

MyBatis

From Wikipedia, the free encyclopedia

MyBatis is a Java persistence framework that couples objects with stored procedures or SQL statements using an XML descriptor or annotations.

MyBatis is free software that is distributed under the Apache License 2.0.

MyBatis is a fork of iBATIS 3.0 and is maintained by a team that includes the original creators of iBATIS.

Contents

- 1 Feature Summary
- 2 Usage
- 3 Spring Integration
- 4 Velocity Language
- 5 MyBatis Generator
- 6 MyBatis Migrations
- 7 History
- 8 See also
- 9 References
- 10 External links

MyBatis

Developer(s)	The MyBatis team
Stable release	3.3.0 / May 24, 2015
Development status	Active
Written in	Java
Operating system	Cross-platform
Type	persistence framework
License	Apache License 2.0
Website	http://www.mybatis.org

Feature Summary

Unlike ORM frameworks, MyBatis does not map Java objects to database tables but Java methods to SQL statements.

MyBatis lets you use all your database functionality like stored procedures, views, queries of any complexity and vendor proprietary features. It is often a good choice for legacy or de-normalized databases or to obtain full control of SQL execution.

It simplifies coding compared to JDBC. SQL statements are executed with a single line.

MyBatis provides a mapping engine that maps SQL results to object trees in a declarative way.

SQL statements can be built dynamically by using a built-in language with XML-like syntax or with Apache Velocity using the Velocity integration plugin.

MyBatis integrates with Spring Framework and Google Guice. This feature allows one to build business code free of dependencies.

MyBatis supports declarative data caching. A statement can be marked as cacheable so any data retrieved from the database will be stored in a cache and future executions of that statement will retrieve the cached data instead hitting the database. MyBatis provides a default cache implementation based on a Java HashMap and default connectors for integrating with: OSCache, Ehcache, Hazelcast and Memcached. It provides an API to plug other cache implementations.

Usage

SQL statements are stored in XML files or annotations. Below depicts a MyBatis mapper, that consists of a Java interface with some MyBatis annotations:

```
package org.mybatis.example;

public interface BlogMapper {
    @Select("select * from Blog where id = #{id}")
    Blog selectBlog(int id);
}
```

The sentence is executed as follows.

```
BlogMapper mapper = session.getMapper(BlogMapper.class);
Blog blog = mapper.selectBlog(101);
```

SQL statements and mappings can also be externalized to an XML file as follows.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="org.mybatis.example.BlogMapper">
    <select id="selectBlog" parameterType="int" resultType="Blog">
        select * from Blog where id = #{id}
    </select>
</mapper>
```

Statements can also be executed using the MyBatis API.

```
Blog blog = session.selectOne("org.mybatis.example.BlogMapper.selectBlog", 101);
```

For details, please refer to the User Guide available at MyBatis site. See external links.

Spring Integration

MyBatis integrates with Spring Framework. This module allows MyBatis to participate in Spring transactions. It will also build MyBatis mappers and sessions and inject them into other beans.

The following sample shows a basic XML configuration that sets up a mapper and injects it into a "BlogService" bean.

```
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
  <property name="dataSource" ref="dataSource" />
</bean>

<bean id="blogMapper" class="org.mybatis.spring.mapper.MapperFactoryBean">
  <property name="sqlSessionFactory" ref="sqlSessionFactory" />
  <property name="mapperInterface" value="org.mybatis.example.BlogMapper" />
</bean>

<bean id="blogService" class="org.mybatis.example.BlogServiceImpl">
  <property name="blogMapper" ref="blogMapper" />
</bean>
```

Calling MyBatis is now just calling a bean:

```
public class BlogServiceImpl implements BlogService {

    private BlogMapper blogMapper;

    public void setBlogMapper(BlogMapper blogMapper) {
        this.blogMapper = blogMapper;
    }

    public void doSomethingWithABlog(int blogId) {
        Blog blog = blogMapper.selectBlog(blogId);
        ...
    }
}
```

Velocity Language

The Velocity language driver lets you use Apache Velocity to generate your dynamic SQL queries on the fly.

```
<select id="findPerson" lang="velocity">
  #set( $pattern = $_parameter.name + '%' )
  SELECT *
  FROM person
  WHERE name LIKE @{pattern, jdbcType=VARCHAR}
</select>
```

MyBatis Generator

MyBatis provides a code generator. MyBatis Generator will introspect a database table (or many tables) and generate MyBatis artifacts needed to perform CRUD operations (Create, Retrieve, Update, Delete).

It will preserve any custom code in case of regeneration. (only if you use the eclipse plugin)

An Eclipse plugin is available.

MyBatis Migrations

MyBatis Migrations is a Java command line tool that keeps track of database schema changes managing DDL files (known as migrations).

Migrations allows to query the current status of the database, apply schema changes and also undo them. It also helps to detect and solve concurrent database schema changes made by different developers.

History

MyBatis project is a subsidiary of iBATIS 3.0 and maintained by a team which includes the original creators of iBATIS.

The project was created on May 19, 2010 when Apache iBATIS 3.0 was published and the team announced that the development will continue under a new name and a new home at Google Code.^[1]

On November 10, 2013 the project announced the movement to Github ^[2]

See also

- Hibernate
- iBATIS
- JDBC
- Java Persistence API
- EclipseLink
- Ebean
- jOOQ
- Apache Cayenne
- pureQuery
- NHydrate
- Hydrate (<http://hydrate.sourceforge.net/>)
- OpenJPA
- Spring Framework
- Google Guice
- O/R Broker (<http://www.orbroker.org>)
- ActiveJPA

References

1. iBATIS Project Team Moving to Google Code (http://mail-archives.apache.org/mod_mbox/ibatis-user-java/201005.mbox/%3CAANLkTimXoLiHwI-3kbW6It7mH0771xJP4RqT609VKCXC@mail.gmail.com%3E)
2. Bye Google Code welcome Github (<http://blog.mybatis.org/2013/11/bye-google-code-welcome-github.html>)

External links

- Official website (<http://www.mybatis.org>)

Retrieved from "<https://en.wikipedia.org/w/index.php?title=MyBatis&oldid=674496624>"

Categories: Object-relational mapping | Persistence frameworks

- This page was last modified on 4 August 2015, at 09:22.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.