

Module QuickStart

This page provides a quick overview of module development for experienced Magnolia developers. An overview of Magnolia modules can be found at in our documentation on the [Modules](#) page.

- [Using a Maven Archetype](#)
 - [Module archetype](#)
 - [Forge Module archetype](#)
 - [STK Theme archetype](#)
 - [Project archetype](#)
 - [Changing a project from CE to EE](#)
- [Setting the File Encoding](#)
- [Contribute](#)
- [Host your module at the Forge!](#)

Using a Maven Archetype

You can use Maven 2's archetype plugin to quickly get started with creating a module or project.

Use this command:

```
mvn org.apache.maven.plugins:maven-archetype-plugin:2.4:generate -DarchetypeCatalog=https://nexus.magnolia-cms.com/content/groups/public/
```

You will first be prompted to "Choose an archetype".

You will further be prompted for various properties, whose names should hopefully be obvious enough for you to figure what to enter. If not, try some random value, see the results and try again 🤖 (Please vote for [this maven-archetype-plugin enhancement](#) !)

Module archetype

The module archetype will generate a basic Magnolia module, including all necessary folders, a module descriptor and two classes:

- a module configuration class, named after the `module-class-name` property. See http://documentation.magnolia-cms.com/technical-guide/modules.html#Module_class for details.
- a module version handler class, similarly named, in the `.setup` subpackage. See http://documentation.magnolia-cms.com/technical-guide/modules.html#Version_handling for details.

Forge Module archetype

The Forge Module archetype is deprecated.

Please follow these improved tips to create a Forge module: [Magnolia Forge#Mavenbuild](#)

STK Theme archetype

The STK Theme archetype will produce a quickstart for an [STK Theme](#) (see also [the part on themes in STK changes in 5.4](#)).

Project archetype

 See the [official documentation on how to setup a Maven based Magnolia webapp](#).

[Tutorial for a community project](#).

[Tutorial for a DX Core enterprise project](#) (requires a licence.)

The Project archetype will produce a skeleton for a "project"; this will contain

- a parent pom
- a webapp, overlaying `magnolia-empty-webapp`, ready to host your custom configuration files.

If you want to add a module to an existing project, generating the `module` artifact from within your project should get you started quickly, as it will adapt the new artifact's parent pom to match your existing project !

Changing a project from CE to EE

When you create a project using an archetype from the `public` group you get a Community Edition (CE) project. This means the project has dependencies to the CE bundle and you get only community modules. If you need a more substantial starting point, change the dependencies in your POM files to the Enterprise Edition (EE) bundle and webapp. This way you get enterprise modules such as Multisite and Workflow.

There are two videos in [Magnolia Academy](#) that explain how to work with Module Dependencies.

In your **project** POM, set:

- **groupId**: `info.magnolia.eebundle`
- **artifactId**: `magnolia-enterprise-bundle-parent`

Change this:

CE dependencies in project POM

```
<dependency>
  <groupId>info.magnolia.bundle</groupId>
  <artifactId>magnolia-bundle-parent</artifactId>
  <version>${magnoliaVersion}</version>
  <type>pom</type>
  <scope>import</scope>
</dependency>
```

To this:

EE dependencies in project POM

```
<dependency>
  <groupId>info.magnolia.eebundle</groupId>
  <artifactId>magnolia-enterprise-bundle-parent</artifactId>
  <version>${magnoliaVersion}</version>
  <type>pom</type>
  <scope>import</scope>
</dependency>
```

In your **webapp** POM, set:

- **groupId**: `info.magnolia.eebundle`
- **artifactId**: `magnolia-enterprise-webapp`

Change this:

CE dependencies in webapp POM

```
<dependency>
  <groupId>info.magnolia</groupId>
  <artifactId>magnolia-empty-webapp</artifactId>
  <type>pom</type>
</dependency>
<dependency>
  <groupId>info.magnolia</groupId>
  <artifactId>magnolia-empty-webapp</artifactId>
  <type>war</type>
</dependency>
```

To this:

EE dependencies in webapp POM

```
<dependency>
  <groupId>info.magnolia.eebundle</groupId>
  <artifactId>magnolia-enterprise-webapp</artifactId>
  <type>pom</type>
</dependency>
<dependency>
  <groupId>info.magnolia.eebundle</groupId>
  <artifactId>magnolia-enterprise-webapp</artifactId>
  <type>war</type>
</dependency>
```

Setting the File Encoding

If you see a warning about file encoding:

```
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform
dependent!
```

See <https://maven.apache.org/general.html#encoding-warning> for instructions.

Contribute

Have a look at <http://jira.magnolia-cms.com/browse/ARCH> to see known problems or suggest new features.

Please try it out and tell us what you think !

More info about Maven archetypes:

- <http://maven.apache.org/guides/introduction/introduction-to-archetypes.html>
- <http://maven.apache.org/guides/mini/guide-creating-archetypes.html>

Host your module at the Forge!

Hey, why not share your new module with the world and host it at the [forge](#)?



✦

Your Rating:  Results:  197 rates