

WebJars 在 SpringBoot 中的使用

一、概要

简单来说，WebJars 就是将 web 依赖（js，css）放到 jar 文件中，然后通过 Maven 的形式依赖到项目中，避免手工下载这些依赖。从纯前端角度来说，依赖管理有 npm, Bower, 后端有 Maven, Gradle, WebJars 就是让 web 依赖能够运行在后端 Jvm 上的技术。

像 Bootstrap, jQuery, Angular JS, Chart.js 等都在 [webjars_官网](http://webjars.org) 上可以找到。

二、为什么使用 WebJars?

显而易见，因为简单。但不仅是依赖这么简单：

- 清晰的管理 web 依赖
- 通过 Maven, Gradle 等项目管理工具就可以下载 web 依赖
- 解决 web 组件中传递依赖的问题以及版本问题
- 页面依赖的版本自动检测功能

三、Maven 依赖

在 Maven 中通过下面的依赖就可以将 bootstrap 和 jquery 引入到项目中。

```
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>bootstrap</artifactId>
  <version>5.0.0-beta2</version>
</dependency>
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>jquery</artifactId>
  <version>3.6.0</version>
</dependency>
```

依赖导入了，这些 js 和 css 文件都在 classPath 路径下的 META-INF/resources/webjars 目录下。

四、自动检测依赖的版本

加入 webjars-locator 组件，这样就不需要在 html 添加依赖的时候填写版本。

```
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>webjars-locator</artifactId>
  <version>0.30</version>
</dependency>
```

之前是

```
<script src="/webjars/bootstrap/5.0.0-beta2/js/bootstrap.min.js"></script>
```

引入 webjars-locator 后是

```
<script src="/webjars/bootstrap/js/bootstrap.min.js">
</script>
```

注意：只能去掉版本号

五、前端页面使用

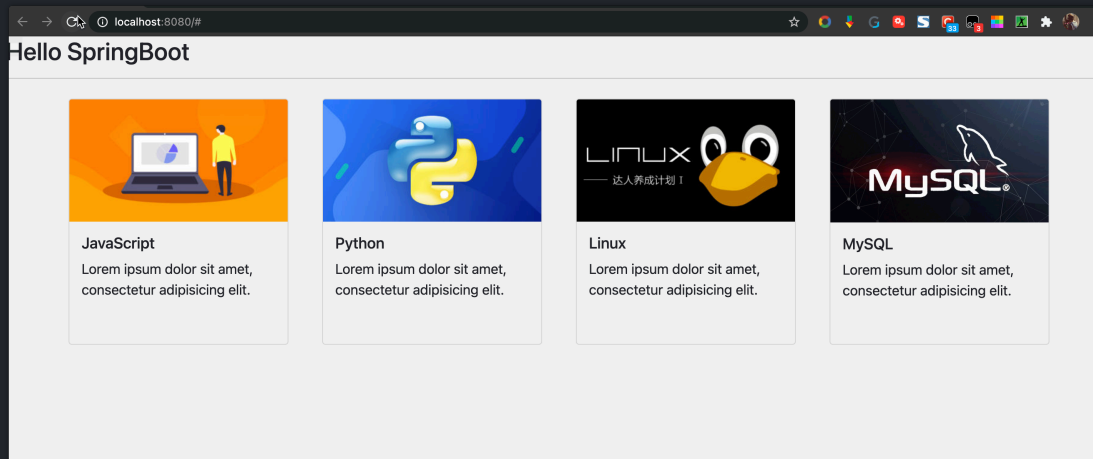
改造下之前的book.html 如下

bootstrap官网: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>

用它的布局和组件

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="UTF-8">
  <title>图书页面</title>
  <script src="/webjars/jquery/jquery.min.js"></script>
  <script src="/webjars/bootstrap/js/bootstrap.min.js">
</script>
  <link rel="stylesheet"
href="/webjars/bootstrap/css/bootstrap.min.css"/>
</head>
<body>
<h2 th:text="${message}">图书页面</h2>
<hr>
<div class="container">
  <div class="row">
    <div class="col-md-3" th:each="book:${bookList}">
      <div class="card">
        
        <div class="card-body">
          <h5 class="card-title"
th:text="${book.name}">Book title</h5>
          <p class="card-text">Lorem ipsum dolor
sit amet, consectetur adipisicing
elit.
          </p>
        </div>
      </div>
    </div>
  </div>
</div>
</body>
</html>
```

启动项目，访问 `http://localhost:8080/` 即可看到效果



SpringBoot 会自动将 classpath 下的 `/static` , `/public` , `/resources` , `/META-INF/resources` 这些目录自动映射到 `/**` 请求路径上。