

Struts 2 Maven Archetypes

DEPRECATED: moved to <http://struts.apache.org/maven-archetypes/>

Struts 2 provides several [Maven](#) archetypes that create a starting point for our own applications.

Contents

- 1 [DEPRECATED: moved to http://struts.apache.org/maven-archetypes/](#)
- 2 [Quickstart](#)
- 3 [Available Archetypes](#)
 - 3.1 [The Blank Convention Archetype \(struts2-archetype-convention\)](#)
 - 3.2 [The Blank Archetype \(struts2-archetype-blank\)](#)
 - 3.3 [The Starter Archetype \(struts2-archetype-starter\)](#)
 - 3.4 [The AngularJS Archetype \(struts2-archetype-angularjs\)](#)
 - 3.5 [The Portlet Blank Archetype \(struts2-archetype-portlet\)](#)
 - 3.6 [The Portlet Database Archetype \(struts2-archetype-dbportlet\)](#)
 - 3.7 [The Plugin Archetype \(struts2-archetype-plugin\)](#)
- 4 [Creating an Application Using a Maven Archetype](#)
 - 4.1 [Staging repository](#)
 - 4.2 [Archetype Parameters](#)
- 5 [Command Quick Reference](#)

Quickstart

A recommended way to start with Struts2 archetypes is to work with the archetype catalog.

```
mvn archetype:generate -DarchetypeCatalog=http://struts.apache.org/
```

Available Archetypes

The Blank Convention Archetype (struts2-archetype-convention)

Features

- Convention-based validation
- Example actions
- Package-level resource bundle
- Unit-testing
- Google AppEngine aware

```
mvn archetype:generate -B -DgroupId=com.mycompany.mysystem \
                        -DartifactId=myWebApp \
                        -DarchetypeGroupId=org.apache.struts \
                        -DarchetypeArtifactId=struts2-archetype-convention \
                        -DarchetypeVersion=<CURRENT_STRUTS_VERSION> \
                        -DremoteRepositories=http://struts.apache.org
```

The Blank Archetype (struts2-archetype-blank)

The Struts 2 Blank Archetype ("blank-archetype") provides a minimal, but complete, Struts 2 application. It demonstrates some of the most basic Struts 2 concepts.

Features

- XML-based configuration, demonstrates including additional config file
- Example actions
- Package-level resource bundle
- XML-based validation
- Unit-testing

See the [Struts 2 Blank Archetype](#) page for more information, including some basic Maven information, how to build and run the app, and basic application information.

```
mvn archetype:generate -B -DgroupId=com.mycompany.mysystem \
    -DartifactId=myWebApp \
    -DarchetypeGroupId=org.apache.struts \
    -DarchetypeArtifactId=struts2-archetype-blank \
    -DarchetypeVersion=<CURRENT_STRUTS_VERSION> \
    -DremoteRepositories=http://struts.apache.org
```

The Starter Archetype (struts2-archetype-starter)

The Starter archetype creates a more featured application using several common technologies used in production applications.

Features

- Sitemesh integration
- Action example (instantiated both through Spring and Struts)
- Spring integration
- Validation example (action and action-alias level)
- Conversion example (global and action level)
- Resource bundle (both global, action and package level)

```
mvn archetype:generate -B -DgroupId=com.mycompany.mysystem \
    -DartifactId=myWebApp \
    -DarchetypeGroupId=org.apache.struts \
    -DarchetypeArtifactId=struts2-archetype-starter \
    -DarchetypeVersion=<CURRENT_STRUTS_VERSION> \
    -DremoteRepositories=http://struts.apache.org
```

The AngularJS Archetype (struts2-archetype-angularjs)

Features

- Convention-based Action Configuration
- Example actions with JSON Result
- Example for AngularJS and Struts2 Integration

```
mvn archetype:generate -B -DgroupId=com.mycompany.mysystem \
    -DartifactId=myWebApp \
    -DarchetypeGroupId=org.apache.struts \
    -DarchetypeArtifactId=struts2-archetype-angularjs \
    -DarchetypeVersion=<CURRENT_STRUTS_VERSION> \
    -DremoteRepositories=http://struts.apache.org
```

The Portlet Blank Archetype (struts2-archetype-portlet)

The Portlet blank archetype creates a minimally populated JSR 168 portlet.

Features

- View, Edit, and Help mode examples
- Simple form for preferences in Edit mode
- Can be deployed as a servlet or portlet application
- Can use Maven Jetty plugin to deploy with the pluto-embedded profile (usage 'mvn jetty:run -Ppluto-embedded', then access <http://localhost:8080/<artifactId>/pluto/index.jsp>)

```
mvn archetype:generate -B -DgroupId=com.mycompany.mysystem \
    -DartifactId=myWebApp \
    -DarchetypeGroupId=org.apache.struts \
    -DarchetypeArtifactId=struts2-archetype-portlet \
    -DarchetypeVersion=<CURRENT_STRUTS_VERSION> \
    -DremoteRepositories=http://struts.apache.org
```

The Portlet Database Archetype (struts2-archetype-dbportlet)

The Portlet database archetype creates a simple JSR 168 portlet that displays the contents of a database table.

Features

- Uses Spring and Hsql to show a real database query
- Builtin caching of query results
- View, Edit, and Help mode examples
- Simple form for preferences in Edit mode
- Can be deployed as a servlet or portlet application
- Can use Maven Jetty plugin to deploy as a servlet webapp

```
mvn archetype:generate -B -DgroupId=com.mycompany.mysystem \
    -DartifactId=myWebApp \
    -DarchetypeGroupId=org.apache.struts \
    -DarchetypeArtifactId=struts2-archetype-dbportlet \
    -DarchetypeVersion=<CURRENT_STRUTS_VERSION> \
    -DremoteRepositories=http://struts.apache.org
```

The Plugin Archetype (struts2-archetype-plugin)

Features

- Example new result type
- Example XML-based configuration

```
mvn archetype:generate -B -DgroupId=com.mycompany.mysystem \
    -DartifactId=myPlugin \
    -DarchetypeGroupId=org.apache.struts \
    -DarchetypeArtifactId=struts2-archetype-plugin \
    -DarchetypeVersion=<CURRENT_STRUTS_VERSION> \
    -DremoteRepositories=http://struts.apache.org
```

Creating an Application Using a Maven Archetype

Run the following command from your new project's parent directory. The project will be created in a sub-directory named after the `artifactId` parameter.

The first two parameters, `groupId` and `artifactId`, should be set according to your needs:

- `groupId`
This should be the root package of your application, e.g., `com.mycompany.myapp`.
- `artifactId`
Names the project directory that will be created in the current directory.

```
mvn archetype:generate -B \
    -DgroupId=tutorial \
    -DartifactId=tutorial \
    -DarchetypeGroupId=org.apache.struts \
    -DarchetypeArtifactId=struts2-archetype-blank \
    -DarchetypeVersion=<CURRENT_STRUTS_VERSION> \
    -DremoteRepositories=http://struts.apache.org
```

where `<CURRENT_STRUTS_VERSION>` is archetype's version you want to use, eg. 2.3.15.1

Staging repository

If the above command will fail because of missing archetypes in central repository, you can try to use staging repository like below

```
mvn archetype:generate -DarchetypeCatalog=https://repository.apache.org/content/repositories/snapshots/
```

or this

```
mvn archetype:generate -DarchetypeCatalog=http://struts.apache.org/
```

Archetype Parameters

The `-B` option runs `archetype:generate` in batch mode; it avoids a prompt to confirm our settings.

Parameter	Description
<code>groupId</code>	The id for the group the application belongs to. Usually is the root package for applications in your company, e.g. <code>com.mycompany</code>
<code>artifactId</code>	The id for the project. The project will be created a sub-directory named after this parameter.
<code>archetypeGroupId</code>	The group id of the archetype. Will always be <code>org.apache.struts</code> for Struts archetypes
<code>archetypeArtifactId</code>	The id of the archetype
<code>archetypeVersion</code>	The version of the archetype
<code>package</code>	(Optional) The base Java package to use for generated source code. Defaults to <code>archetypeGroupId</code> if not specified.
<code>remoteRepositories</code>	(Optional) A list of remote repositories that contain the archetype.

Command Quick Reference

These commands are used from the directory created by the archetype plugin (the project's sub-directory that was created in the previous step).

- To build

```
mvn install
```

- To create IntelliJ IDEA project files

```
mvn idea:idea
```

- To create Eclipse project files

```
mvn eclipse:eclipse -Dwtpversion=1.5
```

- To run test cases

```
mvn test
```

- To clean up (removes generated artifacts)

```
mvn clean
```

- To package (creates a WAR file)

```
mvn package
```

- To grab necessary JAR files etc.

```
mvn initialize
```

- To run it with Jetty

```
mvn jetty:run
```