Developing Plugins Using Maven

You can easily develop TeamCity plugins with Maven.

On this page:

Supported Maven versions

Both Maven 2 (2.2.1+) and Maven 3 (3.0.4+) are supported.

Open API in Maven Repository

TeamCity Open API is available as a set of Maven artifacts residing in the JetBrains Maven repository (http://download.jetbrains.com/teamcity-repository). Add this fragment to the `<repositories>` section of your pom file to access it:

```
<repository>
  <id>jetbrains-all</id>
  <url>http://download.jetbrains.com/teamcity-repository</url>
</repository>
```

Please note that only open API artifacts are present in the repository. If your plugin needs to use the not-open API, the corresponding jars should then be added to the project from the TeamCity distribution as they are not provided in the repository.

The open API in the repository is split into two main parts:

The server-side API:

```
<dependency>
  <groupId>org.jetbrains.teamcity</groupId>
  <artifactId>server-api</artifactId>
  <version>10.0</version>
  <scope>provided</scope>
</dependency>
```

The agent-side API:

```
<dependency>
  <groupId>org.jetbrains.teamcity</groupId>
  <artifactId>agent-api</artifactId>
  <version>10.0</version>
  <scope>provided</scope>
</dependency>
```

ⓘ   Note that API dependencies are used with the `provided` scope. This way you will avoid adding the API and its transitive dependencies to the target distribution.

There is also an artifact to support plugin tests:

```
<dependency>
  <groupId>org.jetbrains.teamcity</groupId>
  <artifactId>tests-support</artifactId>
  <version>10.0</version>
  <scope>test</scope>
</dependency>
```

Maven Archetypes

For a quick start with a plugin, there are three Maven archetypes in the `org.jetbrains.teamcity.archetypes` group:

- `teamcity-plugin` - an empty plugin, includes both the server and the agent plugin parts
- `teamcity-server-plugin` - an empty plugin, includes the server plugin part only
- `teamcity-sample-plugin` - the plugin with the sample code (adds a "Click me" button to the bottom of the TeamCity project Overview page)

Different released versions of the TeamCity server API are listed here.

Here is the Maven command that will generate a project for a server-side-only plugin depending on 10.0 TeamCity version:

```
mvn org.apache.maven.plugins:maven-archetype-plugin:2.4:generate
-DarchetypeRepository=http://download.jetbrains.com/teamcity-repository
-DarchetypeArtifactId=teamcity-server-plugin -DarchetypeGroupId=org.jetbrains.teamcity.archetypes
-DarchetypeVersion=RELEASE -DteamcityVersion=10.0
```

Here is the Maven command that will generate a project that contains both, the server and agent parts of a plugin and depends on 10.0 TeamCity version:

```
mvn org.apache.maven.plugins:maven-archetype-plugin:2.4:generate
-DarchetypeRepository=http://download.jetbrains.com/teamcity-repository
-DarchetypeArtifactId=teamcity-plugin -DarchetypeGroupId=org.jetbrains.teamcity.archetypes
-DarchetypeVersion=RELEASE -DteamcityVersion=10.0
```

Here is the Maven command that will generate a sample project on 10.0 TeamCity version:

```
mvn org.apache.maven.plugins:maven-archetype-plugin:2.4:generate
-DarchetypeRepository=http://download.jetbrains.com/teamcity-repository
-DarchetypeArtifactId=teamcity-sample-plugin -DarchetypeGroupId=org.jetbrains.teamcity.archetypes
-DarchetypeVersion=RELEASE -DteamcityVersion=10.0
```

You will be asked to enter the usual Maven `groupId`, `artifactId` and `version` for your plugin. Please note, that artifactId will be used as your plugin (internal) name.
After the project is generated, you may want to update `teamcity-plugin.xml` in the root directory: enter display name, description, author e-mail and other information.

Finally, change the directory to the root of the generated project and run

```
mvn package
```

The `target` directory of the project root will contain the `<artifactId>.zip` file. It is your plugin package. You can install it to TeamCity or use the TeamClity SDK Maven plugin.

TeamCity SDK Maven plugin

You can also use the TeamCity SDK Maven plugin allowing you to control a TeamCity instance from the command line and to install a new/updated plugin created from a Maven archetype