

Development Tools

Magnolia Compatibility	
5.6	

The Development Tools module provides a basis for creating plugins to interact with a running Magnolia instance. Using the endpoints provided by this module you can create interactive IDEs with intellisense features to help facilitate the development process.

- [Installing](#)
- [Usage](#)
 - [Registry Endpoint](#)
 - [Repository Endpoint](#)
- [Source Code](#)
- [Issues, Bugs, Improvements](#)

Installing

Maven is the easiest way to install the module. Add the following dependency to your [bundle](#):

```
<dependency>
  <groupId>info.magnolia.development</groupId>
  <artifactId>magnolia-development-tools</artifactId>
  <version>1.0-SNAPSHOT</version>
</dependency>
```

Usage


This module comes with 2 REST endpoints. One for querying the different definition registries and another for creating workspaces.

Registry Endpoint

Returns registry data in the form of a JSON object.

Request URL

```
/.rest/registry/v1/{registry}/{module}
```

Parameter	Description	Parameter type	Data type
registry	required The name of the registry to access. Get all items from a specific registry. Example: <code>modules</code> , <code>apps</code> , <code>dialogs</code> , <code>fields</code> , <code>templates</code>  Get a list of problems from all definitions by using <code>problems</code> as the <code>registry</code> path parameter.	path	String
module	optional Used together with a <code>modules</code> registry request. Limit the request to target a specific module and return a list of registries used by that module.	path	String
type	optional Used together with a <code>modules</code> registry request. Find all the definitions of a specific type. Each definition will include the relative path to the definition if one exists.	query	String

onlyDefinitionProviders	optional, default is <i>false</i> Used together with a <code>modules</code> registry request. Set this flag if you only want a list of modules which provide definitions .	query	Boolean
-------------------------	--	-------	---------

Example: Get all modules

Get all the modules from the `ModuleRegistry`.

```
curl -X GET 'http://localhost:8080/magnoliaAuthor/.rest/registry/v1/modules' \  
-u superuser:superuser
```

```
[
  "about-app",
  "appswitcher",
  "cache",
  "cache-app",
  "cache-browser-app",
  "categorization",
  "categorization-support",
  "ce-product-descriptor",
  "config",
  "contacts",
  "core",
  "dam",
  "dam-app",
  "dam-imaging",
  "dam-jcr",
  "dam-templating",
  "definitions-app",
  "development-tools",
  "ehcache3",
  "form",
  "freemarker-support",
  "google-sitemap",
  "groovy",
  "il8n",
  "imaging",
  "imaging-support",
  "javascript-models",
  "jcr-browser-app",
  "jcr-tools",
  "log-tools",
  "magnolia-resources-templating",
  "mail",
  "messages-app",
  "mte",
  "mte-imaging",
  "mtk",
  "pages",
  "password-manager",
  "publishing-app",
  "publishing-core",
  "publishing-receiver",
  "publishing-sender",
  "rendering",
  "resource-loader",
  "resources",
  "resources-app",
  "rest-client",
  "rest-content-delivery",
  "rest-integration",
  "rest-services",
  "resteasy-client",
  "rssaggregator",
  "scheduler",
  "security-app",
  "site",
  "site-app",
  "task-management",
  "templating",
  "templating-jsp",
  "ui-admincentral",
  "ui-contentapp",
  "ui-framework",
  "ui-mediaeditor",
  "virtual-uri",
  "webapp"
]
```

Example: Get a list of definition registries used by a module

Query a specific module for the definition types it provides.

```
curl -X GET 'http://localhost:8080/magnoliaAuthor/.rest/registry/v1/modules/ui-framework' \  
-u superuser:superuser
```

```
[  
  "dialogs",  
  "fieldTypes"  
]
```

Example: Get a list of definitions for a specific type

Get a list of all definitions by type. This will include the relative path if one exists.

```
curl -X GET 'http://localhost:8080/magnoliaAuthor/.rest/registry/v1/modules/ui-framework?type=dialogs' \  
-u superuser:superuser
```

```
[  
  "folder",  
  "importZip",  
  "rename"  
]
```

Example: Get all modules providing definitions

Get all the modules which provide definitions.

```
curl -X GET 'http://localhost:8080/magnoliaAuthor/.rest/registry/v1/modules?onlyDefinitionProviders=true' \  
-u superuser:superuser
```

```
[
  "about-app",
  "cache-app",
  "cache-browser-app",
  "categorization",
  "contacts",
  "dam-app",
  "definitions-app",
  "development-tools",
  "eclipse-plugin-support",
  "form",
  "google-sitemap",
  "groovy",
  "jcr-browser-app",
  "jcr-tools",
  "log-tools",
  "mail",
  "messages-app",
  "mte",
  "mtk",
  "pages",
  "password-manager",
  "publishing-app",
  "rendering",
  "resources",
  "resources-app",
  "rest-services",
  "rssaggregator",
  "scheduler",
  "security-app",
  "site",
  "site-app",
  "templating",
  "ui-admincentral",
  "ui-framework",
  "ui-mediaeditor"
]
```

Repository Endpoint

Create the workspace and node type. The access permissions on the workspace are created for the superuser.

Request URL

```
/.rest/repository/v1/workspace/{name}
```

Parameter	Description	Parameter type	Data type
name	required The name of the workspace to be created.	path	String

Example: Create a workspace

Create a workspace named events with a node type name event.



If the workspace name is plural (i.e. ending in 's') then the node type will be created singular. Otherwise the node type will have the same name as the workspace.

```
curl -X GET 'http://localhost:8080/magnoliaAuthor/.rest/repository/v1/workspace/events' \
-u superuser:superuser
```

```
{ Workspace events created }
```

Source Code

The source for this module can be found here: <https://github.com/magnolia-community/development-tools>

Issues, Bugs, Improvements

Report them here: <https://github.com/magnolia-community/development-tools/issues>