

<http://start.spring.io/>生成

生成的文件解压后

用eclipse导入

过程：Import->Existing Maven Projects-> Next -> 选择解压后的文件

SpringBoot的结构目录

src/main/java程序开发以及主程序入口

src/main/resources配置文件

src/test/java测试程序

SpringBoot建议的目录结构

```
1  com
2      -example
3      -myproject
4      -Application.java
5      -domain
6      -Customer.java
7      -CusromerRepository.java
8      -service
9      -CustomerService.java
10     -controller
11     -CustomerController.java
```

1、Application.java 建议放到跟目录下面,主要用于做一些框架配置

2、domain目录主要用于实体 (pojo) 与数据访问层 (Dao)

3、service 层主要是业务类代码

4、controller 负责页面访问控制

Spring的POM文件

默认文件模块：

spring-boot-starter：核心模块，包括自动配置支持、日志和YAML

spring-boot-starter-test : 测试模块，包括JUnit、Hamcrest、Mockito

引入web模块：

```
1 <dependency>
2     <groupId>org.springframework.boot</groupId>
3     <artifactId>spring-boot-starter-web</artifactId>
4 </dependency>
```

JSON风格

@RestController

代表该Controller标识的方法都要以JSON格式输出

TEST测试

打开的src/test/下的测试入口，编写简单的http请求来测试；使用mockmvc进行，利用MockMvcResultHandlers.print()打印出执行结果。

```
1 @RunWith(SpringJUnit4ClassRunner.class)
2 @SpringApplicationConfiguration(classes = MockServletContext.class)
3 @WebAppConfiguration public class HelloWorldControllerTests {
4     private MockMvc mvc;
5     @Before
6     public void setUp() throws Exception {
7         mvc = MockMvcBuilders.standaloneSetup(new
8         HelloWorldController()).build();
9     }
10    @Test
11    public void getHello() throws Exception {
12        mvc.perform(MockMvcRequestBuilders.get("/hello").accept(MediaType.APPLICATION_JSON))
13            .andExpect(MockMvcResultMatchers.status().isOk())
14            .andDo(MockMvcResultHandlers.print())
15            .andReturn();
16    }
17 }
```

热部署

服务器启动时，做修改后，服务器自动加载修改数据

开启方式在pom文件加入插件

```
1  <dependencies>
2      <dependency>
3          <groupId>org.springframework.boot</groupId>
4          <artifactId>spring-boot-devtools</artifactId>
5          <optional>true</optional>
6      </dependency>
7  </dependencies>
8  <build>
9      <plugins>
10         <plugin>
11             <groupId>org.springframework.boot</groupId>
12             <artifactId>spring-boot-maven-plugin</artifactId>
13             <configuration>
14                 <fork>true</fork>
15             </configuration>
16         </plugin>
17     </plugins>
18 </build>
```