Unit Test

注：测试主要的注解是@mock,@InjectMocks。

然后MockitoAnnotations.initMocks(this);

可以在@Before方法里面初始化需要用到的对象，或者自定义方法初始化对象，要用到时调用即可。

when(Class.method(parameter)).thReturn(Object);

// 断言输出的对象的值是否与预期一致

// import static org.assertj.core.api.Java6Assertions.assertThat;

assertThat(Object. property). isEqualTo(Value);

// 断言调用的方法次数是否与预期一致

// import static org.mockito.Mockito.\*;

verify(Class, times(int)).method(parameter);

Controller unit test:

主要区别：

mockMvc.perform(get("/admin/company?scacCode=PSHY")

.contentType(MediaType.APPLICATION\_ JSON)) // 数据的格式

.andExpect(status().isOk()) // 返回的状态是200

.andDo(print()) // 打印出请求和相应的内容

.andExpect(jsonPath("$.scacCode",is("PSHY"))); // 返回的json串的scacCode属性值为PSHY

如果要求传入的参数是@ResponseBody，则需要传入json串，还需要把内容放在body里面，用.content()放入：

mockMvc.perform(post("/comparison/fail/reason")

.content("{}")

.accept(MediaType.APPLICATION\_JSON)

.contentType(MediaType.APPLICATION\_JSON))

.andDo(print())

.andExpect(status().isOk());

或者：

//设计一个json格式的对象进行赋值

ObjectMapper mapper = new ObjectMapper();

ObjectWriter ow = mapper.writer().withDefaultPrettyPrinter();

String requestJson = ow.writeValueAsString(browserSceneLog);

mockMvc.perform(post("/browser-scene-log?browserSceneLog=" + requestJson)

.content(requestJson)

.contentType(MediaType.APPLICATION\_JSON))

.andDo(print())

.andExpect(status().isOk())

.andExpect(jsonPath("$.type", is("Typec")));

实例：

@RestController

@RequestMapping("/admin")

public class AdminController {

private final CarrierService carrierService;

@Autowired

public AdminController(CarrierService carrierService) {

this.carrierService = carrierService;

}

@GetMapping("/company")

public CarrierDTO getCompanyByCode(@RequestParam(value = "code") String code) {

//直接调用Service

return carrierService.getCarrierData(code);

}

}

public class AdminControllerTest {

@Mock

private CarrierService carrierService;

@InjectMocks

private AdminController adminController;

@Before

public void setUp() throws Exception {

MockitoAnnotations.initMocks(this);

}

//命名should\_\_when\_\_given\_

@Test

public void should\_return\_result\_of\_success\_and\_verify\_success\_when\_call\_getCompanyByScacCode\_given\_scacCode() throws Exception {

//given

String scacCode = "PSHY";

CarrierDTO carrierDTO = new CarrierDTO("PSHY", "POSHCN77692625276132","P O Shipping");

//当触发方法并且传的参数一致时，返回thenReturn的值

when(carrierService.getOneCarrierData(anyString())).thenReturn(carrierDTO);

//when,mock发送request请求

mockMvc.perform(get("/admin/company?scacCode=PSHY")

.contentType(MediaType.APPLICATION\_FORM\_URLENCODED)) // 数据的格式

.andExpect(status().isOk()) // 返回的状态是200

.andDo(print()) // 打印出请求和相应的内容

.andExpect(jsonPath("$.scacCode",is("PSHY"))); // 返回的json串的scacCode属性值为PSHY

//then

// carrierService.getOneCarrierData方法调用一次

verify(carrierService, times(1)).getOneCarrierData(scacCode);

}

}