Spring Boot 进阶

单元测试

课程内容

- 为什么要做单元测试？现状
- 工具：
  - Junit4
  - Spring-boot-starter-test
  - Mockito

为什么要做单元测试？现状

- 不知道怎么编写单元测试
- 项目没有要求，所以不编写
- 单元测试价值不高，完全是浪费时间
- 业务逻辑比较简单，不值得编写单元测试
- 不管怎样，集成测试将会抓住所有的 bug，用不着进行单元测试
- 在项目的前期还是尽量去编写单元测试，但是越到项目的后期就越失控
- 为了完成编码任务，没有足够的时间编写单元测试

为什么要做单元测试？现状

- 在联调前对代码有信心
- 出现的问题都在单测中覆盖避免重复犯错，这样每次修改都有信心
- 避免新人之后的更改无意中搞一些事情，对别人的更改也有信心
- 在重构时候有信心
- 总而言之就是**有信心**改代码

工具：Junit4

- 注解的使用
  - @Test
  - @Before @After
  - @BeforeClass @AfterClass

工具：spring-boot-starter-test

- 注解的使用
  - @RunWith(SpringRunner.class)
  - @SpringBootTest
- 断言：对结果进行验证
  - Assert

模拟HTTP环境

- 注解：@AutoConfigureMockMvc
- 工具类：MockMvc（3步）
  - perform：执行一个RequestBuilder请求，会自动执行SpringMVC的流程并映射到相应的控制器执行处理；
    - - MockMvcRequestBuilders提供了get/post/put/**delete**/upload等http请求的方式
    - - 提供了header/contentType/cookie/characterEncoding/params等设置request参数的方式

模拟HTTP环境

- 工具类：MockMvc（3步）
  - ResultActions
    - - andExpect：添加ResultMatcher验证规则，验证控制器执行完成后结果是否正确；
    - - andDo：添加ResultHandler结果处理器，比如调试时打印结果到控制台；
    - - andReturn：最后返回相应的MvcResult；然后进行自定义验证/进行下一步的异步处理；
  - MvcResult（自定义Assert）
    - - getModelAndView：获得控制层设置的ModeAndView对象
    - - getResponse：获得最终响应结果

工具：Mockito 例子

- 实现功能：用户登录
  - 验证（void）**其他同事开发**
  - 读取用户信息（通过id读取用户信息）
  - 读取用户发布过的文章（通过用户名读取文章列表）**其他同事开发**
  - 返回：用户信息和文章列表

工具：Mockito

- 打桩：实际上就是对接口、类、方法、参数、返回值进行伪造或者模拟。
  - ReflectionTestUtils
- 注解的使用
  - @Mock：真实对象的替代品
  - @Spy：被测试对象需要一部分被执行、一部分被mock，需要用spy对目标对象进行包装

thenReturn与doReturn差异

- 语法：
  - when-thenReturn
    - when(obj.method()).thenReturn(returnValue)
  - doReturn-when
    - doReturn(returnValue).when(obj).method()
- 差异

API	Mock obj	Spy obj
when-thenReturn	×	○
doReturn-when	×	×

总结

- Don't Repeat Yourself ！

代码地址： git@gitee.com:fairyside/springboot.git