A Maven Project

Creating a Project

Maven uses `archetype` `plugins` to create projects. To create a simple java application, we'll use the maven-archetype-quickstart plugin. In the example below, we'll create a maven based java application project.

Let's open a command-line interface and run the following mvn command.

```
mvn archetype:generate \
-DgroupId=com.clarus.maven \
-DartifactId=maven-experiment \
-DarchetypeArtifactId=maven-archetype-quickstart \
-DinteractiveMode=false
```

With this command, Maven will create a complete java application with sample files within a specific directory structure. Below is the output of the command we have run recently.

```
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< org.apache.maven:standalone-pom >------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] --------------------------------[ pom ]---------------------------------
[INFO]
[INFO] >>> maven-archetype-plugin:3.2.0:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:3.2.0:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO]
[INFO] --- maven-archetype-plugin:3.2.0:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.0
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: basedir, Value: /Users/home/Documents/00_ComputerScience/02_Java
[INFO] Parameter: package, Value: com.clarus.maven
[INFO] Parameter: groupId, Value: com.clarus.maven
[INFO] Parameter: artifactId, Value: maven-experiment
[INFO] Parameter: packageName, Value: com.clarus.maven
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /Users/home/Documents/00_ComputerScience/02_Java/maven-experiment
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:18 min
[INFO] Finished at: 2020-07-25T14:02:09+03:00
[INFO] ------------------------------------------------------------------------
```

Now, go into the project folder, open the directory `src/main/java/com/clarus/maven`. You will see the file `App.java`. Replace the content of the file with the content below.

```
1   package com.clarus.maven;
2
3   import java.io.BufferedReader;
4   import java.io.File;
5   import java.io.FileOutputStream;
6   import java.io.IOException;
7   import java.io.InputStream;
8   import java.io.InputStreamReader;
9   import java.io.OutputStreamWriter;
10  import java.io.Writer;
11  import java.util.UUID;
12
13  import com.amazonaws.AmazonClientException;
14  import com.amazonaws.AmazonServiceException;
15  import com.amazonaws.auth.AWSCredentials;
16  import com.amazonaws.auth.AWSStaticCredentialsProvider;
17  import com.amazonaws.auth.profile.ProfileCredentialsProvider;
18  import com.amazonaws.services.s3.AmazonS3;
19  import com.amazonaws.services.s3.AmazonS3ClientBuilder;
20  import com.amazonaws.services.s3.model.Bucket;
21  import com.amazonaws.services.s3.model.GetObjectRequest;
22  import com.amazonaws.services.s3.model.ListObjectsRequest;
23  import com.amazonaws.services.s3.model.ObjectListing;
24  import com.amazonaws.services.s3.model.PutObjectRequest;
25  import com.amazonaws.services.s3.model.S3Object;
26  import com.amazonaws.services.s3.model.S3ObjectSummary;
27
28  public class App {
29
30  public static void main( String[] args ){
31
32      /*
33       * The ProfileCredentialsProvider will return your [default]
34       * credential profile by reading from the credentials file located at
35       * (~/.aws/credentials).
36       */
37      AWSCredentials credentials = null;
38      try {
39          credentials = new ProfileCredentialsProvider().getCredentials();
40      } catch (Exception e) {
41          throw new AmazonClientException(
42                  "Cannot load the credentials from the credential profiles file. " +
43                  "Please make sure that your credentials file is at the correct " +
44                  "location (~/.aws/credentials), and is in valid format.",
45                  e);
46      }
47
48      AmazonS3 s3 = AmazonS3ClientBuilder.standard()
49          .withCredentials(new AWSStaticCredentialsProvider(credentials))
50          .withRegion("us-west-2")
51          .build();
52
53      String bucketName = "my-first-s3-bucket-" + UUID.randomUUID();
54      String key = "MyObjectKey";
55
56      System.out.println("===========================================");
57      System.out.println("Getting Started with Amazon S3");
58      System.out.println("===========================================\n");
59
60      try {
61          /*
62           * Create a new S3 bucket - Amazon S3 bucket names are globally unique,
63           * so once a bucket name has been taken by any user, you can't create
64           * another bucket with that same name.
65           *
66           * You can optionally specify a location for your bucket if you want to
67           * keep your data closer to your applications or users.
68           */
69          System.out.println("Creating bucket " + bucketName + "\n");
70          s3.createBucket(bucketName);
71
72          /*
73           * List the buckets in your account
74           */
75          System.out.println("Listing buckets");
76          for (Bucket bucket : s3.listBuckets()) {
77              System.out.println(" - " + bucket.getName());
78          }
79          System.out.println();
80
81          /*
82           * Upload an object to your bucket - You can easily upload a file to
83           * S3, or upload directly an InputStream if you know the length of
84           * the data in the stream. You can also specify your own metadata
85           * when uploading to S3, which allows you to set a variety of options
86           * like content-type and content-encoding, plus additional metadata
87           * specific to your applications.
88           */
89          System.out.println("Uploading a new object to S3 from a file\n");
90          try {
91              s3.putObject(new PutObjectRequest(bucketName, key, createSampleFile()));
92          } catch (IOException e) {
93              // TODO Auto-generated catch block
94              e.printStackTrace();
95          }
96
97          /*
```

This code automatically creates a bucket in your AWS S3 service, puts an object in the bucket, and then deletes the object and the bucket. You can see the logs of all these actions. If you don't want to delete the object and the bucket, you should comment the lines 142 and 150 just by typing `//` at the beginning of the line. So, if you check your s3, you should see the newly created bucket and the object in it.

This application heavily depends on the modules (packages) of Amazon Web Services. So our POM file should provide us these packages in the `dependencies` section. And again, if your JDK is Java 9 or higher, don't forget to uncomment the `properties` part (between line number 13 - 16) in the POM file.

```
1   <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
3     <modelVersion>4.0.0</modelVersion>
4
5   <groupId>com.clarus.maven</groupId>
6     <artifactId>maven-experiment</artifactId>
7     <packaging>jar</packaging>
8     <version>1.0-SNAPSHOT</version>
9
10  <name>maven-experiment</name>
11    <url>http://maven.apache.org</url>
12
13  <!-- <properties>
14      <maven.compiler.source>1.6</maven.compiler.source>
15      <maven.compiler.target>1.6</maven.compiler.target>
16  </properties> -->
17
18  <dependencyManagement>
19    <dependencies>
20      <dependency>
21        <groupId>com.amazonaws</groupId>
22        <artifactId>aws-java-sdk-bom</artifactId>
23        <version>1.11.827</version>
24        <type>pom</type>
25        <scope>import</scope>
26      </dependency>
27    </dependencies>
28  </dependencyManagement>
29
30  <dependencies>
31      <dependency>
32        <groupId>com.amazonaws</groupId>
33        <artifactId>aws-java-sdk-s3</artifactId>
34      </dependency>
35      <dependency>
36        <groupId>junit</groupId>
37        <artifactId>junit</artifactId>
38        <version>3.8.1</version>
39        <scope>test</scope>
40      </dependency>
41    </dependencies>
42  </project>
43
```

To be able to run the application flawlessly, you also need to have the file `~/.aws/credentials` in your home directory. And that file should have the content :

```
[default]
aws_access_key_id=<your_access_key_id>
aws_secret_access_key=<your_secret_access_key>
```

Caution: If you are using VS Code as your IDE, you should install "Java Extension Pack" as an extension.

Maven Test

If you have inspected the directory structure, you should see that under the `src` directory, there is a `test` folder. Maven automatically puts the test file as `AppTest.java`. And when you run the command `mvn test`, Maven runs this test file and outputs the results in the standard output. Now try to run the command `mvn test`, and see the result below.

```
INFO] Scanning for projects...
[INFO]
[INFO] ---------------< com.clarus.maven:maven-experiment >----------------
[INFO] Building maven-experiment 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ maven
    -experiment ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
    build is platform dependent!
[INFO] skip non existing resourceDirectory /Users/home/Documents/00_ComputerScience
    /02_Java/maven-experiment/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ maven-experiment
    ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build
    is platform dependent!
[INFO] Compiling 1 source file to /Users/home/Documents/00_ComputerScience/02_Java
    /maven-experiment/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ maven
    -experiment ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
    build is platform dependent!
[INFO] skip non existing resourceDirectory /Users/home/Documents/00_ComputerScience
    /02_Java/maven-experiment/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ maven
    -experiment ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build
    is platform dependent!
[INFO] Compiling 1 source file to /Users/home/Documents/00_ComputerScience/02_Java
    /maven-experiment/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ maven-experiment ---
[INFO] Surefire report directory: /Users/home/Documents/00_ComputerScience/02_Java
    /maven-experiment/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running com.clarus.maven.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.021 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.333 s
[INFO] Finished at: 2020-07-25T15:33:39+03:00
[INFO] ------------------------------------------------------------------------
```

Maven also puts any kind of result into `target` directory. Maybe you didn't notice but before running `mvn test` command, there wasn't a `target` directory under the project folder. Now as a next step, we're going to clean the project folder from the target folder.

Maven Clean

Sometimes consecutive Maven commands bring too many outputs and that gets you confused. At that point, Maven's `clean` command comes into rescue. With `mvn clean` you delete the `target` directory. This command generally used together with other commands like `mvn clean compile` or `mvn clean package`. The output of the command is as shown below.

```
[INFO] Scanning for projects...
[INFO]
[INFO] ---------------< com.clarus.maven:maven-experiment >----------------
[INFO] Building maven-experiment 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ maven-experiment ---
[INFO] Deleting /Users/home/Documents/00_ComputerScience/02_Java/maven-experiment
    /target
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.555 s
[INFO] Finished at: 2020-07-25T15:50:36+03:00
[INFO] ------------------------------------------------------------------------
```

Maven Package

For a developer, the dependency injection part of Maven would be of utmost importance. But for a DevOps Engineer, building and packaging tools have more importance. Because packaging gives us the binary executable file which is a jar, war or ear file in this example and it's the phase where the application turns into a shippable state through its lifecycle. If you run the command `mvn clean package`, you should see the output as below and the JAR file should reside under `target` directory.

```
[INFO] Scanning for projects...
[INFO]
[INFO] ---------------< com.clarus.maven3:maven-experiment >----------------
[INFO] Building maven-experiment 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ maven-experiment ---
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ maven
    -experiment ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
    build is platform dependent!
[INFO] skip non existing resourceDirectory /Users/home/Documents/00_ComputerScience
    /02_Java/maven-experiment/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ maven-experiment
    ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build
    is platform dependent!
[INFO] Compiling 1 source file to /Users/home/Documents/00_ComputerScience/02_Java
    /maven-experiment/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ maven
    -experiment ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
    build is platform dependent!
[INFO] skip non existing resourceDirectory /Users/home/Documents/00_ComputerScience
    /02_Java/maven-experiment/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ maven
    -experiment ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build
    is platform dependent!
[INFO] Compiling 1 source file to /Users/home/Documents/00_ComputerScience/02_Java
    /maven-experiment/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ maven-experiment ---
[INFO] Surefire report directory: /Users/home/Documents/00_ComputerScience/02_Java
    /maven-experiment/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running com.clarus.maven.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.031 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ maven-experiment ---
[INFO] Building jar: /Users/home/Documents/00_ComputerScience/02_Java/maven
    -experiment/target/maven-experiment-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.449 s
[INFO] Finished at: 2020-07-25T16:05:27+03:00
[INFO] ------------------------------------------------------------------------
```

Maven Site

If you want a webpage that shows the project information, `mvn site` is the command you are looking for. It produces the HTML pages automatically for you. But before running the command, override `site-plugin` and `project-info-reports-plugin` with the ones shown below. You should place them into `plugins` which is under `build` tag.

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-site-plugin</artifactId>
    <version>3.7.1</version>
</plugin>
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-project-info-reports-plugin</artifactId>
    <version>3.0.0</version>
</plugin>
```

When you run the command, you will see a long output. But also you should see the HTML pages under `/target/site` folder.