible
```

To return information about a specific product by id or name :

```
GET /rest/product/(id_or_name)
```

You can also return information about more than one product by using the following parameters in your query string :

```
GET /rest/product?ids=1&ids=2&ids=3
GET /rest/product?names=ProductOne&names=Product2
```

| name       | type   | description   |
|------------|--------|---|
| id_or_name | mixed  | Integer product ID or product name.   |
| ids        | array  | Product IDs   |
| names      | array  | Product names   |
| type       | string | The group of products to return. Valid values are accessible (default), selectable, and enterable. type can be a single value or an array of values if more than one group is needed with duplicates removed. |

### Response

```
{
  "products": [
    {
      "id": 1,
      "default_milestone": "---",
      "components": [
        {
          "is_active": true,
          "default_assigned_to": "admin@bugzilla.org",
          "id": 1,
          "sort_key": 0,
          "name": "TestComponent",
          "flag_types": {
            "bug": [
              {
                "is_active": true,
                "grant_group": null,
                "cc_list": "",
                "is_requestable": true,
                "id": 3,
                "is_multiplicable": true,
                "name": "needinfo",
                "request_group": null,
                "is_requesteeble": true,
                "sort_key": 0,
                "description": "needinfo"
              }
            ]
          }
        }
      ],
      "attachment": [
        {
          "description": "Review",
          "is_multiplicable": true,
          "name": "review",
          "is_requesteeble": true,
          "request_group": null,
          "sort_key": 0,
          "cc_list": "",
          "grant_group": null,
          "is_requestable": true,
          "id": 2,
          "is_active": true
        }
      ]
    }
  ],
  "default_ga_contact": "",
  "description": "This is a test component."
}
```



```

    }
  ],
  "is_active": true,
  "classification": "Unclassified",
  "versions": [
    {
      "id": 1,
      "name": "unspecified",
      "is_active": true,
      "sort_key": 0
    }
  ],
  "description": "This is a test product.",
  "has_unconfirmed": true,
  "milestones": [
    {
      "name": "---",
      "is_active": true,
      "sort_key": 0,
      "id": 1
    }
  ],
  "name": "TestProduct"
}
]
}

```

products (array) Each product object has the following items :

| name              | type    | description  |
|-------------------|---------|--|
| id                | int     | An integer ID uniquely identifying the product in this installation only.                      |
| name              | string  | The name of the product. This is a unique identifier for the product.                          |
| description       | string  | A description of the product, which may contain HTML.  |
| is_active         | boolean | A boolean indicating if the product is active.   |
| default_milestone | string  | The name of the default milestone for the product.   |
| has_unconfirmed   | boolean | Indicates whether the UNCONFIRMED bug status is available for this product.                    |
| classification    | string  | The classification name for the product.   |
| components        | array   | Each component object has the items described in the Component object below.                   |
| versions          | array   | Each object describes a version, and has the following items : name, sort_key and is_active.   |
| milestones        | array   | Each object describes a milestone, and has the following items : name, sort_key and is_active. |

If the user tries to access a product that is not in the list of accessible products for the user, or a product that does not exist, that is silently ignored, and no information about that product is returned.

Component object :

| name                | type    | description   |
|---------------------|---------|---|
| id                  | int     | An integer ID uniquely identifying the component in this installation only.   |
| name                | string  | The name of the component. This is a unique identifier for this component.  |
| description         | string  | A description of the component, which may contain HTML.   |
| default_assigned_to | string  | The login name of the user to whom new bugs will be assigned by default.  |
| default_qa_contact  | string  | The login name of the user who will be set as the QA Contact for new bugs by default. Empty string if the QA contact is not defined.  |
| sort_key            | int     | Components, when displayed in a list, are sorted first by this integer and then secondly by their name.   |
| is_active           | boolean | A boolean indicating if the component is active. Inactive components are not enabled for new bugs.  |
| flag_types          | object  | An object containing two items <code>bug</code> and <code>attachment</code> that each contains an array of objects, where each describes a flagtype. The flagtype items are described in the Flagtype object below. |

Flagtype object :

| name              | type    | description   |
|-------------------|---------|---|
| id                | int     | Returns the ID of the flagtype.   |
| name              | string  | Returns the name of the flagtype.   |
| description       | string  | Returns the description of the flagtype.  |
| cc_list           | string  | Returns the concatenated CC list for the flagtype, as a single string.  |
| sort_key          | int     | Returns the sortkey of the flagtype.  |
| is_active         | boolean | Returns whether the flagtype is active or disabled. Flags being in a disabled flagtype are not deleted. It only prevents you from adding new flags to it.         |
| is_requestable    | boolean | Returns whether you can request for the given flagtype (i.e. whether the '?' flag is available or not).   |
| is_requestable_to | boolean | Returns whether you can ask someone specifically or not.  |
| is_multiplicable  | boolean | Returns whether you can have more than one flag for the given flagtype in a given bug/attachment.   |
| grant_group       | int     | the group ID that is allowed to grant/deny flags of this type. If the item is not included all users are allowed to grant/deny this flagtype.                     |
| request_group     | int     | The group ID that is allowed to request the flag if the flag is of the type requestable. If the item is not included all users are allowed request this flagtype. |

## Create Product

This allows you to create a new product in Bugzilla.

### Request

POST /rest/product

```
{
  "name" : "AnotherProduct",
  "description" : "Another Product",
  "classification" : "Unclassified",
  "is_open" : false,
  "has_unconfirmed" : false,
  "version" : "unspecified"
}
```

Some params must be set, or an error will be thrown. The required params are marked in bold.

| name               | type    | description  |
|--------------------|---------|--|
| <b>name</b>        | string  | The name of this product. Must be globally unique within Bugzilla.                         |
| <b>description</b> | string  | A description for this product. Allows some simple HTML.                                   |
| <b>version</b>     | string  | The default version for this product.  |
| has_unconfirmed    | boolean | Allow the UNCONFIRMED status to be set on bugs in this product. Default : true.            |
| classification     | string  | The name of the Classification which contains this product.                                |
| default_milestone  | string  | The default milestone for this product. Default '—'.                                       |
| is_open            | boolean | true if the product is currently allowing bugs to be entered into it. Default : true.      |
| create_series      | boolean | true if you want series for New Charts to be created for this new product. Default : true. |

### Response

```
{
  "id": 20
}
```

Returns an object with the following items :

| name | type | description                    |
|------|------|--------------------------------|
| id   | int  | ID of the newly-filed product. |

### Update Product

This allows you to update a product in Bugzilla.

#### Request

```
PUT /rest/product/(id_or_name)
```

You can edit a single product by passing the ID or name of the product in the URL. To edit more than one product, you can specify addition IDs or product names using the `ids` or `names` parameters respectively.

```
{
  "ids" : [123],
  "name" : "BarName",
  "has_unconfirmed" : false
}
```

One of the below must be specified.

| name              | type  | description  |
|-------------------|-------|--|
| <b>id_or_name</b> | mixed | Integer product ID or name.                          |
| <b>ids</b>        | array | Numeric IDs of the products that you wish to update. |
| <b>names</b>      | array | Names of the products that you wish to update.       |

The following parameters specify the new values you want to set for the product(s) you are updating.

| name              | type    | description   |
|-------------------|---------|---|
| name              | string  | A new name for this product. If you try to set this while updating more than one product, an error will occur, as product names must be unique.             |
| default_milestone | string  | When a new bug is filed, what milestone does it get by default if the user does not choose one ? Must represent a milestone that is valid for this product. |
| description       | string  | Update the long description for these products to this value.   |
| has_unconfirmed   | boolean | Allow the UNCONFIRMED status to be set on bugs in products.   |
| is_open           | boolean | true if the product is currently allowing bugs to be entered into it, false otherwise.  |

### Response

```
{
  "products" : [
    {
      "id" : 123,
      "changes" : {
        "name" : {
          "removed" : "FooName",
          "added" : "BarName"
        },
        "has_unconfirmed" : {
          "removed" : "1",
          "added" : "0"
        }
      }
    }
  ]
}
```

products (array) Product change objects containing the following items :

| name    | type   | description  |
|---------|--------|--|
| id      | int    | The ID of the product that was updated.  |
| changes | object | The changes that were actually done on this product. The keys are the names of the fields that were changed, and the values are an object with two items : <ul style="list-style-type: none"> <li>— added : (string) The value that this field was changed to.</li> <li>— removed : (string) The value that was previously set in this field.</li> </ul> |

Booleans will be represented with the strings '1' and '0' for changed values as they are stored as strings in the database currently.

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

## 6.1.13 Users

This part of the Bugzilla API allows you to create user accounts, get information about user accounts and to log in or out using an existing account.

### Login

Logging in with a username and password is required for many Bugzilla installations, in order to search for private bugs, post new bugs, etc. This method allows you to retrieve a token that can be used as authentication for subsequent API calls. Otherwise you will need to pass your `login` and `password` with each call.

This method will be going away in the future in favor of using *API keys*.

#### Request

```
GET /rest/login?login=foo@example.com&password=toosecrettoshow
```

| name             | type    | description  |
|------------------|---------|--|
| <b>login</b>     | string  | The user's login name.   |
| <b>password</b>  | string  | The user's password.   |
| res-strict_login | boolean | If set to a true value, the token returned by this method will only be valid from the IP address which called this method. |

#### Response

```
{
  "token": "786-OLaWfBisMY",
  "id": 786
}
```

| name  | type   | description  |
|-------|--------|--|
| id    | int    | Numeric ID of the user that was logged in.   |
| token | string | Token which can be passed in the parameters as authentication in other calls. The token can be sent along with any future requests to the webservice, for the duration of the session, i.e. til <i>Logout</i> is called. |

### Logout

Log out the user. Basically it invalidates the token provided so it cannot be re-used. Does nothing if the token is not in use. Will also clear any existing authentication cookies the client may still have stored.

#### Request

```
GET /rest/logout?token=1234-VWvO51X69r
```

| name  | type   | description                               |
|-------|--------|---|
| token | string | The user's token used for authentication. |

### Valid Login

This method will verify whether a client's cookies or current login token is still valid or have expired. A valid username that matches must be provided as well.

#### Request

GET /rest/valid\_login?login=foo@example.com&token=1234-VWvO51X69r

| name  | type   | description   |
|-------|--------|---|
| login | string | The login name that matches the provided cookies or token.      |
| token | string | Persistent login token currently being used for authentication. |

### Response

Returns true/false depending on if the current token is valid for the provided username.

### Create User

Creates a user account directly in Bugzilla, password and all. Instead of this, you should use **Offer Account by Email** when possible because that makes sure that the email address specified can actually receive an email. This function does not check that. You must be authenticated and be in the *editusers* group to perform this action.

### Request

POST /rest/user

```
{
  "email" : "user@bugzilla.org",
  "full_name" : "Test User",
  "password" : "K16ldRr922I1"
}
```

| name      | type   | description  |
|-----------|--------|--|
| email     | string | The email address for the new user.  |
| full_name | string | The user's full name. Will be set to empty if not specified.   |
| password  | string | The password for the new user account, in plain text. It will be stripped of leading and trailing whitespace. If blank or not specified, the new created account will exist in Bugzilla but will not be allowed to log in using DB authentication until a password is set either by the user (through resetting their password) or by the administrator. |

### Response

```
{
  "id": 58707
}
```

| name | type | description                                  |
|------|------|--|
| id   | int  | The numeric ID of the user that was created. |

### Update User

Updates an existing user account in Bugzilla. You must be authenticated and be in the *editusers* group to perform this action.

### Request

PUT /rest/user/(id\_or\_name)

You can edit a single user by passing the ID or login name of the user in the URL. To edit more than one user, you can specify addition IDs or login names using the *ids* or *names* parameters respectively.

| name                 | type           | description   |
|----------------------|----------------|---|
| <b>id_or_name</b>    | mixed          | Either the ID or the login name of the user to update.  |
| <b>ids</b>           | array          | Additional IDs of users to update.  |
| <b>names</b>         | array          | Additional login names of users to update.  |
| <b>full_name</b>     | string         | The new name of the user.   |
| <b>email</b>         | string         | The email of the user. Note that email used to login to bugzilla. Also note that you can only update one user at a time when changing the login name / email. (An error will be thrown if you try to update this field for multiple users at once.) |
| <b>password</b>      | string         | The password of the user.   |
| <b>email_enabled</b> | boolean        | A boolean value to enable/disable sending bug-related mail to the user.   |
| <b>login_denied</b>  | string<br>text | A text field that holds the reason for disabling a user from logging into Bugzilla. If empty, then the user account is enabled ; otherwise it is disabled/closed.   |
| <b>groups</b>        | object         | These specify the groups that this user is directly a member of. To set these, you should pass an object as the value. The object's items are described in the Groups update objects below.   |
| <b>bless_groups</b>  | object         | This is the same as groups but affects what groups a user has direct membership to bless that group. It takes the same inputs as groups.  |

Groups and bless groups update object :

| name          | type  | description   |
|---------------|-------|---|
| <b>add</b>    | array | The group IDs or group names that the user should be added to.  |
| <b>remove</b> | array | The group IDs or group names that the user should be removed from.  |
| <b>set</b>    | array | Integers or strings which are an exact set of group IDs and group names that the user should be a member of. This does not remove groups from the user when the person making the change does not have the bless privilege for the group. |

If you specify `set`, then `add` and `remove` will be ignored. A group in both the `add` and `remove` list will be added. Specifying a group that the user making the change does not have bless rights will generate an error.

## Response

— `users` : (array) List of user change objects with the following items :

| name           | type   | description  |
|----------------|--------|--|
| <b>id</b>      | int    | The ID of the user that was updated.   |
| <b>changes</b> | object | The changes that were actually done on this user. The keys are the names of the fields that were changed, and the values are an object with two items : <ul style="list-style-type: none"> <li>— <code>added</code> : (string) The values that were added to this field, possibly a comma-and-space-separated list if multiple values were added.</li> <li>— <code>removed</code> : (string) The values that were removed from this field, possibly a comma-and-space-separated list if multiple values were removed.</li> </ul> |

## Get User

Gets information about user accounts in Bugzilla.

### Request

To get information about a single user in Bugzilla :

```
GET /rest/user/(id_or_name)
```

To get multiple users by name or ID :

```
GET /rest/user?names=foo@bar.com&name=test@bugzilla.org
```

```
GET /rest/user?ids=123&ids=321
```

To get user matching a search string :

```
GET /rest/user?match=foo
```

To get user by using an integer ID value or by using `match`, you must be authenticated.

| name             | type    | description  |
|------------------|---------|--|
| id_or_name       | mixed   | An integer user ID or login name of the user.  |
| ids              | array   | Integer user IDs. Logged-out users cannot pass this parameter to this function. If they try, they will get an error. Logged-in users will get an error if they specify the ID of a user they cannot see.   |
| names            | array   | Login names.   |
| match            | array   | This works just like “user matching” in Bugzilla itself. Users will be returned whose real name or login name contains any one of the specified strings. Users that you cannot see will not be included in the returned list.<br>Most installations have a limit on how many matches are returned for each string ; the default is 1000 but can be changed by the Bugzilla administrator.<br>Logged-out users cannot use this argument, and an error will be thrown if they try. (This is to make it harder for spammers to harvest email addresses from Bugzilla, and also to enforce the user visibility restrictions that are implemented on some Bugzillas.) |
| limit            | int     | Limit the number of users matched by the <code>match</code> parameter. If the value is greater than the system limit, the system limit will be used. This parameter is only valid when using the <code>match</code> parameter.   |
| group_ids        | array   | Numeric IDs for groups that a user can be in.  |
| groups           | array   | Names of groups that a user can be in. If <code>group_ids</code> or <code>groups</code> are specified, they limit the return value to users who are in <i>any</i> of the groups specified.   |
| include_disabled | boolean | By default, when using the <code>match</code> parameter, disabled users are excluded from the returned results unless their full username is identical to the match string. Setting <code>include_disabled</code> to <code>true</code> will include disabled users in the returned results even if their username doesn't fully match the input string.  |

### Response

— `users` : (array) Each object describes a user and has the following items :



| name           | type    | description   |
|----------------|---------|---|
| id             | int     | The unique integer ID that Bugzilla uses to represent this user. Even if the user's login name changes, this will not change.   |
| real_name      | string  | The actual name of the user. May be blank.  |
| email          | string  | The email address of the user.  |
| name           | string  | The login name of the user. Note that in some situations this is different than their email.  |
| can_login      | boolean | A boolean value to indicate if the user can login into bugzilla.  |
| email_enabled  | boolean | A boolean value to indicate if bug-related mail will be sent to the user or not.  |
| login_denied   | string  | A text field that holds the reason for disabling a user from logging into Bugzilla. If empty then the user account is enabled ; otherwise it is disabled/closed.  |
| groups         | array   | Groups the user is a member of. If the currently logged in user is querying their own account or is a member of the 'editusers' group, the array will contain all the groups that the user is a member of. Otherwise, the array will only contain groups that the logged in user can bless. Each object describes the group and contains the items described in the Group object below. |
| saved_searches | array   | User's saved searches, each having the following Search object items described below.   |
| saved_reports  | array   | User's saved reports, each having the following Search object items described below.  |

Group object :

| name        | type   | description                   |
|-------------|--------|-------------------------------|
| id          | int    | The group ID                  |
| name        | string | The name of the group         |
| description | string | The description for the group |

Search object :

| name  | type   | description  |
|-------|--------|--|
| id    | int    | An integer ID uniquely identifying the saved report. |
| name  | string | The name of the saved report.                        |
| query | string | The CGI parameters for the saved report.             |

If you are not authenticated when you call this function, you will only be returned the `id`, `name`, and `real_name` items. If you are authenticated and not in 'editusers' group, you will only be returned the `id`, `name`, `real_name`, `email`, `can_login`, and `groups` items. The groups returned are filtered based on your permission to bless each group. The `saved_searches` and `saved_reports` items are only returned if you are querying your own account, even if you are in the editusers group.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).



Vous devez utiliser MySQL version 5.0.15 ou supérieur.

Il est possible de vérifier la version utilisée en utilisant la commande suivante :

```
mysql -V
```

## 7.1 Installer

### 7.1.1 Windows

Téléchargez l'installateur MSI MySQL 32-bit ou 64-bit sur le [site Web de MySQL](#) (~28 Mo).

MySQL a un installateur Windows standard. Vous pouvez utiliser l'installation MySQL typique (par défaut). LA suite de cette documentation suppose que vous avez installé MySQL dans `C:\mysql`. Si ce n'est pas le cas, ajustez les chemins d'accès en conséquence.

### 7.1.2 Linux/Mac OS X

Les instructions d'installation de paquets données précédemment devraient avoir installé MySQL sur votre machine. Si ce n'est pas le cas, exécutez la commande :

```
mysql_secure_installation
```

et suivez les instructions affichées.

Si vous avez installé MySQL manuellement plutôt qu'avec le gestionnaire de paquets, assurez-vous que le serveur est lancé au démarrage de la machine.

## 7.2 Ajouter un utilisateur

Vous devez ajouter un nouvel utilisateur MySQL pour Bugzilla. Lancez le client en ligne de commande `mysql` et saisissez le texte suivant :

```
GRANT SELECT, INSERT,  
UPDATE, DELETE, INDEX, ALTER, CREATE, LOCK TABLES,  
CREATE TEMPORARY TABLES, DROP, REFERENCES ON bugs.*  
TO bugs@localhost IDENTIFIED BY '$DB_PASS';
```

```
FLUSH PRIVILEGES;
```

Vous devez remplacer `$DB_PASS` par un mot de passe suffisamment fort que vous avez choisi. Notez ce mot de passe quelque part.

La commande ci-dessus permet à un compte appelé `bugs` de se connecter sur la machine locale, `localhost`. Modifiez la commande en conséquence pour s'ajuster à votre configuration si vous vous connectez à partir d'une autre machine ou si vous utilisez un utilisateur différent.

## 7.3 Changer la configuration

**Pour changer la configuration de MySQL, vous devez éditer votre fichier de configuration MySQL** qui se trouve dans :

- Red Hat/Fedora : `/etc/my.cnf`
- Debian/Ubuntu : `/etc/mysql/my.cnf`
- Windows : `C:\mysql\bin\my.ini`
- Mac OS X : `/etc/my/cnf`

### 7.3.1 Allow Large Attachments and Many Comments

Par défaut, MySQL ne vous autorisera à insérer dans la base de données, que des objets plus petits que 1 Mo.

Les fichiers joints peuvent être plus gros que cela. De même, Bugzilla concatène tous les commentaires d'un bogue dans un seul champ pour la recherche plein texte, et la somme de tous les commentaires d'un bogue sera vraisemblablement supérieure à 1 Mo.

Nous vous recommandons d'autoriser au moins des paquets de 16 Mo en ajoutant le paramètre `max_allowed_packet` dans votre fichier de configuration MySQL dans la section `[mysqld]`, comme ceci (notez que c'est M et pas MB ou Mo) :

```
[mysqld]
# Allow packets up to 16M
max_allowed_packet=16M
```

### 7.3.2 Autoriser les mots courts dans les index texte intégral

Par défaut, les mots doivent avoir au moins quatre caractères afin d'être indexés par les index texte intégral de MySQL. Ceci fait que beaucoup de mots clés spécifiques peuvent être manqués, y compris `cc`, `ftp` et `uri`.

MySQL peut être configuré pour indexer ces mots en définissant le paramètre `ft_min_word_len` pour la taille minimale des mots à indexer.

```
[mysqld]
# Allow small words in full-text indexes
ft_min_word_len=2
```

## 7.4 Autoriser la table des fichiers joints à dépasser les 4 Go

Ceci est une configuration optionnelle pour des installations de Bugzilla que l'on s'attend à voir devenir très grosses et doit être fait après que Bugzilla a été complètement installé.

Par défaut, MySQL limitera la table à 4 Go. Cette limite est présente même si le système d'exploitation sur lequel est exécuté MySQL n'a pas cette limite. Pour définir une limite plus grande, lancez le client en ligne de commande `mysql` et saisissez ce qui suit en remplaçant `$bugs_db` par le nom de votre base de données (qui est `bugs` par défaut) :

```
USE $bugs_db;
```

```
ALTER TABLE attachments AVG_ROW_LENGTH=1000000, MAX_ROWS=20000;
```

La commande ci-dessus changera la limite à 20 Go. MySQL devra faire une copie temporaire de toute la table pour changer la limite, donc idéalement ceci doit être fait quand la table des fichiers joints est encore petite.

---

**Note :** Si vous avez défini dans Bugzilla le paramètre permettant de stocker les gros fichiers joints sur le disque dur, le changement ci-dessus n'aura aucun effet.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).



---

## PostgreSQL

---

Testez la version de PostgreSQL que vous avez installée avec la commande suivante :

**psql -V**

Vous devez utiliser PostgreSQL version 8.03.0000 ou supérieure.

Si vous installez PostgreSQL manuellement plutôt qu'avec le gestionnaire de paquets, assurez-vous que le serveur soit lancé au démarrage de la machine.

### 8.1 Ajouter un utilisateur

Vous devez ajouter un nouvel utilisateur PostgreSQL pour que l'application Bugzilla accède à la base de données. Les instructions suivantes que vous utilisez les paramètres par défaut dans `localconfig`. Si vous les avez changés, vous devrez modifier les commandes en conséquence.

Sur la plupart des systèmes, pour créer un utilisateur PostgreSQL, il faut se connecter en tant que `root` puis se substituer à l'utilisateur (Linux) `postgres` :

**su - postgres**

Vous pourrez alors créer un nouvel utilisateur :

**createuser -U postgres -dRSP bugs**

Lors de la demande du mot de passe, fournissez-le et notez quelque part pour plus tard.

L'utilisateur créé ne sera pas super-utilisateur (-S) et ne pourra pas créer de nouveaux utilisateurs (-R). Il aura seulement la possibilité de créer des bases de données (-d).

### 8.2 Autorisations d'accès

Modifiez le fichier `pg_hba.conf` qui se trouve habituellement dans `/var/lib/pgsql/data/`. Dans ce fichier, vous devrez ajouter une nouvelle ligne comme suit :

```
host all bugs 127.0.0.1 255.255.255.255 md5
```

Ceci signifie que pour les connexions TCP/IP (hôte), sont autorisées les connexions à partir de `127.0.0.1` pour `all` toutes les bases de données sur ce serveur pour l'utilisateur `bugs` utilisant l'authentification de mot de passe `md5`.

Vous devez maintenant arrêter totalement PostgreSQL et le redémarrer. (N'utilisez pas la commande `restart` en raison de la probabilité d'une modification dans le fichier `postgresql.conf`).

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).



**Avvertissement :** Bugzilla gère Oracle, mais aucun des développeurs actuels ne l'utilisent.

Vous devez utiliser Oracle version 10.02.0 ou supérieure.

## 9.1 Créer un nouveau tablespace

Vous pouvez utiliser un tablespace existant ou en créer un nouveau pour Bugzilla. Pour créer un nouveau tablespace, exécuter la commande suivante dans `sqlplus` :

```
CREATE TABLESPACE bugs
DATAFILE '*$chemin_d_accès_au_fichier*' SIZE 500M
AUTOEXTEND ON NEXT 30M MAXSIZE UNLIMITED
```

Ici le nom du tablespace est `bugs`, mais vous pouvez choisir un autre nom. *\$chemin\_d\_accès\_au\_fichier* est le chemin d'accès complet au fichier contenant votre base de données, par exemple `/u01/oradata/bugzilla.dbf`. La taille initiale du fichier de base de données dans cet exemple est de 500 Mo, avec un incrément de 30 Mo chaque fois que la taille limite du fichier est atteinte.

## 9.2 Ajouter un utilisateur Oracle

Le nom d'utilisateur et le mot de passe doivent correspondre à ce que vous avez défini dans `localconfig` (`$db_user` et `$db_pass`, respectivement). Ici, nous supposons que l'utilisateur s'appelle `bugs` et que le nom de tablespace est le même que ci-dessus.

```
CREATE USER bugs
IDENTIFIED BY "$db_pass"
DEFAULT TABLESPACE bugs
TEMPORARY TABLESPACE TEMP
PROFILE DEFAULT;
-- GRANT/REVOKE ROLE PRIVILEGES
GRANT CONNECT TO bugs;
GRANT RESOURCE TO bugs;
-- GRANT/REVOKE SYSTEM PRIVILEGES
GRANT UNLIMITED TABLESPACE TO bugs;
GRANT EXECUTE ON CTXSYS.CTX_DDL TO bugs;
```

## 9.3 Configurer le serveur Web

Si vous utilisez Apache, ajoutez ces lignes au fichier `httpd.conf` pour définir les variables `ORACLE_HOME` et `LD_LIBRARY_PATH`. Par exemple :

```
SetEnv ORACLE_HOME /u01/app/oracle/product/10.2.0/  
SetEnv LD_LIBRARY_PATH /u01/app/oracle/product/10.2.0/lib/
```

Quand ceci est fait, redémarrez le serveur Web.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

---

### SQLite

---

**Avertissement :** En raison des [limitations de concurrence](#) nous recommandons SQLite uniquement pour les petites installations ou les installations de développement.

Une fois SQLite installé, aucune configuration supplémentaire n'est nécessaire pour utiliser Bugzilla.

La base de données sera stockée dans `$BUGZILLA_HOME/data/db/$db_name`, où `$db_name` est le nom de la base de données défini dans le fichier `localconfig`.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).



---

## Apache

---

Vous avez deux options pour exécuter Bugzilla sur Apache : `mod_cgi` (par défaut) et `mod_perl`. `mod_perl` est plus rapide mais consomme plus de ressources. Vous ne devriez envisager l'utilisation de `mod_perl` que si votre installation de Bugzilla sera très firtement utilisée.

Ces instructions nécessitent l'édition du fichier de configuration d'Apache :

- Fedora/Red Hat : `/etc/httpd/conf/httpd.conf`
- Debian/Ubuntu : `/etc/apache2/apache2.conf`
- Mac OS X : `/etc/apache2/httpd.conf`

Vous pouvez aussi sous Debian ou Ubuntu, mettre le code indiqué ci-dessous dans un fichier séparé dans le répertoire `/etc/apache2/sites-enabled/`.

Dans ces instructions, quand le démarrage d'Apache est demandé, la commande à exécuter est :

**sudo apachectl start**

(ou l'exécuter en tant que root si votre système n'utilise pas sudo).

### 11.1 Sécuriser Apache

Quand des systèmes externes interagissent avec Bugzilla par le biais de webservices (REST/XMLRPC/JSONRPC), ils incluent les identifiants de connexions de l'utilisateur dans l'URL (dans la « query string »). Par conséquent, pour éviter de stocker les mots de passe en clair dans les journaux du serveurs, nous recommandons de configurer Apache pour ne pas inclure la « query string » dans ses fichiers journaux.

1. Éditer le fichier de configuration d'Apache (voir plus haut).
2. Rechercher la ligne suivante qui définit le format de journalisation pour `vhost_combined` :

```
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
```

3. Remplacer `%r` par `%m %U`.
4. Redémarrer Apache.

### 11.2 Apache with mod\_cgi

Pour configurer votre serveur Web Apache pour fonctionner avec Bugzilla en utilisant `mod_cgi`, suivez les instructions suivantes :

1. Éditer le fichier de configuration d'Apache (voir plus haut).

2. Créer une directive `<Directory>` s'appliquant à l'emplacement de votre installation Bugzilla. dans cette exemple, Bugzilla a été installé dans `/var/www/html/bugzilla`. Sous Mac OS X, utiliser `/Library/WebServer/Documents/bugzilla`.

```
<Directory /var/www/html/bugzilla>
  AddHandler cgi-script .cgi
  Options +ExecCGI +FollowSymLinks
  DirectoryIndex index.cgi index.html
  AllowOverride Limit FileInfo Indexes Options
</Directory>
```

Ces instructions autorisent Apache à exécuter les fichiers `.cgi` se trouvant dans le répertoire `bugzilla`; indiquent au serveur de chercher un fichier appelé `index.cgi`, ou s'il n'est pas trouvé, `index.html`, si quelqu'un saisit seulement le nom du répertoire dans le navigateur; et autorisent les fichiers `.htaccess` de Bugzilla à outrepasser quelques permissions globales.

## 11.3 Apache with mod\_perl

Une certaine configuration est requise pour faire fonctionner Bugzilla avec Apache et `mod_perl`.

---

**Note :** Nous ne savons pas si quelqu'un a déjà essayé `mod_perl` sur Mac OS X.

---

1. Éditer le fichier de configuration d'Apache (voir plus haut).
2. Ajoutez les informations suivantes en substituant où cela est approprié vos propres chemins d'accès locaux.

```
PerlSwitches -w -T
PerlConfigRequire /var/www/html/bugzilla/mod_perl.pl
```

---

**Note :** Ceci doit être utilisé au lieu du bloc `<Directory>` indiqué plus haut. Ceci doit aussi être placé au-dessus de toute autre directive `mod_perl` dans le fichier `httpd.conf` et doit être spécifié dans l'ordre indiqué ci-dessous.

**Avertissement :** Vous devez aussi vous assurer que vous avez désactivé la gestion de `KeepAlive` dans votre installation Apache en utilisant Bugzilla avec `mod_perl` ou vous pourriez rencontrer des [problèmes de performance](#).

Au redémarrage d'Apache, Bugzilla devrait alors fonctionner en environnement `mod_perl`. Veuillez vous assurer d'avoir exécuté `checksetup.pl` pour définir les permissions avant de redémarrer Apache.

Veuillez garder à l'esprit les points suivants en utilisant Bugzilla avec `mod_perl` :

- La gestion `mod_perl` dans Bugzilla peut consommer ÉNORMÉMENT de RAM. Vous pouvez compter facilement 30 Mo par processus `httpd` enfant. En gros, vous avez seulement besoin de beaucoup de RAM. Plus vous en aurez, mieux ce sera. `mod_perl` utilise basiquement la mémoire pour augmenter la vitesse de traitement. Il est recommandé d'avoir au moins 2 Go de RAM pour exécuter Bugzilla sous `mod_perl`.
- Sous `mod_perl`, vous devez redémarrer Apache si vous faites un changement manuel sur tout fichier de Bugzilla. vous ne pouvez pas seulement recharger – vous devez en fait *redémarrer* le serveur (être sûr qu'il soit arrêté puis démarré à nouveau) Vous *pouvez* modifier le fichier `localconfig` et le fichier des paramètres, si vous le voulez, car ceux-ci sont lus chaque fois que vous chargez une page.
- Vous devez exécuter `Prefork` MPM de Apache (c'est l'option par défaut). Le `Worker` MPM peut ne pas fonctionner – nous n'avons pas testé Bugzilla sous `mod_perl` avec la gestion des threads. (Et en fait, nous sommes pratiquement sûrs que cela ne fonctionnera *pas*.)

- Bugzilla s'attend généralement à être la seule application fonctionnant sous mod\_perl sur tout le serveur. Il peut ne pas fonctionner s'il y a d'autres applications fonctionnant aussi sous mod\_perl. Il s'efforcera de cohabiter avec d'autres applications sous mod\_perl, mais il pourra y avoir des conflits.
  - Il est recommandé de n'avoir qu'une seule instance de Bugzilla fonctionnant sous mod\_perl sur votre serveur. Bugzilla n'a pas été testé avec plus d'une instance.
- 

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).





---

## Apache

---

Ces instructions nécessitent d'éditer le fichier de configuration Apache, qui se trouve dans `C:\Program Files\Apache Group\Apache2\conf\httpd.conf`.

### 12.1 Installation

Télécharger Apache HTTP Server, version 2.2.x ou supérieure, sur le [site Web d'Apache](#).

Apache utilise un installateur Windows standard. Suivez les instructions et assurez-vous d'utiliser l'option « Installer pour tous les utilisateurs ». Apache s'installera toujours dans un répertoire `Apache2` dans le chemin que vous avez indiqué. Le répertoire d'installation par défaut est `C:\Program Files\Apache Group`, et donc Apache sera installé dans `C:\Program Files\Apache Group\Apache2`.

Si vous exécutez déjà IIS sur votre machine, vous devez configurer Apache pour qu'il s'exécute sur un autre port que le port 80 qu'utilise IIS. Cependant, le numéro de port n'est pas demandé pendant l'installation. Il pourra être changé plus tard.

dans ce document, nous supposons que Apache a été installé dans le répertoire par défaut : `C:\Program Files\Apache Group\Apache2`.

### 12.2 Droits d'accès Apache

Par défaut, Apache est installé pour s'exécuter avec le compte SYSTEM. Pour des raisons de sécurité, il est préférable de reconfigurer le service pour qu'il s'exécute avec un utilisateur Apache. Pour cela, créer un utilisateur Windows membre d'**aucun** groupe et reconfigurer le service `Apache2` pour qu'il s'exécute avec ce compte.

Quel que soit le compte avec lequel Apache s'exécute, celui a besoin des droits d'écriture et de modification pour les répertoires suivants et leurs sous-répertoires. Selon la version de Windows, ces droits peuvent déjà être donnés.

- `C:\Bugzilla\data`
- `C:\Program Files\Apache Group\Apache2\logs`
- `C:\Temp`
- `C:\Windows\Temp`

Veillez noter que le répertoire `C:\Bugzilla\data` est créé lors du premier lancement du script `checksetup.pl`.

## 12.3 Port et DocumentRoot

Modifier le fichier de configuration Apache (voir plus haut).

Si vous devez modifier le port utilisé par Apache (`listen` ou `bind`), par exemple parce qu'un autre serveur Web tel que IIS s'exécute sur le même serveur, modifiez l'option `Listen` et changez la valeur après les deux points.

Modifiez le paramètre `DocumentRoot` pour qu'il pointe vers `C:/Bugzilla`. Il y a deux endroits dans `httpd.conf` qui nécessitent d'être mis à jour (recherchez `DocumentRoot`). Vous devez utiliser `/` au lieu de `\` comme séparateur de chemin.

## 12.4 Activer la gestion de CGI

Modifier le fichier de configuration d'Apache (voir plus haut).

Pour activer la gestion de CGI dans Apache, vous devez activer le handler CGI, en décommentant la ligne `AddHandler cgi-script .cgi`.

## 12.5 Indiquer l'emplacement de Bugzilla

Modifiez le fichier de configuration (voir plus haut).

Ajoutez le bloc suivant :

```
<Directory "C:/Bugzilla">
    ScriptInterpreterSource Registry-Strict
    Options +ExecCGI +FollowSymLinks
    DirectoryIndex index.cgi index.html
    AllowOverride Limit FileInfo Indexes Options
</Directory>
```

Pour que `ScriptInterpreterSource Registry-Strict` fonctionne, vous devez aussi ajouter une entrée dans la base de registres de Windows pour qu'Apache utilise Perl pour exécuter les fichiers « `.cgi` » files.

Créez une clé `HKEY_CLASSES_ROOT\.cgi\Shell\ExecCGI\Command` avec la valeur par défaut du chemin d'accès complet à l'exécutable `perl.exe` avec le paramètre `-T`. Par exemple : `C:\Perl\bin\perl.exe -T`.

## 12.6 Fichiers journaux

À moins que vous ne vouliez conserver les statistiques d'accès à votre installation Bugzilla, il est préférable de désactiver la journalisation en commentant la directive `CustomLog` dans le fichier de configuration d'Apache.

Si vous ne désactivez pas la journalisation, vous devrez au moins désactiver celle des « query strings ». Quand des systèmes externes interagissent avec Bugzilla par l'intermédiaire de webservices (REST/XMLRPC/JSONRPC), ceux-ci incluent les identifiants de l'utilisateur dans l'URL (la « query string »). Par conséquent, pour éviter de stocker les mots de passe en clair sur le serveur, nous recommandons de configurer Apache pour ne pas inclure la chaîne de requête dans ses fichiers journaux.

1. recherchez la ligne suivante dans le fichier de configuration d'Apache, qui définit le formatage pour `vhost_combined`:

```
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
```

2. Replace %r with %m %U.

(Si vous avez configuré Apache différemment, une ligne différente pour la journalisation pourrait s'appliquer. Ajustez alors ces instructions en conséquence.)

## 12.7 Redémarrer Apache

Enfin, redémarrez Apache pour qu'il prenne en compte les modifications :

```
net stop apache2
```

```
net start apache2
```

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).



---

## Microsoft IIS

---

Bugzilla fonctionne avec IIS comme une application CGI classique. Ces instructions supposent que vous utilisez Windows 7. Les procédures pour les autres versions sont probablement similaires.

Commencer par démarrer le Gestionnaire des services Internet (IIS). *Démarrer* → *Outils d'administration* → *Gestionnaire des services Internet (IIS)*. Ou lancer la commande :

**inetmgr**

### 13.1 Créer une nouvelle Application

Développer votre *Serveur* jusqu'à ce que *Default Web Site* affiche ses éléments.

Faire un clic droit sur *Default Web Site* et sélectionner *Ajouter une application...* dans le menu contextuel.

Sous *Alias*, saisir l'alias du site Web. C'est le chemin pour le domaine où vous voulez voir apparaître Bugzilla.

Dans *Chemin d'accès physique*, saisir le chemin d'accès à Bugzilla, par exemple `C:\Bugzilla`.

Quand c'est terminé, cliquer sur *OK*.

### 13.2 Configurer le document par défaut

Cliquer sur l'application qui vient d'être créée. Faire un double clic sur *Document par défaut*, et cliquer sur *Ajouter...* dans le menu Actions.

Dans *Nom*, saisir `index.cgi`.

Tous les autres documents par défaut peuvent être supprimés pour cette application.

|  |
|--|
| <b>Avertissement :</b> Ne pas supprimer le document par défaut pour <i>Site Web par défaut</i> . |
|--|

### 13.3 Ajouter un mappage de gestionnaire

Assurez-vous d'être dans le site Web par défaut. Dans la section *IIS*, faites un double clic sur *Mappages de gestionnaires*. Dans la section *Actions* à droite, cliquez sur *Ajouter un mappage de scripts...* Vous devez faire ceci à deux reprises.

Pour le premier, indiquez les valeurs suivantes (en modifiant les chemins d'accès si nécessaires) :

- *Chemin des demandes* : \*.pl
- *Exécutable* : C:\Perl\bin\perl.exe "%s" %s%
- *Nom* : Mappage de script Perl

Cliquez sur *OK* et sur l'invite qui apparaît, cliquez sur *Oui*.

---

**Note** : L'installateur ActiveState Perl a peut-être déjà créé cette entrée pour les fichiers .pl limité à GET, HEAD, POST. Si c'est le cas, ce mappage doit être supprimé car les fichiers \*.pl de Bugzilla ne sont pas conçus pour être exécutés par un serveur Web.

---

Pour le second, indiquez les valeurs suivantes (adaptez les chemins si nécessaire) :

- *Chemin des demandes* : \*.cgi
- *Exécutable* : C:\Perl\bin\perl.exe "%s" %s%
- *Nom* : CGI Script Map

Cliquez sur *OK* et sur l'invite qui apparaît, cliquez sur *Oui*.

## 13.4 Application Bugzilla

Assurez-vous d'être dans l'application Bugzilla. Dans la section *IIS* à droite, faites un double clic sur *Mappages de gestionnaires*. Dans la section *Actions*, cliquez sur *Ajouter un mappage de scripts...*

Indiquez les valeurs suivantes (adaptez les chemins si nécessaire) :

- *Chemin des demandes* : \*.cgi
- *Exécutable* : C:\Perl\bin\perl.exe -x"C:\Bugzilla" -wT "%s" %s
- *Nom* : Bugzilla

Cliquez sur *OK* et sur l'invite qui apparaît, cliquez sur *Oui*.

Il faut à présent redémarrer le serveur IIS pour que les changements soient pris en compte. À partir du menu supérieur, qui contient le nom de votre machine, cliquez sur *Redémarrer* sous *Gérer le serveur*. Ou exécuter la commande :

```
iisreset
```

## 13.5 Problèmes courants

**Bugzilla s'exécute mais il n'est pas possible de se connecter.** Vous avez probablement configuré IIS pour utiliser la DLL ISAPI d'ActiveState – en d'autres termes, vous utilisez PerlEx, ou IIS est configuré pour utiliser PerlS.dll ou Perl30.dll.

Reconfigurer IIS pour utiliser perl.exe.

**IIS renvoie des erreurs HTTP 502.** Vous avez probablement oublié l'argument -T dans la ligne de commande perl en configurant l'exécutable dans IIS.

**L'interface XMLRPC ne fonctionne pas avec IIS.** C'est un bogue connu. Voir bug 708252.

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).

---

Cette documentation contient très probablement des bogues ; si vous en découvrez, veuillez les signaler [ici](#).