# Configuration Reference Overview

HotRod's Configuration File is a key aspect of the code generator. This file gathers more than a decade of experience of the authors on code generation for different projects with diverse needs.

In it simplest form, the configuration file includes a list of the tables the application uses. You can generate a comprehensive persistence layer for MyBatis just by writing a few lines in the configuration file.

In more advanced form, this file can specify multiple aspects of the code generation or add tweaks to it. It's not required for a developer to learn all the tweaks right from the beginning, but he/she will find them useful later on during the project when the need arises.

When using it comprehensively, the configuration file can specify Java names and types for DAOs, add optimized SQL sentences, provide facets subset regeneration, add custom DAOs, divide it in separate areas, and other tweaks. These are advanced features useful for more optimized projects, when having multiple modules, or when having many developers.

## The Main Structure

The configuration file includes a generators section that stipulates details on the whole code generation, and then the details of the database objects that will be considered on it.

The header of the configuration file specifies the main directives to HotRod. Among others, it defines the directory structure for the generated files such as the DAO Java classes, MyBatis mappers, and MyBatis main configuration file.

The example shown below, taken from the Hello World example application, depicts these sections:

<?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 gray lines are boilerplate text and should be included verbatim. They correspond to the main XML structure and definition of the XML grammar.

The section in red corresponds to the generators section. It provides details on how the generated code will be produced. There are two generators available in HotRod: the MyBatis Generator, and the Spring JDBC Generator.

The section in blue includes the full list of database objects your application uses. In this case this section is very short and includes a single table. More developed applications include a bulk of definitions in here.

For details on the Generators section see the MyBatis Generator and Spring JDBC Generator sections.

For details on the list of database objects see the corresponding Configuration Reference section for Tables, Views, Selects, Custom DAOs, Updates, Sequences, Fragments and Facets.