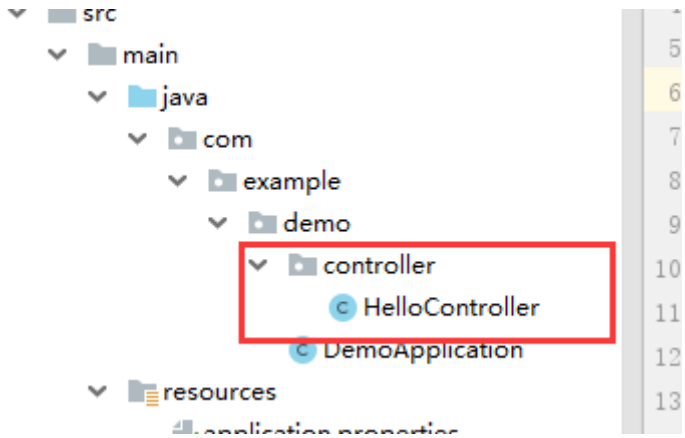
1. 新建 package



2. 新建 HelloController 类

```
@RestController
public class HelloController {
    @RequestMapping("/hello")
    public String index(){
        return "Hello World";
    }
}
```

3. 启动该应用，通过浏览器访问：http://localhost:8080/hello

编写单元测试

```
package com.example.demo;
```

```
import com.example.demo.controller.HelloController;
import org.junit.Before;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.boot.SpringBootConfiguration;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.http.MediaType;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
import org.springframework.test.context.junit4.SpringRunner;
import org.springframework.test.context.web.WebAppConfiguration;
import org.springframework.test.web.servlet.MockMvc;
import org.springframework.test.web.servlet.MockMvcBuilder;
import org.springframework.test.web.servlet.request.MockMvcRequestBuilders;
import org.springframework.test.web.servlet.setup.MockMvcBuilders;

import static org.hamcrest.Matchers.equalTo;
import static
org.springframework.test.web.servlet.result.MockMvcResultMatchers.content;
import static
org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;

@RunWith(SpringJUnit4ClassRunner.class)
@SpringBootTest
@WebAppConfiguration
public class DemoApplicationTests {

   private MockMvc mockMvc;

   @Before
   public void setup(){
      mockMvc = MockMvcBuilders.standaloneSetup(new
HelloController()).build();
   }
   @Test
   public void hello() throws Exception {
      mockMvc.perform(MockMvcRequestBuilders.get("/hello")
            .accept(MediaType.APPLICATION_JSON))
            .andExpect(status().isOk())
            .andExpect(content().string(equalTo("Hello World")));
   }
}
```

- @RunWith(SpringJUnit4ClassRunner.class)：引入 Spring 对 JUnit4 的支持
- @WebAppConfiguration：开启 Web 应用配置，用于模拟 SerlvetContext。
- MockMvc 对象：用于模拟调用 Controller 的接口发起请求，再@Test 定义的 hello 测试用例中，perform 函数执行一次请求调用，accept 用于执行接收的数据类型，andExcept 用于判断接口返回的期望值。
- @Before：JUnit 中定义再测试用例@Test 内容执行前预加载的内容，这里用来初始化对 HelloController 的模拟

注意引入下面的静态引用，让 status、content、equalTo 函数可用

```
import static org.hamcrest.Matchers.equalTo;
import static
org.springframework.test.web.servlet.result.MockMvcResultMatchers.content;
import static
org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;
```