Creating Java and JavaScript Adapters

Overview

Adapters are Maven projects that contains server-side code implemented in either Java or JavaScript. An adapter project can be created by using Maven, MobileFirst developer CLI or MobileFirst Operations Console. You can edit the adapter code using your IDE of choice.

This tutorial demonstrates how to create either a Java or JavaScript adapter using the above methods.

The adapter project is based on the Maven Archetype "adapter-maven-archetype" which is based on the Maven archetype toolkit (https://maven.apache.org/guides/introduction/introduction-to-archetypes.html).

Prerequisite: Make sure that you read the Adapters Overview (../adapters-overview) tutorial first.

Jump to:

- Creating Adapters Using Maven
 - o Install Maven
 - o Create an Adapter
 - Build and Deploy Adapters
 - Dependencies
 - Grouping Adapters in a Single Maven Project
- Creating Adapters Using MobileFirst Developer CLI
- Creating Adapters Using MobileFirst Operations Console

Creating Adapters Using Maven Archetype "adapter-mavenarchetype"

The "adapter-maven-archetype" is based on the Maven archetype toolkit (https://maven.apache.org/guides/introduction/introduction-to-archetypes.html) in order to create the adapter as a Maven project.

Install Mayen

In order to create an adapter, you first need to download and install Maven. Go to the Apache Maven website (https://maven.apache.org/) and follow the instructions how to download and install Maven.

Create an Adapter

To create a Maven adapter project, use the archetype: generate command. You can choose to run the command interactively or directly.

Interactive Mode

1. Replace the **DarchetypeArtifactId** placeholder with the actual value and run:

 $mvn\ archetype: generate\ -DarchetypeGroupId=com.ibm.mfp\ -DarchetypeArtifactId=< adapter\ type\ artifact\ ID>\ -DarchetypeVersion=8.0.0$

• The Archetype Group Id and Archetype Version are required parameters to identify the archetype.

- The Archetype Artifact Id is a required parameter to identify the adapter type:
 - Use adapter-maven-archetype-java to crate a Java adapter
 - Use adapter-maven-archetype-http to create a JavaScript HTTP adapter
 - Use adapter-maven-archetype-sql to create a JavaScript SQL adapter
- 2. Enter a Group Id (https://maven.apache.org/guides/mini/guide-naming-conventions.html) of the Maven project to be build. For example:

Define value for property 'groupld': : com.mycompany

3. Enter an Artifact Id of the Maven project **which will later be used also as the adapter name** . For example:

Define value for property 'artifactId': : SampleAdapter

4. Enter a Maven project version (the default is 1.0-SNAPSH0T). For example:

Define value for property 'version': 1.0-SNAPSHOT:: 1.0

5. Enter an adapter package name (the default is the groupId). For example:

Define value for property 'package': com.mycompany::com.mypackage

6. Enter y to confirm:

[INFO] Using property: archetypeVersion = 8.0.0

Confirm properties configuration:

groupld: com.mycompany artifactId: SampleAdapter

version: 1.0

package: com.mypackage archetypeVersion: 8.0.0

Y::y

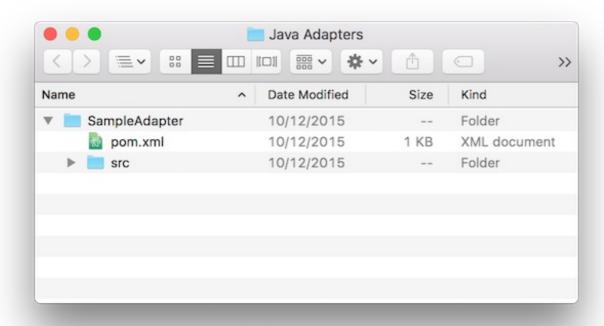
Direct Mode

Replace the placeholders with the actual values and run:

mvn archetype:generate -DarchetypeGroupId=com.ibm.mfp -DarchetypeArtifactId=<adapter type artifact ID > -DarchetypeVersion=8.0.0 -DgroupId=<maven_project_groupid> -DartifactId=<maven_project_artifactid> -Dversion=<maven_project_version> -Dpackage=<java_adapter_package_name>

For more information about the archetype: generate command see the Maven documentation.

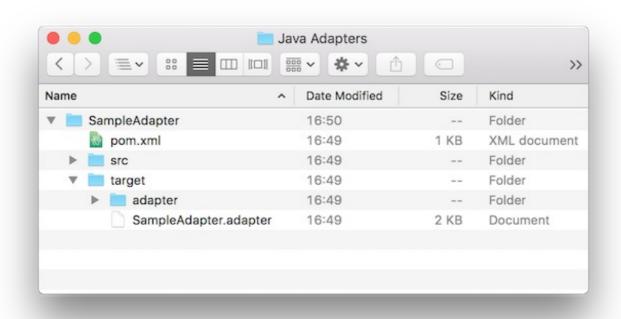
After creating the adapter the result will be a Maven project containing a src folder and a pom.xml file:



Build and Deploy Adapters

Build

The adapter is built each time you run the mvn install command to build the Maven project. This generates an **.adapter** file which can be found in the **target** folder:



Deploy

1. The pom.xml file contains the following properties parameters:

```
<properties>
    <!-- parameters for deploy mfpf adapter -->
    <mfpfUrl>http://localhost:9080/mfpadmin</mfpfUrl>
    <mfpfUser>demo</mfpfUser>
    <mfpfPassword>demo</mfpfPassword>
</properties>
```

- Replace the localhost: 9080 with your MobileFirst Server IP and port.
- Replace the mfpfUser and mfpfPassword values with your MobileFirst admin user name and password.
- 2. Open the project's root folder in terminal and run the |mvn:adapter| command:

```
mvn adapter:deploy
```

NOTE: The deploy command is available only during development (for security reasons).

Tip: You can also build and deploy the adapter using a single command: mvn install adapter:deploy

Dependencies

In order to use an external library in your adapter, follow these suggested instructions:

- 1. Add a lib folder under the root Maven project folder and put the external library in it.
- 2. Add the library path under the dependencies element in the Maven project pom.xml file. For example:

```
<dependency>
    <groupId>sample</groupId>
    <artifactId>com.sample</artifactId>
    <version>1.0</version>
    <scope>system</scope>
    <systemPath>${project.basedir}/lib/</systemPath>
    </dependency>
```

For more information about dependencies see the Maven documentation.

Grouping Adapters in a Single Maven Project

If you have several adapters in your project you may want to arrange them under a single Maven project. Grouping adapters provides many benefits such as build all and deploy all abilities, sharing dependencies etc.

To group adapters you need to:

- 1. Create a root folder and call it, for example, "GroupAdapters".
- 2. Put the Maven adapter projects in it.
- 3. Create a pom.xml file:

```
ma-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.
0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupid>com.sample
 <artifactId>GroupAdapters</artifactId>
 <version>1.0-SNAPSHOT
 <packaging>pom</packaging>
 <modules>
     <module>Adapter1</module>
     <module>Adapter2</module>
 </modules>
 cproperties>
    <!-- parameters for deploy mfpf adapter -->
   <mfpfUrl>http://localhost:9080/mfpadmin</mfpfUrl>
   <mfpfUser>demo</mfpfUser>
    <mfpfPassword>demo</mfpfPassword>
 <bul>duild>
   <plugins>
     <plugin>
       <groupid>com.ibm.mfp</groupid>
       <artifactId>adapter-maven-plugin</artifactId>
       <version>8.0.0</version>
       <extensions>true</extensions>
     </plugin>
   </plugins>
 </build>
</project>
```

- 1. Define a **groupId** element of your choice
- 2. Add an artifactId element the root folder's name
- 3. Add a **module** element for each adapter
- 4. Add the build element
- 5. Replace the **localhost:9080** with your MobileFirst Server IP and port.
- 6. Replace the **mfpfUser** and **mfpfPassword** values with your MobileFirst admin user name and password.
- 4. To build or deploy all adapters, run the commands from the root "GroupAdapters" project.

Creating Adapters Using MobileFirst Developer CLI

Creating Adapters Using MobileFirst Operations Console

- 1. Open your browser of choice and load the MobileFirst Operations Console using the address http://<IP>:<PORT>/mfpconsole/.
- 2. Click on the "Create new" button next to Adapters. You have two options to create an adapter:

- Using Maven or MobileFirst developer CLI as previously explained above.
- o Download a template adapter project (step 2).
- 3. Build the adapter Using Maven or MobileFirst developer CLI.
- 4. Choose one of the following ways to upload the generated **.adapter** file which can be found in the target folder of the adapter project:
 - o Click on the Deploy Adapter button (step 5).
 - Drag and drop the file into the Create new adapter screen.

