# Welcome to HotRod!

HotRod is a lightweight Java code generator for MyBatis.

HotRod is also developing support for Spring JDCB and this generator is in experimental mode as of version 1.0.0.

HotRod's main goal is to dramatically reduce the amount of time devoted to writing and debugging the persistence layer of a Java application. HotRod uses JDBC to retrieve the structure of an existing database and generates all DAO Java classes that expose simple Java methods to perform all interaction with the database.

All SQL code is supported—even native SQL and dynamic SQL. The developer doesn't see any JDBC code anymore. Nada, nil, zip!

All basic operations are made automatically available, including CRUD, FK navigation, and byExample queries. When complex, native, and/or dynamic SQL is needed the developer provides the SQL statements and HotRod takes it from there, automating all the rest.

HotRod is a lightweight O/R mapper. Its goal is to dramatically cut down the development effort of the persistence layer without entering the realm of heavyweight frameworks, tools, or libraries. There is no inheritance, aggregation or composition, no heavy objects, no entity EBJs, no obscure caching, no byzantine layer of abstraction. All generated DAOs/VOs stay close to the tables, views, and SQL statements for easy understanding and debugging. Additionally it generates a high performance persistence layer since it produces minimal SQL queries, even when complex logic is needed.

The MyBatis Generator produces all the DAOs, the mappers, and the main MyBatis configuration file. The Spring JDBC Generator (experimental) produces all the Java interfaces and concrete classes needed to communicate with the database.  
  
Additionally, all generated DAOs support adding custom logic to them. Upon database changes they can be rapidly regenerated without losing the custom logic.

HotRod includes simplified MyBatis transactions and standard MyBatis transactions.

HotRod is very flexible while generating the DAOs. Even though it automates the names for the tables, views, sequences, and columns, all of them can be overridden at will by the developer using the configuration file. Likewise, HotRod provides default java types for all database columns that can also be overridden as needed.  
  
HotRod supports Oracle database, IBM DB2, SAP ASE (ex-Sybase), Microsoft SQL Server, PostgreSQL, MySQL, MariaDB, HyperSQL (HSQLDB), and H2 databases.