



链滴

# Spring Boot 整合 Thymeleaf

作者: [mtkx](#)

原文链接: <https://ld246.com/article/1555456568428>

来源网站: [链滴](#)

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

在目前的企业级应用开发中，前后端分离是主流，Spring Boot对视图层技术提供了很好的支持，官推荐使用的模板引擎是 Thymeleaf 不过像 FreeMarker 也支持，JSP技术不再推荐使用。

## 1.新建Spring Boot工程，然后添加 spring-boot-starter-web 和 spring-boot-starter-thymeleaf 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-thymeleaf</artifactId>
</dependency>
```

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

```
#是否开启缓存，开发时可设置为 false，默认为 true
spring.thymeleaf.cache=false
#检查模板是否存在，默认为 true
spring.thymeleaf.check-template=true
#检查模板位置是否存在，默认为 true
spring.thymeleaf.check-template-location=true
#模板文件编码
spring.thymeleaf.encoding=UTF-8
#模板文件位置
spring.thymeleaf.prefix=classpath:/templates/
#Content Type 配置
spring.thymeleaf.servlet.content-type=text/html
#模板文件后缀
spring.thymeleaf.suffix=.html
```

## 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.html 具体代码如下:

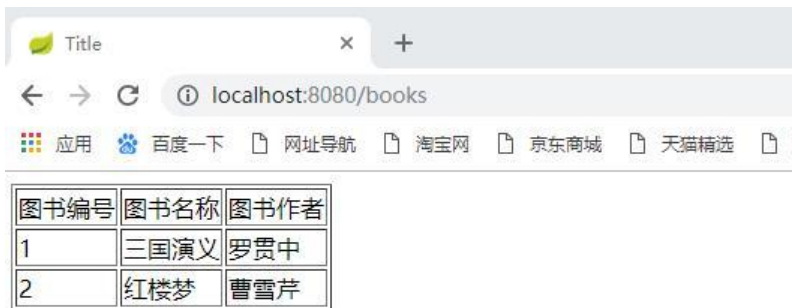
```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<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>
  <tr th:each="book:${books}">
    <td th:text="${book.id}"></td>
    <td th:text="${book.name}"></td>
    <td th:text="${book.author}"></td>
  </tr>
</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 Thymeleaf 并非 Thymeleaf 的基础用法，关于 Thymeleaf 的更多料，可以查看 <https://www.thymeleaf.org>