



- JPA (23)
- HIBERNATE (79)
- JSONP (17)
- 事务 (34)
- 游标 (3)
- 反射机制 (24)
- JXL与POI (31)
- REST (11)
- WEBSERVICE (45)
- C C++ (2)
- HTML (11)
- URLConnection (10)
- JNI (1)
- HTTP请求 (17)
- TOMCAT (28)
- SERVLET (25)
- JDK JRE (3)
- JSP标签 (9)
- IO (8)
- 资源文件 (5)
- 定时任务调度 (29)
- COOKIE (4)
- IFRAME (5)
- 浏览器兼容性 (16)
- 动态代理 (7)
- 设计模式 (32)
- SPRING-MVC (103)
- SPRING模板类 (20)
- 批量处理 (5)
- JDBC (4)
- JAX-WS (3)
- ECLIPSE (16)
- EXE4J (1)
- 多线程 (37)
- 算法 (23)
- 编码 (16)
- 分页 (2)
- SOCKET (2)
- 闭包 (1)
- SQL HQL语句 (23)
- TEXTAREA显示 (5)
- 性能优化 (1)
- DISCUZ (1)
- JERSEY (6)
- 面试 (50)
- CLOB与BLOB (6)
- HTTPCLIENT (45)
- SMB (3)
- JCIFS (2)
- SVN (6)
- 断点续传 (1)
- EJB (3)
- 文件预览 (4)
- 发送邮件 (4)
- NIO (32)
- SPRING-DATA-JPA (7)
- EL表达式 (7)

```
@RequestMapping("/list")
public String actionList(Model model) {
    List<FlyActionsRunner> actionsRunners = actionSetService.selectAll();
    model.addAttribute("runners",actionsRunners);
    return "action/list";
}
```

返回的内容就是templates下面

最近有好久没有更新博客了，感谢小伙伴的默默支持，不知道是谁又打赏了我一个小红包，谢谢。

今天我们讲讲怎么在Spring Boot中使用模板引擎freemarker，先看看今天的大纲：

写道

- (1) freemarker介绍；
- (2) 新建spring-boot-freemarker工程；
- (3) 在pom.xml引入相关依赖；
- (4) 编写启动类；
- (5) 编写模板文件hello.ftl；
- (6) 编写访问类HelloController；
- (7) 测试；
- (8) freemarker配置；
- (9) freemarker常用语法；
- (10) freemarkerlayout

(1) freemarker介绍；

FreeMarker是一款模板引擎：即一种基于模板和要改变的数据，并用来生成输出文本（HTML网页、电子邮件、配置文件、源代码等）的通用工具。它不是面向最终用户的，而是一个Java类库，是一款程序员可以嵌入他们所开发产品的组件。

(2) 新建spring-boot-freeMarker工程；

我们新建一个maven工程，取名为：spring-boot-freeMarker

(3) 在pom.xml引入相关依赖；

这里使用freeMarker需要引入相关依赖包：spring-boot-starter-freeMarker，

```
<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>com.kfit</groupId>
    <artifactId>spring-boot-velocity</artifactId>
```

- OGNL表达式 (3)
- 读取配置文件 (5)
- 打印页面 (2)
- 下载文件 (3)
- SQL SERVER (5)
- DOM4J (7)
- Apusic金蝶 (4)
- 上传文件 (16)
- 其他 (2)
- 读写文件 (13)
- GOOGLE CHART (1)
- 匿名函数 (1)
- DBLINK (4)
- JACOB (1)
- IBATIS/MYBATIS (59)
- MAVEN (82)
- JAVA 打包 (3)
- AJAX (7)
- PUSHLET (7)
- COMET (5)
- 过滤器&监听器 (5)
- DWR (1)
- JVM (18)
- XSL (5)
- VELOCITY (2)
- LINUX (23)
- JMS (13)
- LDAP (1)
- JOTM (3)
- JTA (4)
- MongoDB&BSON (25)
- UML (2)
- LOGBACK (6)
- 安全方面 (11)
- 验证码 (1)
- CXF (7)
- JSTL (4)
- 短信猫 (1)
- ZIP解压缩 (4)
- BAT批处理命令 (1)
- RMI (1)
- JSR-VAILD (3)
- 自定义注解 (16)
- TILES (1)
- ITEXT-PDF (2)
- 命令行 (1)
- 进制转换 (5)
- IDEA (21)
- java 异常处理 (10)
- 泛型 (20)
- OSGI (6)
- GRADLE (3)
- EXCEL导出 (26)
- JACKSON (23)
- JUNIT (8)
- 反向代理 (1)
- BONECP (1)

```
<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>spring-boot-velocity</name>

<url>http://maven.apache.org</url>

<properties>

  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

  <!-- jdk版本号，angel在这里使用1.8，大家修改为大家本地配置的jdk版本号即可 -->

  <java.version>1.8</java.version>

</properties>

<!--

  spring boot 父节点依赖，

  引入这个之后相关的引入就不需要添加version配置，

  spring boot会自动选择最合适的版本进行添加。

-->

<parent>

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

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

  <version>1.4.1.RELEASE</version> <!-- 1.4.1.RELEASE , 1.3.3.RELEASE-->

</parent>

<dependencies>

  <dependency>

    <groupId>junit</groupId>

    <artifactId>junit</artifactId>

    <scope>test</scope>

  </dependency>

  <!-- spring boot web支持：mvc,aop... -->

  <dependency>

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

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

  </dependency>

  <!-- 引入freeMarker的依赖包. -->

  <dependency>

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

    <artifactId>spring-boot-starter-freeMarker</artifactId>
```

- JIT (2)
- GRAILS (3)
- GROOVY (10)
- WEBSOCKET (11)
- GIT (22)
- 调试工具 (2)
- MEMCACHE (10)
- 单元测试/压测工具 (4)
- GC (18)
- REDIS (38)
- HESSIAN (5)
- MYSQL (77)
- 微信开发 (9)
- ZOOKEEPER (39)
- HIVE (11)
- HBASE (10)
- 原子变量 (13)
- SPRING-BOOT (16)
- 爬虫 (2)
- XPP3 (1)
- ASPECTJ (1)
- 抽奖概率 (8)
- 深入Java集合学习系列 (13)
- 序列化 (7)
- FREEMARKER (13)
- SITEMESH (5)
- XML解析-Digester (4)
- 页面静态化 (5)
- NODEJS/GRUNT (20)
- DROOLS (1)
- 条形码/二维码 (9)
- HADOOP (2)
- LUCENE (6)
- JSOUP (15)
- NGINX (7)
- JSBRI (0)
- JSBRIDGE (2)
- 并发集合 (17)
- EMOJI (10)
- HTML5 (11)
- LOMBOK (1)
- SPRING工具类 (11)
- guava (9)
- JAVA 8 (2)
- 加解密 (18)
- EAR (3)
- 水印 (1)
- Thrift (13)
- Java并发包学习 (10)
- SonarQube (1)
- JAVA类加载 (3)
- REACT (19)
- SHIRO (4)
- SPEL (2)
- WEBDRIVER (2)
- PYTHON (3)
- 分布式锁 (7)

```
</dependency>

</dependencies>

</project>

(4) 编写启动类；

    启动类没有什么特别之处，不过多介绍，请看代码：

package com.kfit;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

/**
 *
 * @author Angel --守护天使
 * @version v.0.1
 * @date 2016年10月4日
 */
@SpringBootApplication

public class App {

    publicstaticvoid main(String[] args) {

        SpringApplication.run(App.class, args);

    }

}

(5) 编写模板文件hello.ftl;

    编写一个hello.ftl文件，此文件的路径在src/main/resources/templates（默认路径）下，其中hello.ftl文件的内容如下：

<html>
<body>

    welcome ${name} to freemarker!

</body>
</html>

(6) 编写访问类HelloController;

    有了模板文件之后，我们需要有个Controller控制类，能够访问到hello.ftl文件，这里也很简单，具体看如下代码：

package com.kfit.demo.web;

import java.util.Map;
```

Disruptor/LMAX (2)

AngularJS (2)

Vue.js (1)

Netty (50)

SOCKETIO (9)

消息MQ (2)

ProtoBuf (12)

TIF (0)

SPRING-MVC原理分析 (31)

ASM (2)

TestNG (5)

Mockito (3)

LBS排序 (3)

zookeeper&hbase (16)

IO多路复用 (5)

架构 (1)

cglib (4)

Epoll (0)

AIO (6)

Dubbo (1)

Jetty (10)

远程通讯 (1)

RPC (1)

SPI (4)

CORS跨域 (9)

中文相关 (7)

XPATH (3)

Aviator (1)

HtmlUnit (2)

埋点 (3)

FTP、SFTP (7)

Spring源码分析 (3)

Spring-ws (4)

Java对象引用 (8)

XStream (2)

JS原理 (1)

WEBPACK (5)

枚举 (5)

alibaba/fastjson (2)

动画效果 (3)

熔断降级 (2)

SpringMvc异常处理 (5)

SpringMvc实践 (2)

SSM/SSH (2)

ModelMapper (1)

ES6/7 (3)

Java线程专栏 (18)

线程中断 (9)

匿名内部类 (3)

MVCC (3)

Spring生命周期 (8)

ElasticSearch (11)

DB主从同步 (7)

JAVA7-8 (1)

Selenium (3)

SPRING-TEST (5)

RestTemplate (3)

```
import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

/**
 * 测试velocity;
 * @author Angel --守护天使
 * @version v.0.1
 * @date 2016年10月4日
 */

@Controller

public class HelloController {

    @RequestMapping("/hello")

    public String hello(Map<String,Object> map){

        map.put("name", "[Angel -- 守护天使]");

        return "hello";

    }

}
```

(7) 测试；

好了，到这里，我们就可以启动我们的程序进行测试了，访问地址：  
<http://127.0.0.1:8080/hello>，如果你在浏览器中看到如下信息：

welcome [Angel -- 守护天使] to freemarker!

那么说明你的demo ok 了。

(8) freemarker配置；

在spring boot的application.properties属性文件中为freemarker提供了一些常用的配置,如下：

```
#####
###FREEMARKER (FreeMarkerAutoConfiguration)
#####

spring.freemarker.allow-request-override=false

spring.freemarker.cache=true

spring.freemarker.check-template-location=true

spring.freemarker.charset=UTF-8

spring.freemarker.content-type=text/html

spring.freemarker.expose-request-attributes=false

spring.freemarker.expose-session-attributes=false
```

RedisTemplate	(6)
Mybatis 学习	(18)
数据库优化拆分	(13)
Spring schema	(1)
Spring 线程池	(3)
Spring 属性替换&转换	(7)
Spring 注入&注解&扫描	(14)
Spring Bean管理	(4)
Spring ( filter/listener/event )	(9)
Spring 事务	(14)
Spring 定时任务	(5)
Canvas	(4)
RequireJs	(2)
SPRING IOC源码	(1)
人脸/OPENCV/图像处理	(11)
SpringMVC上传文件	(13)
NO-SQL	(1)
大数据	(1)

文章存档	
2017年09月	(13)
2017年08月	(46)
2017年07月	(92)
2017年06月	(17)
2017年05月	(25)
展开	

阅读排行	
IntelliJ IDEA WEB项目的部署...	(132453)
SpringMVC——接收请求参...	(126049)
解决Setting property 'sourc...	(112930)
HttpClient post 请求实例	(80410)
SpringMVC数据绑定全面示...	(64580)
js中获取时间new Date()详细...	(61606)
[MyBatis]诡异的Invalid bou...	(60453)
SQL plus连接远程Oracle数据...	(58860)
HttpClient设置HTTP请求头H...	(55687)
读取Java文件到byte数组的三...	(53497)

评论排行	
解决Setting property 'sourc...	(51)
java实现附件预览 ( openoffi...	(47)
Spring MVC @Transactiona...	(15)
[MyBatis]诡异的Invalid bou...	(13)
SpringMVC——接收请求参...	(12)
IntelliJ IDEA WEB项目的部署...	(11)
Eclipse插件：Eclipse Color T...	(9)
interrupt、interrupted、isL...	(9)
Spring Data JPA入门	(8)
信息: TLD skipped. URI: http...	(7)

推荐文章	
* CSDN日报20170828——《4个方法快速打造你的阅读清单》	

```
spring.freemarker.expose-spring-macro-helpers=false
#spring.freemarker.prefix=
#spring.freemarker.request-context-attribute=
#spring.freemarker.settings.*=
#spring.freemarker.suffix=.ftl
#spring.freemarker.template-loader-path=classpath:/templates/ #comma-s
#spring.freemarker.view-names= # whitelist of view names that can be resc
```

(9) freemarker常用语法；

freemarker的语法并不是本节的重点，这里还是简单的介绍下几个常用的if else,list；

首先我们改造下HelloController的hello方法

@RequestMapping("/hello")

```
public String hello(Map<String,Object> map){
```

```
    map.put("name", "[Angel -- 守护天使]");
```

```
    map.put("gender",1);//gender:性别，1：男；0：女；
```

```
List<Map<String,Object>> friends =new ArrayList<Map<String,Object>>();
```

```
Map<String,Object> friend = new HashMap<String,Object>();
```

```
friend.put("name", "张三");
```

```
friend.put("age", 20);
```

```
friends.add(friend);
```

```
friend = new HashMap<String,Object>();
```

```
friend.put("name", "李四");
```

```
friend.put("age", 22);
```

```
friends.add(friend);
```

```
map.put("friends", friends);
```

```
return "hello";
```

```
}
```

这里我们返回了gender和friends的列表；

接下来我们看看怎么在freemarker进行展示呢？

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org"
    xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">
<head>
    <title>Hello World!</title>
</head>
<body>
    <p>
        welcome ${name} to freemarker!
```



```
<footer>

    This is a footer,welcome  ${name} to my web site!

</footer>

    修改hello.ftl :

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thym
    xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3

<head>

    <title>Hello World!</title>

</head>

<body>

    <#include "/header.ftl" >

    <p>

        welcome ${name} to freemarker!

    </p>

    <p>性别 :

        <#if gender==0>

            女

        <#elseif gender==1>

            男

        <#else>

            保密

        </#if>

    </p>

    <h4>我的好友 : </h4>

    <#list friends as item>

        姓名 : ${item.name} , 年龄${item.age}

        <br>

    </#list>

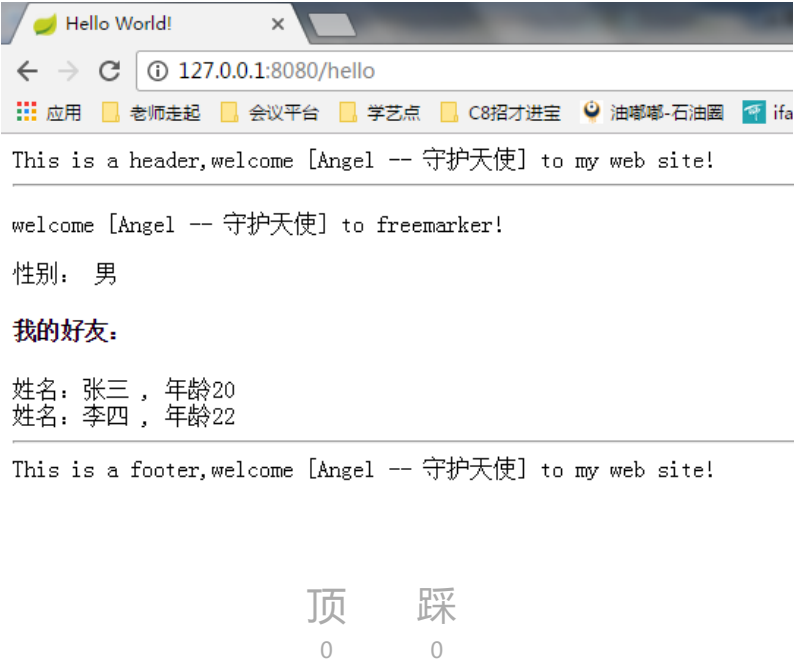
    <#include "/footer.ftl" >

</body>

</html>
```



到这里就ok了，我们访问/hello页面，应该会看到如下图的效果：



- [上一篇](#)    spring boot 使用FreeMarker模板
- [下一篇](#)    Springmvc RestTemplate 使用 && json xml rest 配置

相关文章推荐

- |  |  |
|--|--|
| • <a href="#">SpringBoot入门：添加FreeMarker支持</a>    | • <a href="#">freemarker---数字小数点格式化问题</a>                  |
| • <a href="#">SDCC 2017之大数据技术实战线上峰会</a>          | • <a href="#">史上最全Android基础知识总结</a>                        |
| • <a href="#">SpringMVC中Freemarker获取项目根目录</a>    | • <a href="#">FreeMarker宏macro的使用</a>                      |
| • <a href="#">Hadoop大数据从入门到精通知识体系详解</a>          | • <a href="#">机器学习需要的数学知识</a>                              |
| • <a href="#">freemarker macro(宏)的使用</a>         | • <a href="#">Spring Boot 集成 FreeMarker 详解案例</a>           |
| • <a href="#">SDCC 2017之区块链技术实战线上峰会</a>          | • <a href="#">springMvc整合Freemarker引入CSS,JS文件404...</a>    |
| • <a href="#">Freemaker &lt;#macro&gt;(宏)的使用</a> | • <a href="#">springboot整合freemarker</a>                   |
| • <a href="#">C++跨平台开发和ffmpeg，opencv音视频技术</a>    | • <a href="#">SpringBoot+Freemarker 项目ftl中使用include...</a> |

查看评论

暂无评论

发表评论

用户名：

评论内容：

提交

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机科技有限公司, 江苏乐知网络技术有限公司  
京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved