**Springboot 整合 Mybatis 的完整 Web 案例**

摘要: 原创出处:www.bysocket.com 泥瓦匠BYSocket 希望转载，保留摘要，谢谢！

推荐一本书《腾讯传》。

新年第一篇 Springboot 技术文诞生。泥瓦匠准备写写 Springboot 相关最佳实践。一方面总结下一些 Springboot 相关，一方面和大家交流交流 Springboot 框架。

现在业界互联网流行的数据操作层框架 Mybatis，下面详解下 Springboot 如何整合 Mybatis ，这边没有使用 Mybatis Annotation 这种，是使用 xml 配置 SQL。因为我觉得 SQL 和业务代码应该隔离，方便和 DBA 校对 SQL。二者 XML 对较长的 SQL 比较清晰。

**一、运行 springboot-mybatis 工程**

git clone 下载工程 springboot-learning-example ，项目地址见 GitHub。下面开始运行工程步骤（Quick Start）：

**1.数据库准备**

a.创建数据库 springbootdb：

CREATE DATABASE springbootdb;

b.创建表 city ：(因为我喜欢徒步)

DROP TABLE IF EXISTS `city`;

CREATE TABLE `city` (

`id` int(10) unsigned NOT NULL AUTO\_INCREMENT COMMENT '城市编号',

`province\_id` int(10) unsigned NOT NULL COMMENT '省份编号',

`city\_name` varchar(25) DEFAULT NULL COMMENT '城市名称',

`description` varchar(25) DEFAULT NULL COMMENT '描述',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8;

c.插入数据

INSERT city VALUES (1 ,1,'温岭市','BYSocket 的家在温岭。');

**2. 项目结构介绍**

项目结构如下图所示：

org.spring.springboot.controller - Controller 层

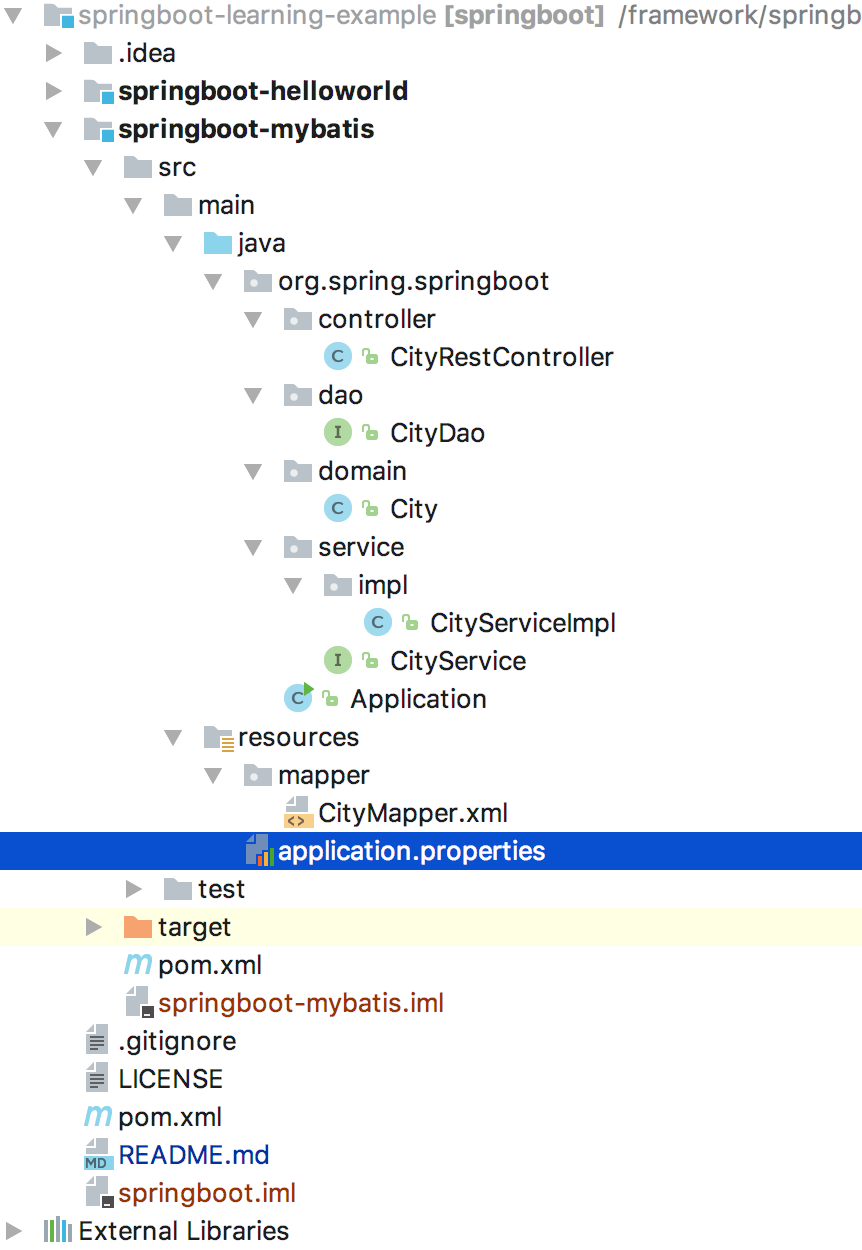
org.spring.springboot.dao - 数据操作层 DAO

org.spring.springboot.domain - 实体类

org.spring.springboot.service - 业务逻辑层

Application - 应用启动类

application.properties - 应用配置文件，应用启动会自动读取配置



**3.改数据库配置**

打开 application.properties 文件， 修改相应的数据源配置，比如数据源地址、账号、密码等。（如果不是用 MySQL，自行添加连接驱动 pom，然后修改驱动名配置。）

**4.编译工程**

在项目根目录 springboot-learning-example，运行 maven 指令：

mvn clean install

**5.运行工程**

右键运行 Application 应用启动类的 main 函数，然后在浏览器访问：

http://localhost:8080/api/city?cityName=温岭市

可以看到返回的 JSON 结果：

{

"id": 1,

"provinceId": 1,

"cityName": "温岭市",

"description": "我的家在温岭。"

}

如图：  


**二、springboot-mybatis 工程配置详解**

1.pom 添加 Mybatis 依赖

<!-- Spring Boot Mybatis 依赖 -->

<dependency>

<groupId>org.mybatis.spring.boot</groupId>

<artifactId>mybatis-spring-boot-starter</artifactId>

<version>${mybatis-spring-boot}</version>

</dependency>

整个工程的 pom.xml:

<?xml version="1.0" encoding="UTF-8"?>

<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>springboot</groupId>

<artifactId>springboot-mybatis</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>springboot-mybatis :: 整合 Mybatis Demo</name>

<!-- Spring Boot 启动父依赖 -->

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.5.1.RELEASE</version>

</parent>

<properties>

<mybatis-spring-boot>1.2.0</mybatis-spring-boot>

<mysql-connector>5.1.39</mysql-connector>

</properties>

<dependencies>

<!-- Spring Boot Web 依赖 -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Spring Boot Test 依赖 -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<!-- Spring Boot Mybatis 依赖 -->

<dependency>

<groupId>org.mybatis.spring.boot</groupId>

<artifactId>mybatis-spring-boot-starter</artifactId>

<version>${mybatis-spring-boot}</version>

</dependency>

<!-- MySQL 连接驱动依赖 -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>${mysql-connector}</version>

</dependency>

<!-- Junit -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

</dependency>

</dependencies>

</project>

2.在 application.properties 应用配置文件，增加 Mybatis 相关配置

## Mybatis 配置

mybatis.typeAliasesPackage=org.spring.springboot.domain

mybatis.mapperLocations=classpath:mapper/\*.xml

mybatis.typeAliasesPackage 配置为 org.spring.springboot.domain，指向实体类包路径。mybatis.mapperLocations 配置为 classpath 路径下 mapper 包下，\* 代表会扫描所有 xml 文件。

mybatis 其他配置相关详解如下：

mybatis.config = mybatis 配置文件名称

mybatis.mapperLocations = mapper xml 文件地址

mybatis.typeAliasesPackage = 实体类包路径

mybatis.typeHandlersPackage = type handlers 处理器包路径

mybatis.check-config-location = 检查 mybatis 配置是否存在，一般命名为 mybatis-config.xml

mybatis.executorType = 执行模式。默认是 SIMPLE

3.在 Application 应用启动类添加注解 MapperScan  
Application.java 代码如下：

/\*\*

\* Spring Boot 应用启动类

\*

\* Created by bysocket on 16/4/26.

\*/

// Spring Boot 应用的标识

@SpringBootApplication

// mapper 接口类扫描包配置

@MapperScan("org.spring.springboot.dao")

public class Application {

public static void main(String[] args) {

// 程序启动入口

// 启动嵌入式的 Tomcat 并初始化 Spring 环境及其各 Spring 组件

SpringApplication.run(Application.class,args);

}

}

mapper 接口类扫描包配置注解 MapperScan ：用这个注解可以注册 Mybatis mapper 接口类。

4.添加相应的 City domain类、CityDao mapper接口类  
City.java:

/\*\*

\* 城市实体类

\*

\* Created by bysocket on 07/02/2017.

\*/

public class City {

/\*\*

\* 城市编号

\*/

private Long id;

/\*\*

\* 省份编号

\*/

private Long provinceId;

/\*\*

\* 城市名称

\*/

private String cityName;

/\*\*

\* 描述

\*/

private String description;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Long getProvinceId() {

return provinceId;

}

public void setProvinceId(Long provinceId) {

this.provinceId = provinceId;

}

public String getCityName() {

return cityName;

}

public void setCityName(String cityName) {

this.cityName = cityName;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

}

CityDao.java:

/\*\*

\* 城市 DAO 接口类

\*

\* Created by bysocket on 07/02/2017.

\*/

public interface CityDao {

/\*\*

\* 根据城市名称，查询城市信息

\*

\* @param cityName 城市名

\*/

City findByName(@Param("cityName") String cityName);

}

其他不明白的，可以 git clone 下载工程 springboot-learning-example ，工程代码注解很详细。 [https://github.com/JeffLi1993/springboot-learning-example。](https://github.com/JeffLi1993/springboot-learning-example%E3%80%82)

**三、其他**

利用 Mybatis-generator自动生成代码 <http://www.cnblogs.com/yjmyzz/p/4210554.html>  
Mybatis 通用 Mapper3 <https://github.com/abel533/Mapper>  
Mybatis 分页插件 PageHelper <https://github.com/pagehelper/Mybatis-PageHelper>

欢迎扫一扫我的公众号关注 — 及时得到博客订阅哦！  
— <http://www.bysocket.com/> —  
— <https://github.com/JeffLi1993> —