# ~~To create a new GWT (Google Web Toolkit) app from nothing~~

~~mvn archetype:generate -DarchetypeGroupId=com.totsp.gwt -DarchetypeArtifactId=maven-googlewebtoolkit2-archetype -DarchetypeVersion=1.0.3 -DremoteRepositories=http://gwt-maven.googlecode.com/svn/trunk/mavenrepo -DartifactId=myGwtProject -DgroupId=com.myco~~

~~Where:~~

* **~~-DartifactId=myGwtProject~~** ~~= name of the project. Creates a sub-directory underneath the folder where the command is run.~~
* **~~-DgroupId=com.myco~~** ~~= package name. Creates the client package under com/myco.~~
* **~~-DremoteRepositories=http://gwt-maven.googlecode.com/svn/trunk/mavenrepo~~** ~~= remote maven repository~~

# Creating a bare bones GWT application with codehaus mojo gwt-maven-plugin

The gwt-maven-plugin Archetype creates a bare bones GWT application with all the gwt-maven-plugin setup and structure in place (a template).

## Using the Archetype

Use it as you would any other Maven archetype to create a template/stub project.0

mvn archetype:generate -DarchetypeGroupId=org.codehaus.mojo -DarchetypeArtifactId=gwt-maven-plugin -DarchetypeVersion=1.1 -DgroupId=com.coolisland -DartifactId=gwtSetup

Where:

* **-DartifactId=** gwtSetup -> name of the project. Creates a sub-directory with the *artifactId* underneath the folder where the command is run.
* **-DgroupId=** com.coolisland = package name. Creates the client package under com/coolisland.

Maven will ask the following questions. Just hit enter:

**Define value for version:** *1.0-SNAPSHOT: :*

**Confirm properties configuration:**

groupId: com.coolisland

artifactId: gwtSetup

version: 1.0-SNAPSHOT

package: com.coolisland

*Y: :*

# Building and Running GWT application with Maven’s codehaus mojo gwt-maven-plugin

call mvn -e clean gwt:clean gwt:compile gwt:eclipse gwt:run

This command will build the project and launch a server with the application.

# Codehaus mojo plugin for GWT

| **Goal** | **Report?** | **Description** |
| --- | --- | --- |
| [gwt:clean](http://mojo.codehaus.org/gwt-maven-plugin/clean-mojo.html) | No | Cleanup the webapp directory for GWT module compilation output |
| [gwt:compile](http://mojo.codehaus.org/gwt-maven-plugin/compile-mojo.html) | No | Invokes the GWTCompiler for the project source. |
| [gwt:debug](http://mojo.codehaus.org/gwt-maven-plugin/debug-mojo.html) | No | Extends the gwt goal and runs the project in the GWT Hosted mode with a debugger port hook (optionally suspended). |
| [gwt:eclipse](http://mojo.codehaus.org/gwt-maven-plugin/eclipse-mojo.html) | No | Goal which creates Eclipse lauch configurations for GWT modules. |
| [gwt:eclipseTest](http://mojo.codehaus.org/gwt-maven-plugin/eclipseTest-mojo.html) | No | **Deprecated.** use google eclipse plugin http://code.google.com/intl/fr-FR/eclipse/docs/users\_guide.html |
| [gwt:generateAsync](http://mojo.codehaus.org/gwt-maven-plugin/generateAsync-mojo.html) | No | Goal which generate Asyn interface. |
| [gwt:help](http://mojo.codehaus.org/gwt-maven-plugin/help-mojo.html) | No | Display help information on gwt-maven-plugin. Call  mvn gwt:help -Ddetail=true -Dgoal=<goal-name>  to display parameter details. |
| [gwt:i18n](http://mojo.codehaus.org/gwt-maven-plugin/i18n-mojo.html) | No | Creates I18N interfaces for constants and messages files. |
| [gwt:mergewebxml](http://mojo.codehaus.org/gwt-maven-plugin/mergewebxml-mojo.html) | No | Merges GWT servlet elements into deployment descriptor (and non GWT servlets into shell). |
| [gwt:resources](http://mojo.codehaus.org/gwt-maven-plugin/resources-mojo.html) | No | Copy GWT java source code and module descriptor as resources in the build outputDirectory. Alternative to declaring a <resource> in the POM with finer filtering as the module descriptor is read to detect sources to be copied. |
| [gwt:run](http://mojo.codehaus.org/gwt-maven-plugin/run-mojo.html) | No | Goal which run a GWT module in the GWT Hosted mode. |
| [gwt:sdkInstall](http://mojo.codehaus.org/gwt-maven-plugin/sdkInstall-mojo.html) | No | Install a GWT (home built) SDK in local repository |
| [gwt:soyc](http://mojo.codehaus.org/gwt-maven-plugin/soyc-mojo.html) | Yes | No description. |
| [gwt:test](http://mojo.codehaus.org/gwt-maven-plugin/test-mojo.html) | No | Mimic surefire to run GWTTestCases during integration-test phase, until SUREFIRE-508 is fixed |

# POM for downloading GWT repositories

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

<!--

POM generated by gwt-maven-plugin archetype

-->

<modelVersion>4.0.0</modelVersion>

<groupId>com.coolisland.gwt.maven.demo</groupId>

<artifactId>demo</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<properties>

<!-- convenience to define GWT version in one place -->

<gwt.version>2.0.0</gwt.version>

<!-- tell the compiler we can use 1.5 -->

<maven.compiler.source>1.5</maven.compiler.source>

<maven.compiler.target>1.5</maven.compiler.target>

</properties>

<dependencies>

<!-- GWT dependencies (from central repo) -->

<dependency>

<groupId>com.google.gwt</groupId>

<artifactId>gwt-servlet</artifactId>

<version>${gwt.version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>com.google.gwt</groupId>

<artifactId>gwt-user</artifactId>

<version>${gwt.version}</version>

<scope>provided</scope>

</dependency>

<!-- test -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.7</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<outputDirectory>war/WEB-INF/classes</outputDirectory>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>gwt-maven-plugin</artifactId>

<version>1.2</version>

<executions>

<execution>

<goals>

<goal>compile</goal>

<goal>generateAsync</goal>

<goal>test</goal>

</goals>

</execution>

</executions>

<configuration>

<runTarget>com.coolislan.gwt.maven.demo.gwt.maven.demo.Application/Application.html</runTarget>

</configuration>

</plugin>

<!--

If you want to use the target/web.xml file mergewebxml produces,

tell the war plugin to use it.

Also, exclude what you want from the final artifact here.

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-war-plugin</artifactId>

<configuration>

<webXml>target/web.xml</webXml>

<warSourceExcludes>.gwt-tmp/\*\*</warSourceExcludes>

</configuration>

</plugin>

-->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>2.0.2</version>

<configuration>

<source>${maven.compiler.source}</source>

<target>${maven.compiler.target}</target>

</configuration>

</plugin>

<!-- added the maven-war-plugin artifact according to http://stackoverflow.com/questions/1894003/maven-gwt-2-0-and-eclipse -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-war-plugin</artifactId>

<version>2.0.2</version>

<configuration>

<warSourceDirectory>war</warSourceDirectory>

<webXml>src/main/webapp/WEB-INF/web.xml</webXml>

</configuration>

</plugin>

</plugins>

</build>

</project>