**Spring Boot 集成 FreeMarker**

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

“年轻就不应该让自己过得太舒服” -  From yong

**一、Springboot 那些事**

SpringBoot 很方便的集成 FreeMarker ，DAO 数据库操作层依旧用的是 Mybatis，本文将会一步一步到来如何集成 FreeMarker 以及配置的详解：

Springboot 那些事：

系类文章：

[《Spring Boot 之 RESRful API 权限控制》](http://www.bysocket.com/?p=1080)

[《Spring Boot 之 HelloWorld详解》](http://www.bysocket.com/?p=1124)

[《Springboot 整合 Mybatis 的完整 Web 案例》](http://www.bysocket.com/?p=1610)

[《Springboot 实现 Restful 服务，基于 HTTP / JSON 传输》](http://www.bysocket.com/?p=1627)

[《Springboot 集成 FreeMarker》](http://www.bysocket.com/?p=1666)

**二、运行 springboot-freemarker 工程**

git clone 下载工程 springboot-learning-example ，项目地址见 [GitHub](https://github.com/JeffLi1993/springboot-learning-example) - <https://github.com/JeffLi1993/springboot-learning-example>。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

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

resources/web - \*.ftl文件，是 FreeMarker 文件配置路径。在 application.properties 配置

resources/mapper - DAO Maper XML 文件

**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，pom 自行添加连接驱动依赖，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 springboot-freemarker 工程 Application 应用启动类的 main 函数，然后在浏览器访问：

获取 ID 编号为 1 的城市信息页面：

localhost:8080/api/city/1

获取城市列表页面：

localhost:8080/api/city

**6.补充**

运行环境：JDK 7 或 8，Maven 3.0+

技术栈：SpringBoot、Mybatis、FreeMarker

**三、 springboot-freemarker 工程配置详解**

具体代码见 GitHub - <https://github.com/JeffLi1993/springboot-learning-example>

**1.pom.xml 依赖**

pom.xml 代码如下：

<?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-freemarker</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <name>springboot-freemarker :: Spring Boot 集成 FreeMarker 案例</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 Freemarker 依赖 -->

        <dependency>

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

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

        </dependency>

        <!-- 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>

在 pom.xml 依赖中增加 Spring Boot FreeMarker 依赖。

**2.配置 FreeMarker**

然后在 application.properties 中加入 FreeMarker 相关的配置：

## Freemarker 配置

## 文件配置路径

spring.freemarker.template-loader-path=classpath:/web/

spring.freemarker.cache=false

spring.freemarker.charset=UTF-8

spring.freemarker.check-template-location=true

spring.freemarker.content-type=text/html

spring.freemarker.expose-request-attributes=true

spring.freemarker.expose-session-attributes=true

spring.freemarker.request-context-attribute=request

spring.freemarker.suffix=.ftl

这是我这块的配置，如果需要更多的 FreeMarker 配置，可以查看下面的详解：

spring.freemarker.allow-request-override=false # Set whether HttpServletRequest attributes are allowed to override (hide) controller generated model attributes of the same name.

spring.freemarker.allow-session-override=false # Set whether HttpSession attributes are allowed to override (hide) controller generated model attributes of the same name.

spring.freemarker.cache=false # Enable template caching.

spring.freemarker.charset=UTF-8 # Template encoding.

spring.freemarker.check-template-location=true # Check that the templates location exists.

spring.freemarker.content-type=text/html # Content-Type value.

spring.freemarker.enabled=true # Enable MVC view resolution for this technology.

spring.freemarker.expose-request-attributes=false # Set whether all request attributes should be added to the model prior to merging with the template.

spring.freemarker.expose-session-attributes=false # Set whether all HttpSession attributes should be added to the model prior to merging with the template.

spring.freemarker.expose-spring-macro-helpers=true # Set whether to expose a RequestContext for use by Spring's macro library, under the name "springMacroRequestContext".

spring.freemarker.prefer-file-system-access=true # Prefer file system access for template loading. File system access enables hot detection of template changes.

spring.freemarker.prefix= # Prefix that gets prepended to view names when building a URL.

spring.freemarker.request-context-attribute= # Name of the RequestContext attribute for all views.

spring.freemarker.settings.\*= # Well-known FreeMarker keys which will be passed to FreeMarker's Configuration.

spring.freemarker.suffix= # Suffix that gets appended to view names when building a URL.

spring.freemarker.template-loader-path=classpath:/templates/ # Comma-separated list of template paths.

spring.freemarker.view-names= # White list of view names that can be resolved.

**3.展示层 Controller 详解**

/\*\*

 \* 城市 Controller 实现 Restful HTTP 服务

 \* <p>

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

 \*/

@Controller

public class CityController {

    @Autowired

    private CityService cityService;

    @RequestMapping(value = "/api/city/{id}", method = RequestMethod.GET)

    public String findOneCity(Model model, @PathVariable("id") Long id) {

        model.addAttribute("city", cityService.findCityById(id));

        return "city";

    }

    @RequestMapping(value = "/api/city", method = RequestMethod.GET)

    public String findAllCity(Model model) {

        List<City> cityList = cityService.findAllCity();

        model.addAttribute("cityList",cityList);

        return "cityList";

    }

}

a.这里不是走 HTTP + JSON 模式，使用了 @Controller 而不是先前的 @RestController

b.方法返回值是 String 类型，和 application.properties 配置的 Freemarker 文件配置路径下的各个 \*.ftl 文件名一致。这样才会准确地把数据渲染到 ftl 文件里面进行展示。

c.用 Model 类，向 Model 加入数据，并指定在该数据在 Freemarker 取值指定的名称。

**四、小结**

FreeMarker 是常用的模板引擎，很多开发 Web 的必选。

推荐阅读《[Springboot 那些事](http://www.bysocket.com/?page_id=1639)》

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 —

[](http://www.bysocket.com/wp-content/uploads/2017/01/qrcode_for_gh_cd421e7eb7d6_430.jpg)