

# Spring Boot 整合 FreeMarker

作者: [mtkx](#)

原文链接: <https://ld246.com/article/1555682107033>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

FreeMarker 是一个非常古老的模板引擎，可以用在Web环境或者非Web环境中。与 Thymeleaf 不同 FreeMarker 需要经过解析才能够在浏览器中展示出来。FreeMarker 不仅可以用来配置HTML页面模，也可以作为电子邮件模板、配置文件模板以及源码模板等。Spring Boot 中对 FreeMarker整合也供了很好的支持，主要整合步骤如下：

## 1.新建Spring Boot工程，然后添加 `spring-boot-starter-web` 和 `spring-boot-starter-freemarker` 依赖，代码如下：

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-freemarker</artifactId>
</dependency>
```

## 2.配置FreeMarker，通过 `application.properties` 配置文件进行设置。代码如下：

```
#HttpServletRequest 的属性是否可以覆盖 controller 中 model 的同名项
```

```
spring.freemarker.allow-request-override=false
#Http Session 的属性是否可以覆盖 controller model 的同名项
spring.freemarker.allow-session-override=false
#是否开启缓存
spring.freemarker.cache=false
#模板文件编码
spring.freemarker.charset=UTF-8
#是否检查模板位置
spring.freemarker.check-template-location=true
#Content-Type的值
spring.freemarker.content-type=text/html
#是否将HttpServletRequest中的属性添加到Model中
spring.freemarker.expose-request-attributes=false
#是否将HttpSession中的属性添加到Model中
spring.freemarker.expose-session-attributes=false
#模板文件后缀
spring.freemarker.suffix=.ftl
#模板文件位置
spring.freemarker.template-loader-path=classpath:/templates/
```

## 3.创建 Book 实体类，然后在 Controller 中返回 ModelAndView ，代码如下：

```
public class Book {
  private Integer id;
  private String name;
  private String author;

  @Override
  public String toString() {
```

```

    return "Book{" +
        "id=" + id +
        ", name=" + name + "\" +
        ", author=" + author + "\" +
        '}';
}

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getAuthor() {
    return author;
}

public void setAuthor(String author) {
    this.author = author;
}

```

@Controller

```

public class BookController {
    @GetMapping("/books")
    public ModelAndView books(){
        List<Book> books=new ArrayList<>();
        Book b1=new Book();
        b1.setId(1);
        b1.setAuthor("罗贯中");
        b1.setName("三国演义");
        Book b2=new Book();
        b2.setId(2);
        b2.setAuthor("曹雪芹");
        b2.setName("红楼梦");
        books.add(b1);
        books.add(b2);

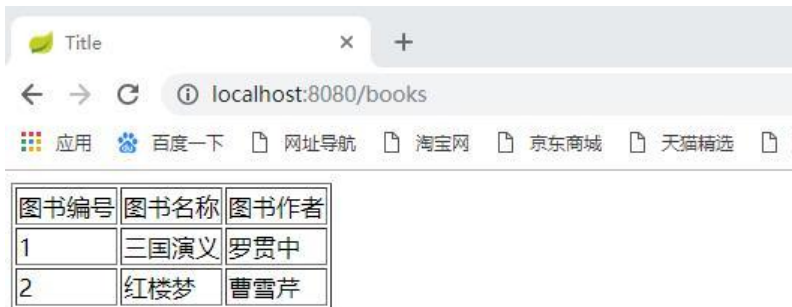
        ModelAndView mv=new ModelAndView();
        mv.addObject("books",books);
        mv.setViewName("books");
        return mv;
    }
}

```

#### 4.在 resources 目录下的 templates 录中创建 books.ftl 具体代码如下:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <table border="1">
    <tr>
      <td>图书编号</td>
      <td>图书名称</td>
      <td>图书作者</td>
    </tr>
    <#if books ??&&(books?size>0)>
    <#list books as book>
    <tr>
      <td>${book.id}</td>
      <td>${book.name}</td>
      <td>${book.author}</td>
    </tr>
    </#list>
    </#if>
  </table>
</body>
</html>
```

#### 5.运行项目，在浏览器中输入 <http://localhost:8080/books> 即可看到运行果，如图所示。



本文重点介绍 Spring Boot 整合 FreeMarker 并非 FreeMarker 的基础用法，关于 FreeMarker 的多资料，可以查看 <https://freeMarker.apache.org>