

Setting up a Magnolia website project

New developers can start here. Step-by-step on how to create a new project and import it into Eclipse IDE.

- [Prerequisites](#)
- [Create the project](#)
- [Bundles](#)
 - [Webapps](#)
- [Build the project](#)
- [Import the project](#)
 - [Import existing maven project](#)
 - [Server configuration](#)
 - [VM Arguments](#)

Prerequisites

- Java 8, 9 JDK Installed
- Maven Installed
 - Suggested Maven 3.3+
 - settings.xml created (See [Maven setup](#)).

Create the project

New Magnolia projects start with the archetype.

From the command line run:

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

Choose archetype:

- 1: An archetype to create STK Theme modules
- 2: **An archetype to create a Magnolia project (a parent pom and a webapp)**
- 3: An archetype to create basic Magnolia modules
- 4: An archetype to create a Magnolia module to be hosted on the Magnolia Forge
- 5: An archetype to create Magnolia modules using Blossom

Each archetype can have it's own set of versions.

Choose info.magnolia.maven.archetypes:magnolia-project-archetype version:

- 1: 1.1.0
- 2: 1.2.0
- 3: 1.2.1
- 4: 1.2.2
- 5: 1.2.3
- 6: 1.2.4
- 7: **1.2.5-SNAPSHOT**

Define the project:

<code>groupId</code>	com.mycompany.webapps
<code>artifactId</code>	magnolia-cms
<code>version</code>	5.7-SNAPSHOT
<code>package</code>	com.mycompany.webapps (press enter key to accept)
<code>magnolia-bundle-version</code>	5.7
<code>project-name</code>	magnolia-cms

Confirm success:

```
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: magnolia-project-archetype:1.2.4-SNAPSHOT
[INFO] -----
[INFO] Parameter: groupId, Value: com.mycompany.webapps
[INFO] Parameter: artifactId, Value: magnolia-cms
[INFO] Parameter: version, Value: 5.6-SNAPSHOT
[INFO] Parameter: package, Value: com.mycompany.webapps
[INFO] Parameter: packageInPathFormat, Value: com/mycompany/webapps
[INFO] Parameter: magnolia-bundle-version, Value: 5.6
[INFO] Parameter: package, Value: com.mycompany.webapps
[INFO] Parameter: version, Value: 5.6-SNAPSHOT
[INFO] Parameter: groupId, Value: com.mycompany.webapps
[INFO] Parameter: project-name, Value: magnolia-cms
[INFO] Parameter: artifactId, Value: magnolia-cms
[INFO] Parent element not overwritten in /Users/rgange/Temp/magnolia-cms/magnolia-cms-webapp/pom.xml
[INFO] project created from Archetype in dir: /Users/rgange/Temp/magnolia-cms
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 09:28 min
[INFO] Finished at: 2018-04-23T14:37:15+02:00
[INFO] Final Memory: 15M/170M
[INFO] -----
```

Bundles

Choose the bundle that you would like to use.

- Community Edition Bundle
- Enterprise Edition Bundle



Be aware of the properties `magnoliaVersion` and `magnoliaBundleVersion`. The `magnoliaVersion` refers to the [main project version](#), which is a collection of modules including the core. The `magnoliaBundleVersion` ties together a project with other resources. These numbers can diverge since a bundle release does not necessarily mean a new project release.

parent pom

```
<properties>
  <magnoliaBundleVersion>5.7</magnoliaBundleVersion>
  <javaVersion>1.9</javaVersion>
</properties>

<!-- Option A -->
<!-- Importing dependencyManagement of CE bundle. -->
<!--
<dependency>
  <groupId>info.magnolia.bundle</groupId>
  <artifactId>magnolia-bundle-parent</artifactId>
  <version>${magnoliaBundleVersion}</version>
  <type>pom</type>
  <scope>import</scope>
</dependency>
-->

<!-- Option B -->
<!-- If you want to use the Enterprise Edition. -->
<dependency>
  <groupId>info.magnolia.eebundle</groupId>
  <artifactId>magnolia-enterprise-bundle-parent</artifactId>
  <version>${magnoliaBundleVersion}</version>
  <type>pom</type>
  <scope>import</scope>
</dependency>
```

Webapps

Choose the webapp that you would like to use.

- Empty
- Community
- Enterprise Std
- Enterprise Pro

webapp pom

```
<!-- option i - magnolia-empty-webapp -->
<!-- Dependencies versions are already imported by parent pom. Requires "Option A" in the parent 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>
-->

<!-- option ii - magnolia-community-webapp -->
<!-- Dependencies versions are already imported by parent pom. Requires "Option A" in the parent pom. -->
<!--
<dependency>
  <groupId>info.magnolia.bundle</groupId>
  <artifactId>magnolia-community-webapp</artifactId>
  <type>pom</type>
</dependency>
<dependency>
  <groupId>info.magnolia.bundle</groupId>
  <artifactId>magnolia-community-webapp</artifactId>
  <type>war</type>
</dependency>
-->

<!-- option iii - magnolia-enterprise-standard-webapp -->
<!-- Dependencies versions are already imported by parent pom. Requires "Option B" in the parent pom. -->
<dependency>
  <groupId>info.magnolia.eebundle</groupId>
  <artifactId>magnolia-enterprise-standard-webapp</artifactId>
  <type>pom</type>
</dependency>
<dependency>
  <groupId>info.magnolia.eebundle</groupId>
  <artifactId>magnolia-enterprise-standard-webapp</artifactId>
  <type>war</type>
</dependency>

<!-- option iv - magnolia-enterprise-pro-webapp -->
<!-- Dependencies versions are already imported by parent pom. Requires "Option B" in the parent pom. -->
<!--
<dependency>
  <groupId>info.magnolia.eebundle</groupId>
  <artifactId>magnolia-enterprise-pro-webapp</artifactId>
  <type>pom</type>
</dependency>
<dependency>
  <groupId>info.magnolia.eebundle</groupId>
  <artifactId>magnolia-enterprise-pro-webapp</artifactId>
  <type>war</type>
</dependency>
-->
```

Build the project

From the command line use maven to build the project. Execute the clean install from within the project directory.

```
mvn clean install
```

Eclipse users also need to explode the war file.

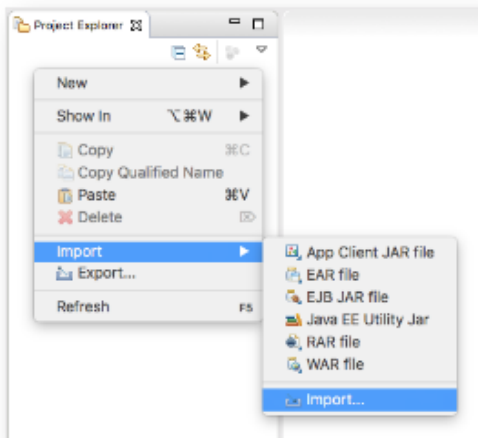
```
mvn war:inplace
```

Import the project

Import the project into your IDE. Here is an example using Eclipse Oxygen.

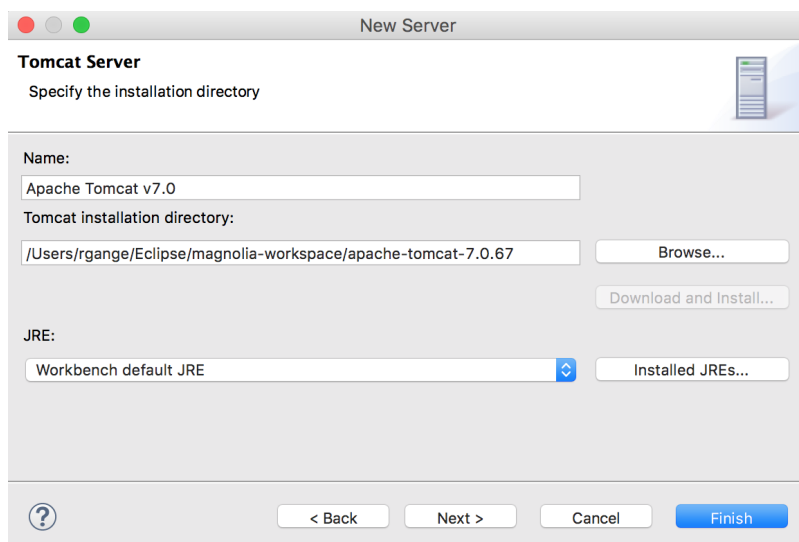
Import existing maven project

The archetype has created the poms needed for the project. Maven has built the project using those poms. Now import the project into the IDE as an existing maven project.



Server configuration

Using [Tomcat 7](#) for development is recommended so that we can use the feature [Serve modules without publishing](#).



VM Arguments

We recommend: `-Xmx1024M`

