# SpringBoot与web开发

## 创建过程

1. 创建SpringBoot应用，根据业务场景选中需要的模块，比如web,sql等。
2. SpringBoot已经将这些场景配置好了，只需要在配置文件中，指定少量配置就可以运行起来。
3. 自己编写业务代码。

关键核心在于弄清楚**自动配置原理：**

* **这个场景SpringBoot帮我们配置了什么，能不能修改，能修改哪些配置，能不能扩展？**
* **基本上都是\*\*\*AutoConfiguration：帮我们给容器中自动配置组件**
* **\*\*\*Properties：配置类来封装配置文件内容**

## SpringBoot对静态资源的映射规则

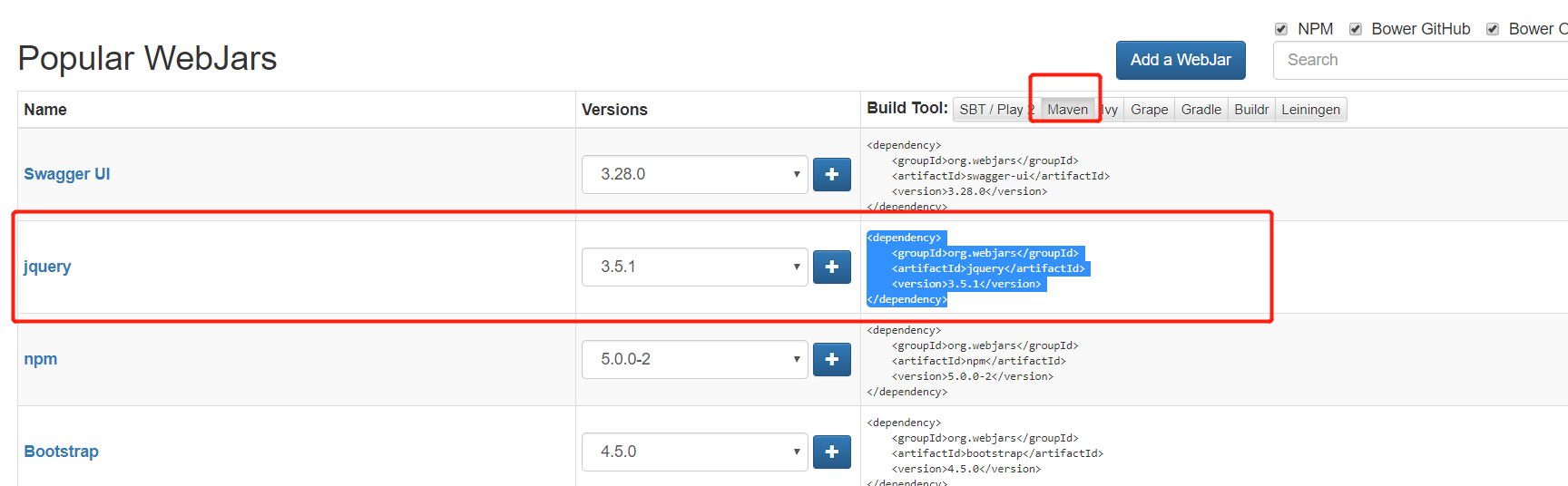
在WebMvcAutoConfiguration下的一个方法描述了映射规则：

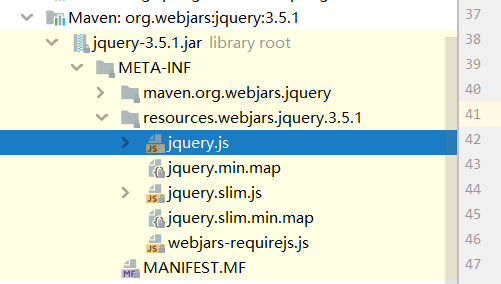
|  |
| --- |
| **public void** addResourceHandlers(ResourceHandlerRegistry registry) {  **if** (!**this**.resourceProperties.isAddMappings()) {  logger.debug(**"Default resource handling disabled"**);  } **else** {  Duration cachePeriod = **this**.resourceProperties.getCache().getPeriod();  CacheControl cacheControl = **this**.resourceProperties.getCache().getCachecontrol().toHttpCacheControl();  **if** (!registry.hasMappingForPattern(**"/webjars/\*\*"**)) {  **this**.customizeResourceHandlerRegistration(registry.addResourceHandler(**new** String[]{**"/webjars/\*\*"**}).addResourceLocations(**new** String[]{**"classpath:/META-INF/resources/webjars/"**}).setCachePeriod(**this**.getSeconds(cachePeriod)).setCacheControl(cacheControl));  }  String staticPathPattern = **this**.mvcProperties.getStaticPathPattern();  **if** (!registry.hasMappingForPattern(staticPathPattern)) {  **this**.customizeResourceHandlerRegistration(registry.addResourceHandler(**new** String[]{staticPathPattern}).addResourceLocations(WebMvcAutoConfiguration.getResourceLocations(**this**.resourceProperties.getStaticLocations())).setCachePeriod(**this**.getSeconds(cachePeriod)).setCacheControl(cacheControl));  }  } } |

1. 所有/webjars/\*\*，都去**classpath:/META-INF/resources/webjars/找资源。**

Webjars:以jar包的方式引入静态资源。Webjars网站：<https://www.webjars.org/>

例如引入jquery



如果访问jquery下的jquery.js资源，

则访问路径：

localhost:8080/webjars/jquery/3.5.1/jquery.js

1. “/\*\*”访问当前项目的任何资源

**"classpath:/META-INF/resources/"**,

**"classpath:/resources/"**,

**"classpath:/static/"**,

**"classpath:/public/",**

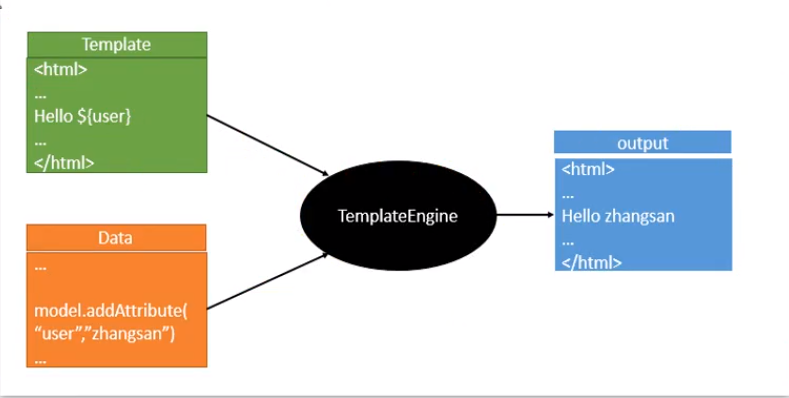
**"/"：当前项目的根路径**

1. 自定义静态文件夹下的路径

**spring.resources.static-locations**=**classpath:/hello/,classpath:/pcq/**

## 模板引擎

市面上有JSP、Velocity、Freemarker、Thymeleaf等。



页面以及数据交给模板引擎，组装好返回给浏览器。

### Thymeleaf的使用

1. 引入thymeleaf

|  |
| --- |
| <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-thymeleaf</**artifactId**> </**dependency**> |

版本可以在properties的标签中定义。

1. 使用方式

只要把html页面放在类路径templates下，thymeleaf就能自动渲染。

