

# Dragnet API Specification

Dragnet is a module directory derived from modules reported during the crash of the Firefox web browser. It is intended to assist developers in understanding the modules in use, as well as in highlighting malware and suspect module files at crash time.

## The Module Object

The Dragnet API returns JSON, and the objects it returns are module objects. Each module object contains the following keys:

Field Name	Default	Description
id	n/a	The ID of the module.
date_created	n/a	The date the module was created.
date_modified	Date created	This is the date that the module was last modified in the system; it defaults to date created if it has not been modified.
created_by	system	This is the user that created the module. Modules created by the system are created under the username "system".
modified_by	system	The user that last modified the module. Modules created by the system and are unmodified by another user have the username "system" as the last modifier.
file_name	n/a	This is the name of the module file.
common_name		Some files are given a common name which can be stored here.
version		The version of the file being stored in the system. Multiple files of the same name can be stored with different data; version helps distinguish which version is being returned.
platform		The platform (Windows, Linux, Mac OS X, or blank) that the module is found on.
vendor		The vendor that distributes the module file.
distributors		The distributors that distribute the module file.
md5_hash		Each module can be identified with a unique MD5 hash. If known, this is stored here.
debug		This is a debug hash provided by the module.
debug_filename		The debug filename provided by the module.
status		The status of the file (either blank, Unknown, Valid or Malware).
released		The date the file was released by the vendor/distributor.
obsolete	False	Whether or not the file is obsolete, either by decommissioning or being supplanted by an updated version.
replaced_by		The module file that replaced the current module file.
details		A free text field for details about the module file.

## Searching for Modules

Modules may have the same file name, due to the fact that these modules can be different. Therefore, it is possible to search for modules in different ways.

Modules are identified by three unique data points: the file name, the debug ID, and the debug filename. Whenever providing a module to search for, one or all of these components may be provided as a JSON object.

## Methods and Actions

### **POST: search/list**

Provides a list of modules based on the provided arguments.

#### **Arguments:**

Field Name	Required?	Description
modules	Yes	A JSON array of module search objects that you wish to query in the system.

#### **Response:**

Field Name	Description
found	A JSON array containing the modules that were found (and the module objects)
not_found	A JSON array containing strings of the module names that were not found.

### **GET: search/single**

Provides a single module object based on the module name or module ID provided.

NOTE: this may produce multiple results if there are multiple files with the same name; your application should be prepared for that possibility.

#### **Arguments:**

Field Name	Required?	Description
module OR id	Yes	Either the module search object or the ID of the module object to find.

#### **Response:**

Field Name	Description
modules	A JSON array containing the modules that were found (and the module objects). If nothing was found, this array will be empty.