

Creating a deployment package using Eclipse

This section shows how to package your Java code into a deployment package using Eclipse IDE and Maven plugin for Eclipse.

Prerequisites

Install the **Maven** Plugin for Eclipse.

1. Start Eclipse. From the **Help** menu in Eclipse, choose **Install New Software**.
 2. In the **Install** window, type <http://download.eclipse.org/technology/m2e/releases> in the **Work with:** box, and choose **Add**.
 3. Follow the steps to complete the setup.
-

Create and build a project

In this step, you start Eclipse and create a Maven project. You will add the necessary dependencies, and build the project. The build will produce a .jar, which is your deployment package.

1. Create a new Maven project in Eclipse.
 - a. From the **File** menu, choose **New**, and then choose **Project**.
 - b. In the **New Project** window, choose **Maven Project**.
 - c. In the **New Maven Project** window, choose **Create a simple project**, and leave other default selections.
 - d. In the **New Maven Project, Configure project** windows, type the following **Artifact** information:
 - **Group Id:** doc-examples
 - **Artifact Id:** lambda-java-example
 - **Version:** 0.0.1-SNAPSHOT
 - **Packaging:** jar
 - **Name:** lambda-java-example
2. Add the aws-lambda-java-core dependency to the pom.xml file.

It provides definitions of the RequestHandler, RequestStreamHandler, and Context interfaces. This allows you to compile code that you can use with AWS Lambda.

- a. Open the context (right-click) menu for the pom.xml file, choose **Maven**, and then choose **Add Dependency**.
- b. In the **Add Dependency** windows, type the following values:

Group Id: com.amazonaws

Artifact Id: aws-lambda-java-core

Version: 1.2.1

Note

If you are following other tutorial topics in this guide, the specific tutorials might require you to add more dependencies. Make sure to add those dependencies as required.

3. Add Java class to the project.
 - a. Open the context (right-click) menu for the src/main/java subdirectory in the project, choose **New**, and then choose **Class**.
 - b. In the **New Java Class** window, type the following values:
 - **Package:** **example**
 - **Name:** **Hello**

Note

If you are following other tutorial topics in this guide, the specific tutorials might recommend different package name or class name.

- c. Add your Java code. If you are following other tutorial topics in this guide, add the provided code.
4. Build the project.

Open the context (right-click) menu for the project in **Package Explorer**, choose **Run As**, and then choose **Maven Build ...**. In the **Edit Configuration** window, type **package** in the **Goals** box.

Note

The resulting .jar, lambda-java-example-0.0.1-SNAPSHOT.jar, is not the final standalone .jar that you can use as your deployment package. In the next step, you add the Apache maven-shade-plugin to create the standalone .jar. For more information, go to [Apache Maven Shade plugin](#).

5. Add the maven-shade-plugin plugin and rebuild.

The maven-shade-plugin will take artifacts (jars) produced by the *package* goal (produces customer code .jar), and created a standalone .jar that contains the compiled customer code, and the resolved dependencies from the pom.xml.

- a. Open the context (right-click) menu for the pom.xml file, choose **Maven**, and then choose **Add Plugin**.
- b. In the **Add Plugin** window, type the following values:
 - **Group Id:** org.apache.maven.plugins
 - **Artifact Id:** maven-shade-plugin
 - **Version:** 3.2.2
- c. Now build again.

This time we will create the jar as before, and then use the maven-shade-plugin to pull in dependencies to make the standalone .jar.

- i. Open the context (right-click) menu for the project, choose **Run As**, and then choose **Maven build**
- ii. In the **Edit Configuration** windows, type **package shade:shade** in the **Goals** box.
- iii. Choose Run.

You can find the resulting standalone .jar (that is, your deployment package), in the /target subdirectory.

Open the context (right-click) menu for the /target subdirectory, choose **Show In**, choose **System Explorer**, and you will find the lambda-java-example-0.0.1-SNAPSHOT.jar.