

nsIHandlerService API Sync to Async Study

Eden

nsIHandlerService APIs

[nsIHandlerService.idl](#)

nsIHandlerService is a service used for accessing the mimeTypeypes.rdf.

```
void store(in nsIHandlerInfo aInfo)
```

```
void remove(in nsIHandlerInfo aInfo)
```

```
nsISimpleEnumerator enumerate()
```

```
boolean exists(in nsIHandlerInfo aInfo)
```

```
ACString getTypeFromExtension(in ACString aFileExtension)
```

```
void fillHandlerInfo(in nsIHandlerInfo aInfo,  
                    in ACString aOverrideType)
```

[nsIHandlerInfo](#)

Consumers of nsIHandlerService

Consumer	API
WebContentConverter.js	store()
PdfJs.jsm	store()
dialog.js	store()
application.js	store() enumerate()
nsExternalHelperAppService.cpp	exists() fillHandlerInfo() getTypeFromExtension()

Usage Study: store()

store() in WebContentConverter.js

```
455         handlerInfo.alwaysAskBeforeHandling = true;
456
457         let hs = Cc["@mozilla.org/uriloader/handler-service;1"].
458             getService(Ci.nsIHandlerService);
459         hs.store(handlerInfo);
460     }
461 };
```

Call nsIHandlerService::store() then do other thing

The result of store() is never checked

store() can be asynchronous

Usage Study: enumerate()

enumerate() in applications.js

```
1112  * Load the set of handlers defined by the application datastore.
1113  */
1114  _loadApplicationHandlers: function() {
1115      var wrappedHandlerInfos = this._handlerSvc.enumerate();
1116      while (wrappedHandlerInfos.hasMoreElements()) {
1117          let wrappedHandlerInfo =
1118              wrappedHandlerInfos.getNext().QueryInterface(Ci.nsIHandlerInfo);
1119          let type = wrappedHandlerInfo.type;
1120
1121          let handlerInfoWrapper;
1122          if (type in this._handledTypes)
1123              handlerInfoWrapper = this._handledTypes[type];
1124          else {
1125              handlerInfoWrapper = new HandlerInfoWrapper(type, wrappedHandlerInfo);
1126              this._handledTypes[type] = handlerInfoWrapper;
1127          }
1128
1129          handlerInfoWrapper.handledOnlyByPlugin = false;
1130      }
1131  },
```

_loadApplicationHandlers is called when opening application panel

Performance probably is not an issue

enumerate() can be asynchronous

Usage Study: `exits()`, `fillHandlerInfo()`, `getTypeFromExtension()`

[nsExternalHelperAppService.cpp](#)

These APIs are used in [page loading flow](#).

Docshell needs to find the proper content handler for handling the content.

Searching `mimeTypes.rdf` is one step of finding algorithm and it can't go to next step before checking the searching result.

These APIs can not be asynchronous.

Synchronous to Asynchronous

API	Asynchronous
store()	O
remove()	O (No consumer for this API)
enumerate()	O
exits()	X
fillHandlerInfo()	X
getTypeFromExtension()	X

nsIHandlerService Improvement conclusion

- Keep API being synchronous
- Make JSON store implementation with
 - Asynchronous write
 - synchronous read