Creating Java and JavaScript Adapters

Overview

This tutorial demonstrates how to create either Java or JavaScript adapter using the Maven Archetype "adapter-maven-archetype". 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.

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

Install Maven

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.

Creating an Adapter

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

In Interactive Mode

1. Run:

   ```
   $ mvn archetype:generate -DarchetypeGroupId=com.ibm.mfp -DarchetypeArtifa
   ctId=<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

       `adapter-maven-archetype-jms` - JavaScript JMS adapter

       `adapter-maven-archetype-sap` - JavaScript SAP adapter

       `adapter-maven-archetype-cast-iron` - JavaScript Cast Iron adapter
2. You will be asked to enter the Group Id of the Maven project to be build:

   ```
   Define value for property 'groupId': : sample.group.id
   ```

3. You will be asked to enter the Artifact Id of the Maven project **which will later be used also as the adapter name**:

   ```
   Define value for property 'artifactId': : SampleAdapter
   ```

4. You will be asked to enter the Maven project version (the default is `1.0-SNAPSHOT`):

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

5. You will be asked to enter the Java adapter package name (the default is the `groupId`):

```
Define value for property 'package':  sample.group.id: : com.sample.adapt
er
```

6. Enter `y` to confirm:

```
[INFO] Using property: archetypeVersion = 7.2.0.0
Confirm properties configuration:
groupId: sample.group.id
artifactId: Sample
version: 1.0
package: com.sample.adapter
adapter-name: TestAdapter
archetypeVersion: 7.2.0.0
 Y: : y
```

In Direct Mode

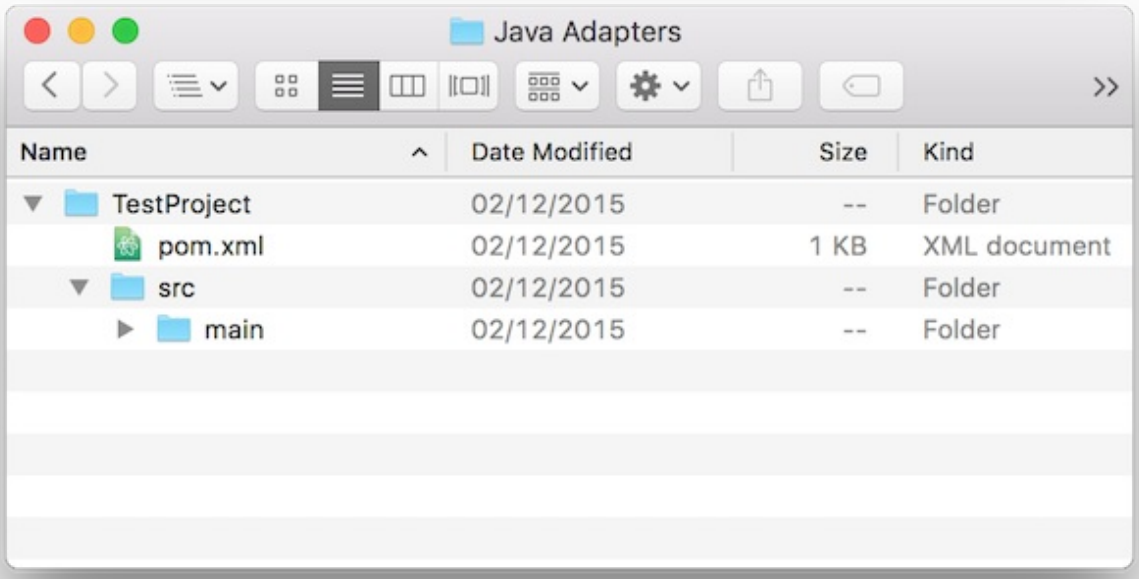
Replace the placeholders with the actual values and run:

```
$ mvn archetype:generate -DarchetypeGroupId=com.ibm.mfp -DarchetypeArtifactId=
<adapter type artifact ID> -DarchetypeVersion=8.0.0 -DgroupId=<maven_project_g
roupid> -DartifactId=<maven_project_artifactid> -Dversion=<maven_project_versi
on> -Dpackage=<java_adapter_package_name>
```

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

File Structure

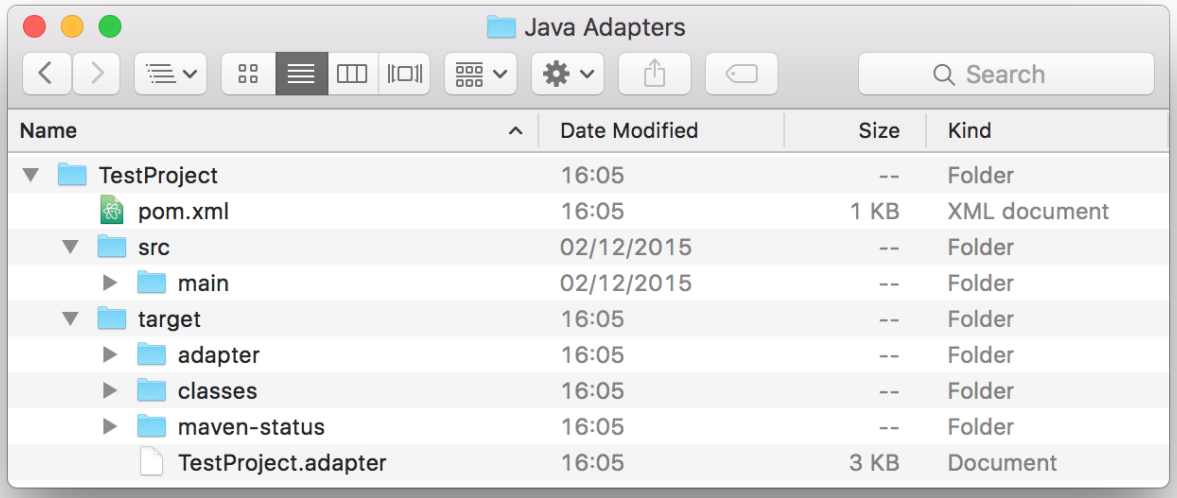
The result will be a Maven project containing a `src` folder and a `pom.xml` file:



Building and Deploying Java Adapter

Build

The adapter will be built every time you run the `mvn install` command to build your Maven project. The end result is the `.adapter` file in the project `target` folder:



Deploy

NOTE: The deploy command is available only during development.

1. Edit the `pom.xml` file with the following `configuration` parameters:

```
...
<configuration>
  <!-- parameters for deploy adapter -->
  <serverUrl>http://<IP>:<PORT>/mfpadmin</serverUrl>
  <user>ADMIN_USER</user>
  <password>ADMIN_PASSWORD</password>
</configuration>
```

- Replace the `IP` and `PORT` with the MobileFirst Server IP and port.
- Replace the `ADMIN_USER` and `ADMIN_PASSWORD` with the MobileFirst admin user and password.

2. Open the project's root folder in terminal and run the `mvn:adapter` command:

```
$ mvn adapter:deploy
```

*** You can also deploy the Java adapter using Ant tasks. For more information see the topic about Deploying applications and adapters in the user documentation.

Grouping adapters in a single maven project

Dependencies

Unlike using the Studio or the CLI `lib` folder to put your dependencies, in the Java adapter Maven project you'll put them under the `dependencies` element in the Maven project pom.xml file.

> For more information about `dependencies` see the Maven documentation.

Migrating an Existing Java Adapter

In order to migrate an existing Java adapter to Java adapter Maven project you need to:

- Create a new Maven project using the existing adapter name and adapter package name.
- Overwrite the adapter .xml file (under src/main/resources) with the existing Java adapter .xml file.
- Overwrite the adapter application and resource Java files (under src/main/java/adapter*package*name) with the existing Java adapter application and resource Java files.
- To add dependencies from the existing Java adapter `lib` folder see the Dependencies section.