MockMvc Spring boot project

在src/test/java 下建一个包

@RunWith(SpringJUnit4ClassRunner.**class**)

@WebAppConfiguration

@ContextConfiguration

**public** **class** AppTests {

**private** MockMvc mockMvc;

//测试接口

@Before

**public** **void** setup() {

//MockitoAnnotations.initMocks(this);

//this.mockMvc = MockMvcBuilders.webAppContextSetup(this.wac).dispatchOptions(true).build();

**this**.mockMvc = MockMvcBuilders.*standaloneSetup*(**new** dycs()).build();//获取mockMvc实例

}

@Test

**public** **void** simple() **throws** Exception {

//测试VIEW

mockMvc.perform(*get*("/SetData"))//执行一个RequestBuilder请求

.andExpect(*view*().name("test"))//验证返回视图是否是"index2"

.andExpect(*status*().isOk())

.andDo(*print*()); //打印结果到控制台

//测试接口

mockMvc.perform(*get*("/SetData1"))//执行一个RequestBuilder请求

.andExpect(*status*().isOk())

.andDo(*print*()); //打印结果到控制台

}

}

测试class

@Controller

**public** **class** dycs {

//测试view

@RequestMapping("/SetData")

**public** String SetData() {

**return** "test";

}

//测试接口

@RequestMapping("/SetData1")

@ResponseBody

**public** String SetData1(HttpServletRequest request) {

**return** "888888888888"+request.getParameter("name");

}

}

//测试控制器返回指定的视图

1. pom.xml

<!--添加对tomcat的支持-->

<dependency>

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

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

<scope>provided</scope>

</dependency>

<!--对jsp的支持-->

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

</dependency>

2.application.properties

spring.mvc.view.prefix=/Pages/

spring.mvc.view.suffix=.jsp