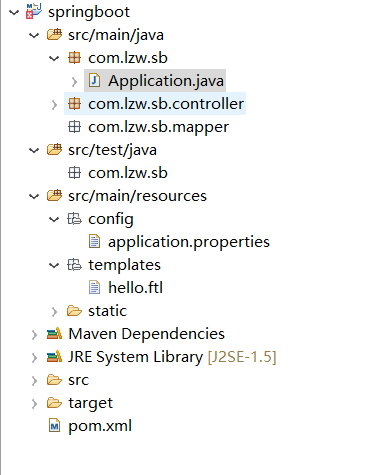
# 新建工程

## 项目结构



# 添加jar包

<parent>

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

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

<version>1.5.2.RELEASE</version>

</parent>

<properties>

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

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

</properties>

<dependencies>

<!-- 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>

</dependency>

<!-- 引入thymeleaf的依赖包. -->

<dependency>

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

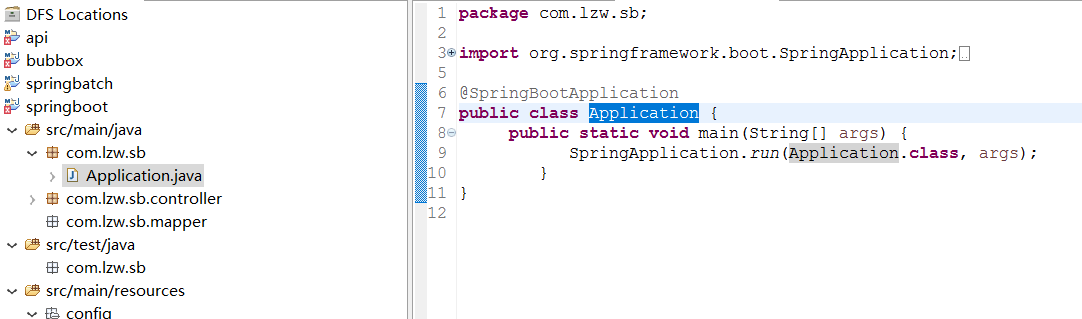
<artifactId>spring-boot-starter-thymeleaf</artifactId>

<version>1.5.1.RELEASE</version>

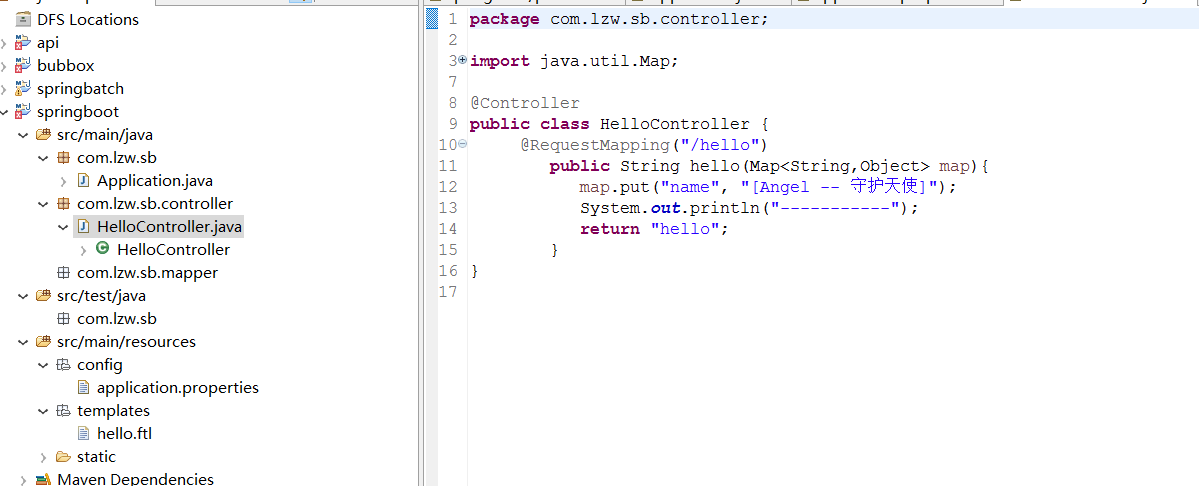
</dependency>

</dependencies>

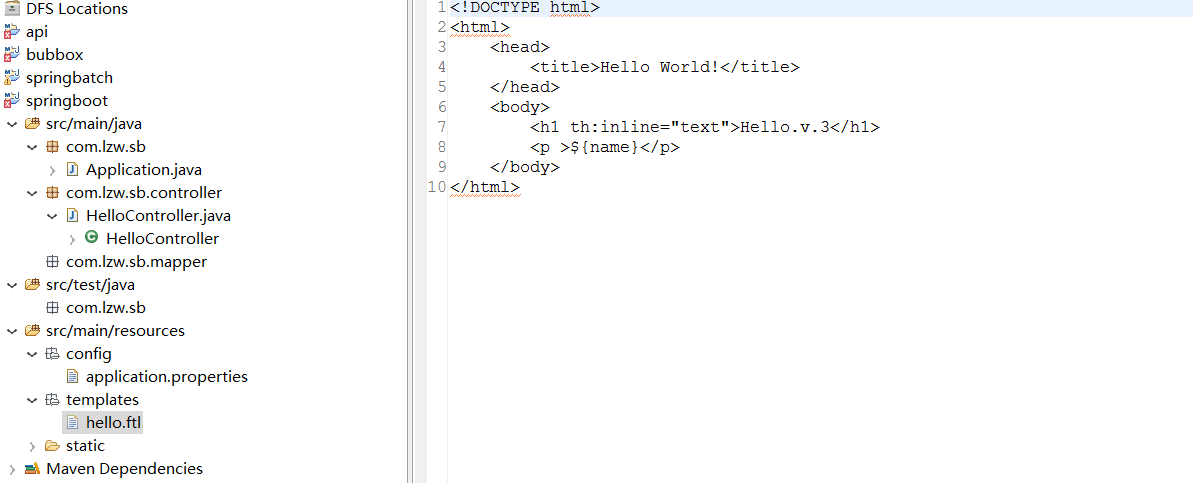
# 增加入口程序Application



# 新增controller



# 新增访问模板页面



Maven的资源文件目录：/src/java/resources   
spring-boot项目静态文件目录：/src/java/resources/static   
spring-boot项目模板文件目录：/src/java/resources/templates   
spring-boot静态首页的支持，即index.html放在以下目录结构会直接映射到应用的根目录下：

## thymeleaf默认的模板后缀为html

修改方式：

1. spring.thymeleaf.prefix=classpath:/templates/
2. spring.thymeleaf.suffix=.html
3. spring.thymeleaf.mode=HTML5
4. spring.thymeleaf.encoding=UTF-8
5. # ;charset=<encoding> is added
6. spring.thymeleaf.content-type=text/html
7. # set to false for hot refresh

HTML添加xmlns:th命名空间

<!DOCTYPE html>

<html xmlns=*"http://www.w3.org/1999/xhtml"* xmlns:th=*"http://www.thymeleaf.org"*>

<head>

<title>Hello World!</title>

</head>

<body>

<h1 th:inline=*"text"*>Hello2</h1>

<p th:text=*"${name}"*>13212</p>

</body>

</html>

thymeleaf常用基础知识点   
  
1、在html页面中引入thymeleaf命名空间，即<html xmlns:th=http://www.thymeleaf.org></html>，此时在html模板文件中动态的属性使用th:命名空间修饰   
  
2、引用静态资源文件，比如CSS和JS文件，语法格式为“@{}”，如@{/js/blog/blog.js}会引入/static目录下的/js/blog/blog.js文件   
  
3、访问spring-mvc中model的属性，语法格式为“${}”，如${user.id}可以获取model里的user对象的id属性   
  
4、循环，在html的标签中，加入th:each=“value:${list}”形式的属性，如<span th:each=”user:${users}”></span>可以迭代users的数据   
  
5、判断，在html标签中，加入th:if=”表达式”可以根据条件显示html元素   
<span th:if="${not #lists.isEmpty(blog.publishTime)}">   
<span id="publishtime" th:text="${#dates.format(blog.publishTime, 'yyyy-MM-dd HH:mm:ss')}"></span>   
</span>   
以上代码表示若blog.publishTime时间不为空，则显示时间   
  
6、时间的格式化，   
${#dates.format(blog.publishTime,'yyyy-MM-dd HH:mm:ss')}   
表示将时间格式化为”yyyy-MM-dd HH:mm:ss”格式化写法与Java格式化Date的写法是一致的。   
  
7、字符串拼接，有两种形式   
比如拼接这样一个URL:/blog/delete/{blogId}   
第一种：th:href="'/blog/delete/' + ${blog.id }"   
第二种：th:href="${'/blog/delete/' + blog.id }"

修改thymeleaf版本在properties添加

<thymeleaf.version>3.0.2.RELEASE</thymeleaf.version>

<thymeleaf-layout-dialect.version>2.0.4</thymeleaf-layout-dialect.version>

## 关于thymeleaf2.0和3.0也是有很大区别的：

##### 1.完整的HTML5 标记支持

Thymeleaf 3.0 不再是基于XML结构的。由于引入新的解析引擎，模板的内容格式不再需要严格遵守XML规范。即不在要求标签闭合，属性加引号等等。

##### 2.模板类型

Thymeleaf 3 移除了之前版本的模板类型，新的模板类型为：  
HTML  
XML  
TEXT  
JAVASCRIPT  
CSS  
RAW  
2个标记型模板(HTML和XML)，3个文本型模板(TEXT, JAVASCRIPT和CSS) 一个无操作(no-op)模板 (RAW)。  
HTML模板支持包括HTML5，HTML4和XHTML在内的所有类型的HTML标记。且不会检查标记是否完整闭合。此时，标记的作用范围按可能的最大化处理。

##### 3.片段（Fragment）表达式；

##### 4.无操作标记；

##### 5.模板逻辑解耦：Thymeleaf 3.0 允许 HTML和XML模式下的模板内容和控制逻辑完全解耦。

6.性能提示：  
7.不依赖于Servlet API；  
8.新的方言系统；  
9.重构了核心API;  
具体可见<http://www.tuicool.com/articles/ayeQ3qn>

## Freemarker默认的模板后缀为ftl