

Maven Metamodel Generator

This document explains how to configure the maven plugin to generate metamodel classes from your Blueriq model.

Getting started

Create a Maven project with the following POM:

pom.xml

```
<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/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.mycustomer</groupId>
  <artifactId>myproject-metamodel</artifactId>
  <version>1.0-SNAPSHOT</version>
  <name>myproject-metamodel</name>
  <properties>
    <com.blueriq.version>10.0-SNAPSHOT</com.blueriq.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>com.blueriq</groupId>
      <artifactId>blueriq-sdk</artifactId>
      <version>${com.blueriq.version}</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.6.1</version>
        <configuration>
          <source>1.7</source>
          <target>1.7</target>
        </configuration>
      </plugin>
      <plugin>
        <groupId>com.blueriq</groupId>
        <artifactId>blueriq-metamodelgenerator-maven-plugin</artifactId>
        <version>${com.blueriq.version}</version>
        <executions>
          <execution>
            <phase>generate-sources</phase>
            <goals>
              <goal>generate-metamodel</goal>
            </goals>
          </execution>
        </executions>
        <configuration>
          <connections>
            <studio>
              <repository>MyRepository</repository>
              <branch>Trunk</branch>
              <project>MyProject</project>
              <destDir>src/main/java</destDir>
              <packageName>org.mycustomer.myproject.metamodel<
                /packageName>
              <url>http://SERVER:PORT/Studio/Server/Services
                /ManagementService</url>
              <authentication>BASIC</authentication>
              <userName></userName>
              <password></password>
              <exportProject>true</exportProject>
              <generatePojos>true</generatePojos>
              <generateRequiredMethods>true</generateRequiredMethods>
              <generateInstanceLocator>true</generateInstanceLocator>
            </studio>
          </connections>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

```

<includeSystemEntityObjects>true</includeSystemEntityObjects>
    </studio>

    <offline>
        <applicationFile>${project.basedir}/MyProject.project.zip</applicationFile>

        <destDir>src/main/java</destDir>
        <packageName>org.mycustomer.myproject.metamodel<

        <generatePojos>true</generatePojos>
        <generateRequiredMethods>true</generateRequiredMethods>
        <generateInstanceLocator>true</generateInstanceLocator>
        <includeSystemEntityObjects>true<

    </offline>
    </connections>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

Configuration of the POM

You have to define the location of your Blueriq model(s) from which you will generate metamodel classes. The models can be retrieved directly from the Studio server or from an export. To define this, within the <connections> section you can add one or more <studio> (connection to Studio server) or <offline> (export) connections.

Studio server connection

Parameter	Description	Required	Default value
repository	Name of the Blueriq repository	Yes	
branch	Name of the Blueriq branch	Yes	
project	Name of the Blueriq project	Yes	
destDir	Destination directory where all the java classes are stored	Yes	src/main/java
packageName	Name of the base java package	Yes	
url	URL of the management service of the Studio server	Yes	
authentication	Authentication method. Since R16 only BASIC is supported	Yes	BASIC
userName	Blueriq username	Yes	
passWord	Blueriq password	Yes	
exportProject	Set to true if you want to generate the Blueriq project export	No	false
exportProjectDir	Foldername to save the export	No	target
generatePojos	Set to true if you want to generate the POJO class of a Blueriq Entity	No	false
generateRequiredMethods	Set to true if you want to generate the getRequired<attribute> methods	No	false
generateInstanceLocator	Set to true if you want to use the InstanceLocator class	No	false
includeSystemEntityObjects	Set to true if you want to add classes for system object like the system and user entity	No	false

Offline connection

This connection will pick up the given file name and presumes there is an export in it. It will then use the export to create the metamodel.

Parameter	Description	Required	Default value
applicationFile	Path of the export file	Yes	\${project.basedir}/MyProject.project.zip
destDir	Destination directory where all the java classes are stored	Yes	src/main/java
packageName	Name of the base java package	Yes	
generatePojos	Set to true if you want to generate the POJO class of a Blueriq Entity	No	false
generateRequiredMethods	Set to true if you want to generate the getRequired<attribute> methods	No	false
generateInstanceLocator	Set to true if you want to use the InstanceLocator class	No	false

includeSystemEntityObjects	Set to true if you want to add classes for system object like the system and user entity	No	false
----------------------------	--	----	-------