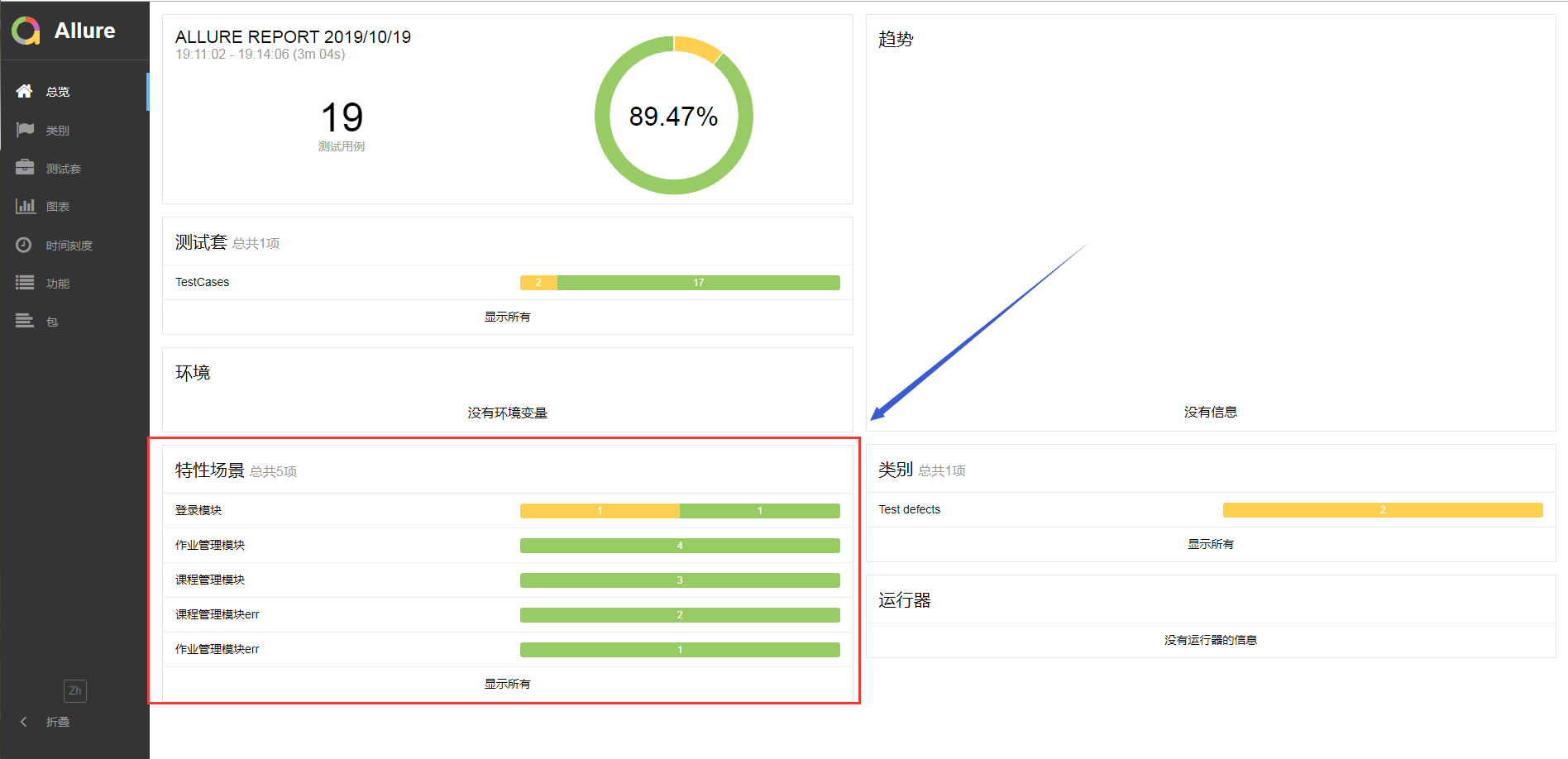
**一、定制化后的效果展示**

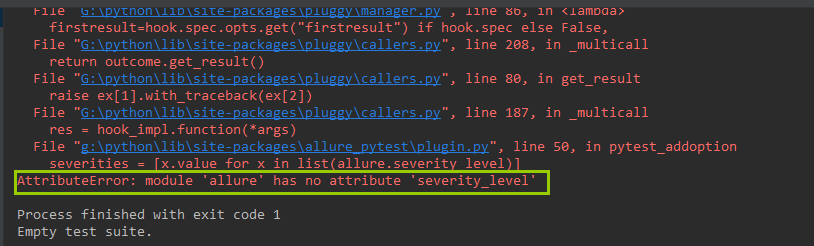
　　用两张图展示效果：



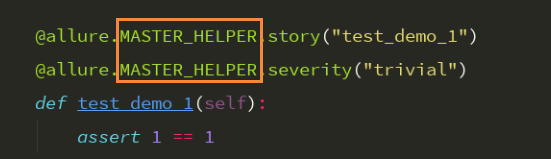


**二、注意别踩坑**

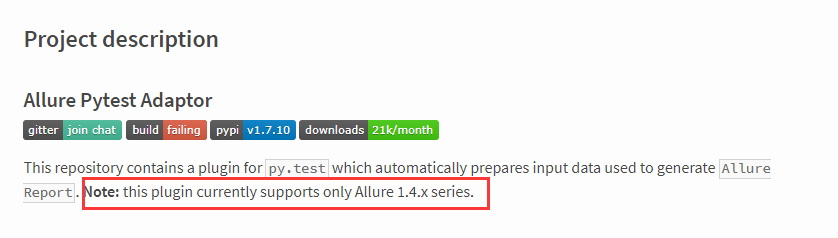
　　allure定制化想必大部分情况都会去选择pip install pytest-allure-adaptor这个插件，安装完成后，运行定制化代码出现以下报错：



 　　安装了pytest-allure-adaptor插件的定制代码像这样的：



 　　我们会使用allure.MASTER\_HELPER下的方法去进行定制，通过查看该插件的官网<https://pypi.org/project/pytest-allure-adaptor/>，发现版本不匹配，因此抛出了异常



 　　解决办法：

* + pytest-allure-adaptor版本过低不兼容当前版本的allure，选择**不安装该插件**
  + 使用allure下的方法去进行定制

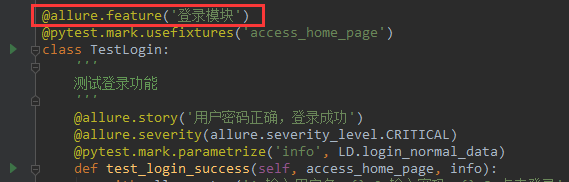
**三、定制化内容**

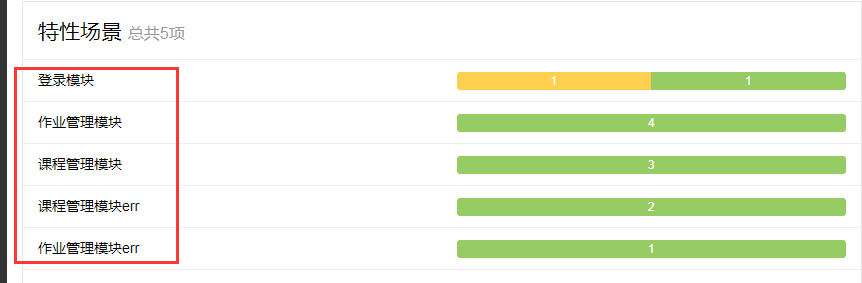
**使用前，先引入allure模块**

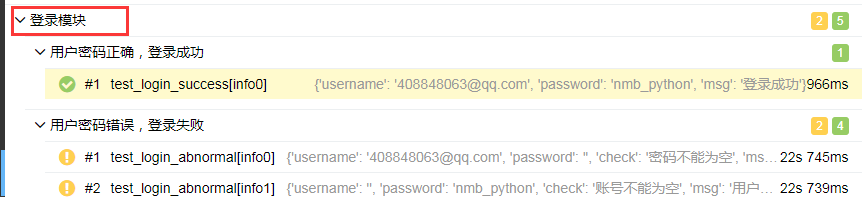
import allure

**1.feature——测试用例特性（主要功能模块）**

　　使用方法：@allure.feature()

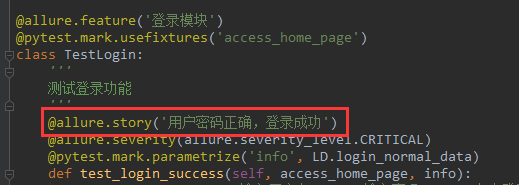


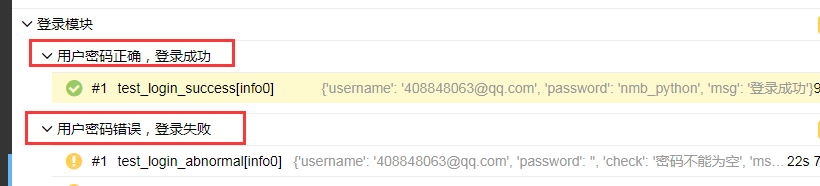




**2.story——feature功能模块下的分支功能**

　　使用方法：@allure.story()



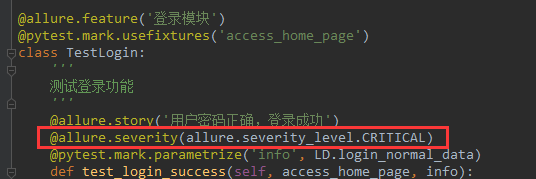


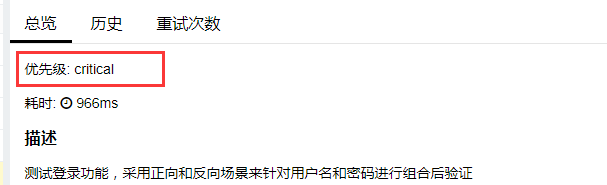
**3.severity——测试用例的严重级别**

　　Allure中对严重级别的定义：

* + blocker级别：中断缺陷（客户端程序无响应，无法执行下一步操作）
  + critical级别：临界缺陷（ 功能点缺失）
  + normal级别：普通缺陷（数值计算错误）
  + minor级别：次要缺陷（界面错误与UI需求不符）
  + trivial级别：轻微缺陷（必输项无提示，或者提示不规范）

　　使用方法：@allure.severity(allure.severity\_level.CRITICAL)  或者  @allure.severity('critical')



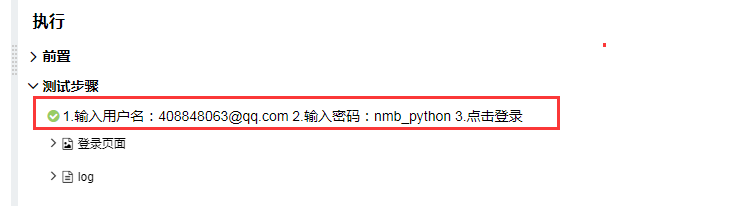


**4.step——测试用例的步骤**

使用方法：

1.@allure.step()  只能以装饰器的形式放在类或者方法上面

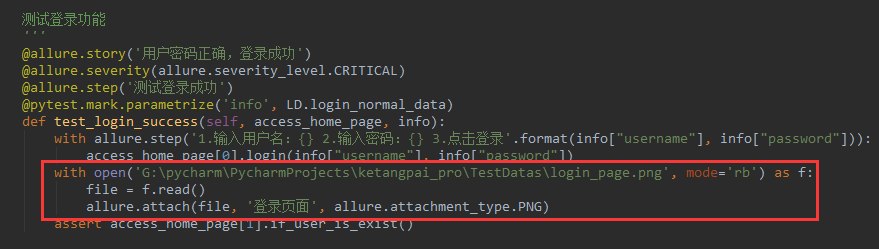
2.with allure.step():  可以放在测试用例方法里面，但测试步骤的代码需要被该语句包含

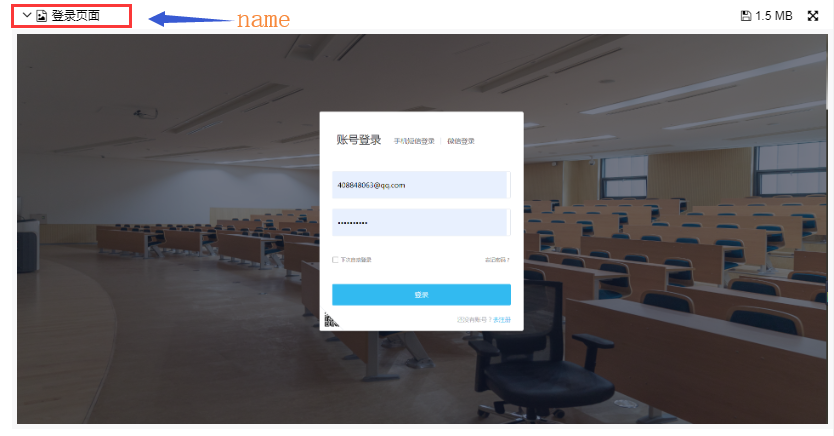


**5.attach——用于向测试报告中输入一些附加的信息，通常是一些测试数据信息**

　　使用方法：allure.attach(body, name, attachment\_type, extension)

* + body- 要写入文件的原始内容
  + name- 包含文件名的字符串
  + attachment\_type- 其中一个allure.attachment\_type值
  + extension- 提供的将用作创建文件的扩展名



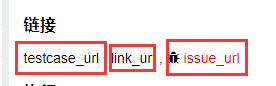


**6.link/issue/testcase——链接**

　　使用方法：

* + @allure.link()
  + @allure.issue()
  + @allure.testcase()





**7.description——用例描述**

　　使用方法：

* + @allure.description()  提供描述字符串的装饰器
  + @allure.description\_html()  提供一些HTML在测试用例的描述部分



