# MyBatis Generator Configuration

HotRod's MyBatis Generator generates all DAO classes, mappers, and MyBatis configuration file ready to be used by your application.

The example shown below, taken from the Hello World example application, shows a typical configuration for the MyBatis Generator:

<?xml version=*"1.0"*?>

<!DOCTYPE hotrod SYSTEM "hotrod.dtd">

<hotrod>

<generators>

<mybatis>

<daos gen-base-dir=*"auto-generated/java"* dao-package=*"daos"* />

<mappers gen-base-dir=*"auto-generated/mappers"* relative-dir=*"persistence"* />

<mybatis-configuration-template file=*"mybatis-template.xml"* />

<session-factory singleton-full-class-name=*"sessionfactory.DatabaseSessionFactory"* />

<select-generation temp-view-base-name=*"hotrod\_temp\_view"* />

</mybatis>

</generators>

<table name=*"vehicle"*>

<auto-generated-column name=*"id"* />

</table>

</hotrod>

The <mybatis> tag is added inside the <generators> tag and includes all the configuration options the MyBatis Generator needs to produce fully functional MyBatis code.

## The <mybatis> Tag

The <mybatis> tag provides the details of the MyBatis Generator. It includes:

* a <daos> tag,
* a <mappers> tag,
* a <mybatis-configuration-template> tag,
* a <session-factory> tag, and
* a <select-generation> tag.

## The <daos> Tag

The <daos> tag indicates where the DAO Java classes are going to be placed and how their names are going to be produced. Its attributes are:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Required** |
| gen-base-dir | The base dir of the source Java classes. This is the base dir where the Java packages will be created on. | Yes |
| dao-package | The base package of the DAO Java classes. The base package adds directories to place the DAO Java classes inside the gen-base-dir attribute. | Yes |
| primitives-package | Specifies the extra package added to the DAO primitive Java classes. Defaults to: primitives | No |
| dao-prefix | Specifies the prefix to prepend to the DAO Java class name. Defaults to an empty string, i.e. no prefix. | No |
| dao-suffix | Specifies the suffix to append to the DAO Java class name. Defaults to: DAO | No |
| primitives-prefix | Specifies the prefix to prepend to the DAO Primitives Java class name. Defaults to an empty string, i.e. no prefix. | No |
| primitives-suffix | Specifies the suffix to append to the DAO Primitives Java class name. Defaults to: Primitives | No |

## The <mappers> tag

The <mappers> tag indicates where the mappers are going to be placed. Its attributes are:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Required** |
| gen-base-dir | The base dir of the generated MyBatis mapper files. Mapper files will be places in the relative-dir directory based on this gen-base-dir. | Yes |
| relative-dir | This is the relative dir where the MyBatis mapper files will be generated. This directory is relative to the gen-base-dir. The relative-dir is the directory the main MyBatis configuration file will use to reference the mapper files. | Yes |

## The <mybatis-configuration-template> tag

The <mybatis-configuration-template> tag specifies which template file will be used to generate the main MyBatis configuration file. Since the MyBatis configuration options are abundant, HotRod's MyBatis Generator takes a template file prepared by the developer and adds the full list of generated mappers to it. The template file is not rewritten, but a new ready-to-use file is produced in the mappers dir with the name mybatis-configuration.xml.

For details on the MyBatis configuration file please see the MyBatis documentation. As a brief overview, this file specifies the data source properties, as well as many MyBatis tweaks, and JDBC properties. The template file, however, includes an empty <mappers/> tag that will be replaced will the full list of mappers in the resulting from the code generation.

The attributes of the <mybatis-configuration-template> tag are:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Required** |
| file | The path and name of the MyBatis configuration template file. | Yes |

## The <session-factory> tag

The <session-factory> tag specifies the Java singleton class the developer will provide to produce MyBatis SQLSession objects all the DAOs need to operate. Its attributes are:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Required** |
| singleton-full-class-name | The full class name of the singleton class that will be used by the DAOs to produce SQLSession objects. | Yes |

## The <select-generation> tag

The <select-generation> tag specifies the details of the temporary database views HotRod will create to process the <select> tags. The full explanation is quite technical, but suffice is to say that HotRod needs to create some temporary database views in the development schemas (never in test or production) that are deleted right away during the code generation. HotRod, however, needs a base name to use when creating them so to avoid name conflicts with any existing application view. Its attributes are:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Required** |
| temp-view-base-name | The base name to create temporary database views in the development database. Choose a base name not used in your database and that clearly indicates the purpose of the view, such as hotrod-temp-view. All temp views will use this value and append a consecutive number to it when created. | Yes |

**Note**: if you use the <select> tag your database user must have privileges to create and to drop database views in the database schema where you are developing the application.