

0327课堂笔记

1. git工作流程

在公司里，很可能你们的项目是由其他人进行创建的，那么作为项目参与者，我们需要从git仓库上拉取代码到自己本地，并做修改增加等等动作

1.拉取仓库代码

```
git clone https://gitee.com/sandroad/apiautotest0320.git
```

2.本地分支

通常情况下，我们在自己的本地分支上进行处理，所以要先创建分支

```
git checkout -b shamo
# 查看分支，星号标准的就是你现在所在的分支
git branch
```

在分支上针对代码进行修改，修改完成后，和本地的master进行合并

```
# 切换并且合并分支到master上
git checkout master
# 添加到缓存区
git add conftest.py
# 提交到本地
git commit -m "conftest.py增加注释"
```

3.提交代码到远程

提交时，是有可能会和远程仓库的代码产生冲突的，因为项目是多人协作，有可能别人也会去修改代码，

所以我们的流程是先拉取远程仓库的master和本地master进行合并，合并后再提交到远程

合并时，有可能别人和我们操作的是同一行代码，此时就会冲突，那么需要手动解决冲突，然后再提交

```
62         delete_goods_api((url, goods_id)).send()
63
64     @pytest.fixture(scope='session', autouse=True)
65     def redis_util():
66         redis_info = load_yaml_file('/config/redis.yml')['mtxshop']
67         redis_util = RedisUtil(host=redis_info['host'], pwd=redis_info['password'])
68         yield redis_util
69
70     <<<<<< HEAD
71     #这是数据库连接的fixture
72
73     # 这是别人增加的注释
74     >>>>>> 38b163b432db6154f09d9788730ce356458c984d
75     @pytest.fixture(scope='session', autouse=True)
76     def db_util():
77         db_info = load_yaml_file('/config/db.yml')['mtxshop']
78         db_util = DBUtil(host=db_info['host'], user=db_info['username'], password=db_
79         yield db_util
80         db_util.close()
81
82
```

这是我增加的

这是别人增加的

```
#这是数据库连接的fixture 这是别人增加的注释 将冲突的地方改成这样
@pytest.fixture(scope='session',autouse=True)
def db_util():
    db_info = load_yaml_file('/config/db.yml')['mtxshop']
    db_util = DB_Util(host=db_info['host'], user=db_info['username'], password=db_info['password'])
    yield db_util
    db_util.close()
```

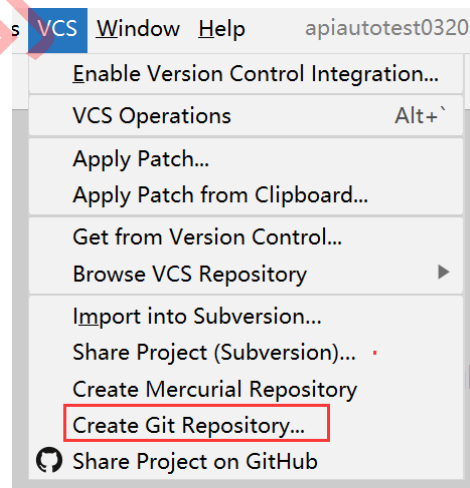
冲突解决后，在自己本地进行代码调试，调试通过后，然后再提交

```
# 添加到缓存区
git add conftest.py
# 提交到本地
git commit -m "冲突已解决"
# 提交到远程
git push
```

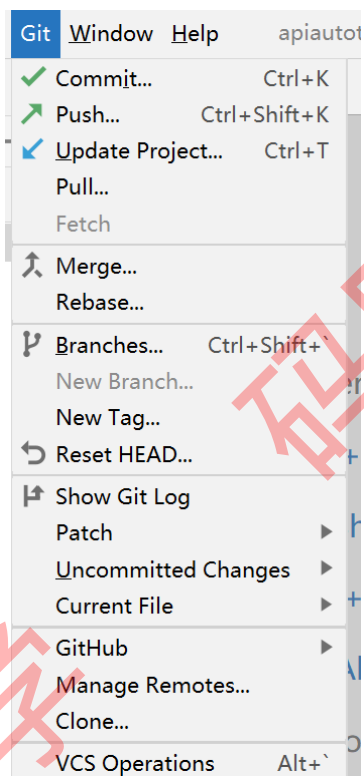
2.pycharm git可视化操作

找一个不在git版本控制的项目来学习，

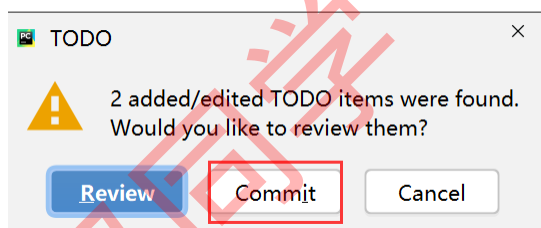
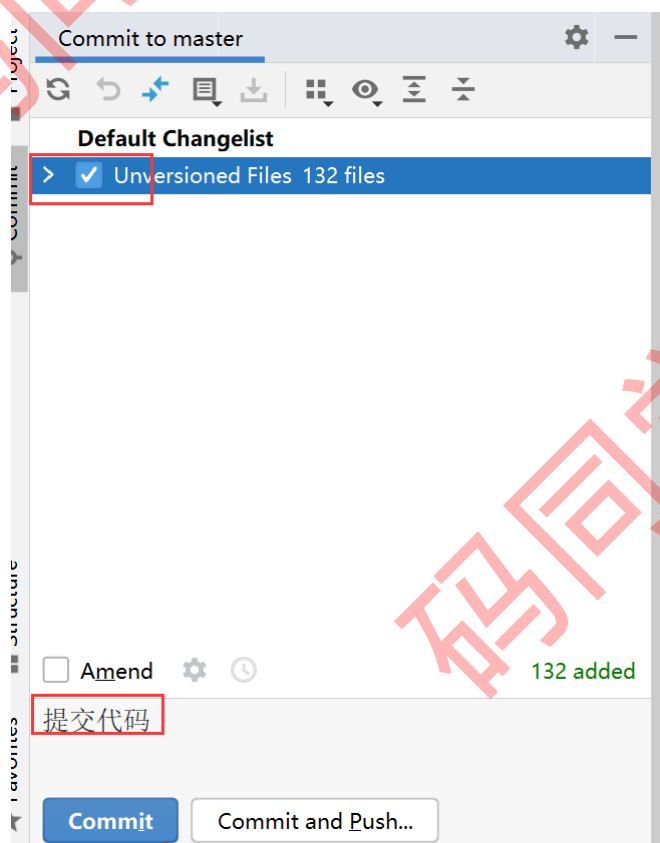
- 创建git版本仓库



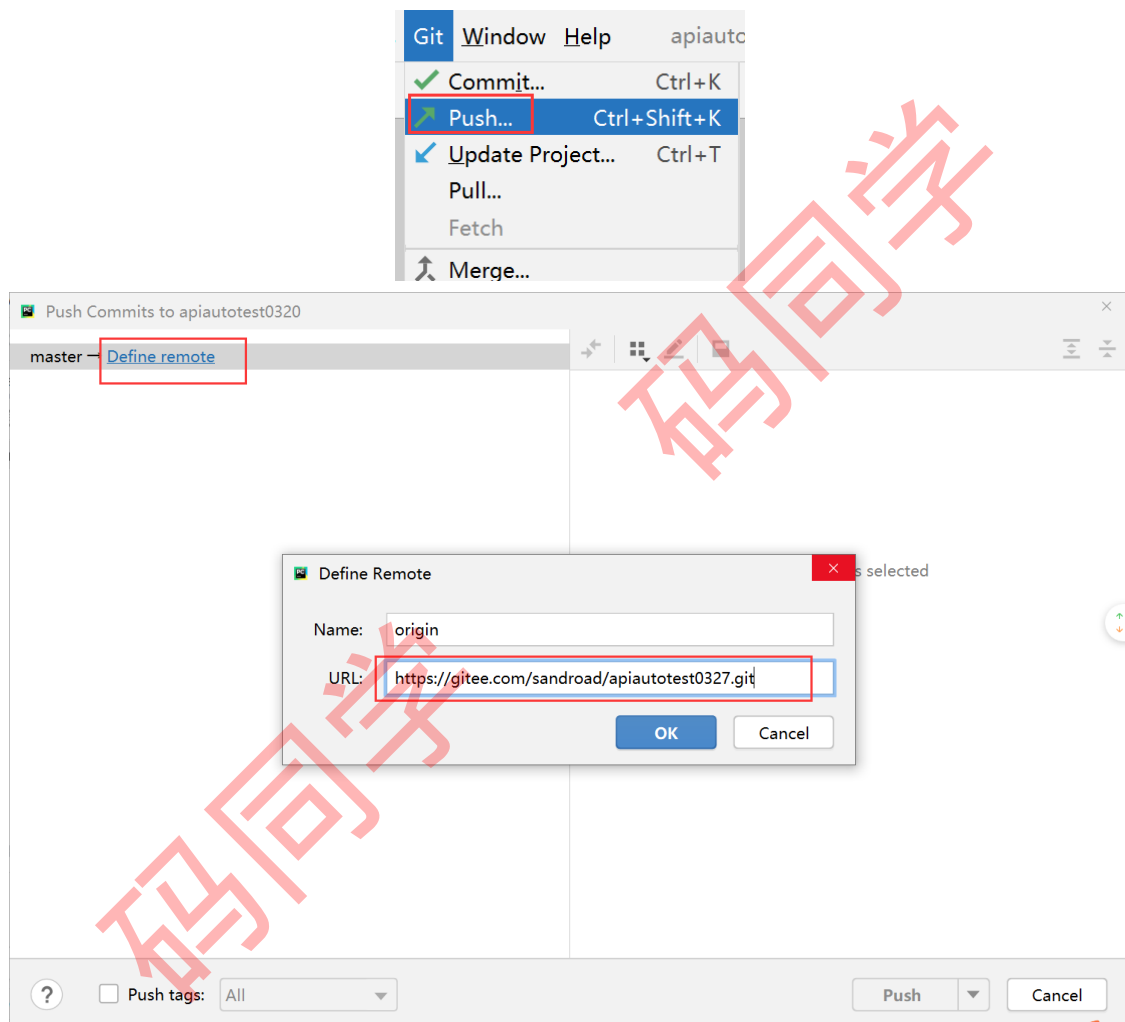
- 提交本地代码
选择commit



增加注释，选择要提交的内容



- 提交远程代码



点OK，点Push，代码提交到远程

- 修改代码进行提交

修改代码之后，点git-->commit，进行提交到本地，然后再点git-->push，提交到远程

3.git忽略文件

通常我们在本地调试时，会产生一些本地文件，比如日志文件、测试报告文件，所以我们可以给git项目增加一个配置文件，该文件中描述不需要提交的某些目录或者文件

增加一个.gitignore 文件，内容写上要忽略的文件目录或者文件

```
# Created by pytest automatically.
.pytest_cache
report
.gitignore
logs
```

- 更多的常用命令

```
git branch 查看本地所有分支
git status 查看当前状态
git commit 提交
git branch -a 查看所有的分支
git branch -r 查看远程所有分支
git commit -am "init" 提交并且加注释
git remote add origin git@192.168.1.119:ndshow
git push origin master 将文件给推到服务器上
```

```

git remote show origin 显示远程库origin里的资源
git push origin master:develop
git push origin master:hb-dev 将本地库与服务器上的库进行关联
git checkout --track origin/dev 切换到远程dev分支
git branch -D master develop 删除本地库develop
git checkout -b dev 建立一个新的本地分支dev
git merge origin/dev 将分支dev与当前分支进行合并
git checkout dev 切换到本地dev分支
git remote show 查看远程库
git add .
git rm 文件名(包括路径) 从git中删除指定文件
git clone git://github.com/schacon/grit.git 从服务器上将代码给拉下来
git config --list 看所有用户
git ls-files 看已经被提交的
git rm [file name] 删除一个文件
git commit -a 提交当前repos的所有的改变
git add [file name] 添加一个文件到git index
git commit -v 当你用-v参数的时候可以看commit的差异
git commit -m "This is the message describing the commit" 添加commit信息
git commit -a -a是代表add, 把所有的change加到git index里然后再commit
git commit -a -v 一般提交命令
git log 看你commit的日志
git diff 查看尚未暂存的更新
git rm a.a 移除文件(从暂存区和工作区中删除)
git rm --cached a.a 移除文件(只从暂存区中删除)
git commit -m "remove" 移除文件(从Git中删除)
git rm -f a.a 强行移除修改后文件(从暂存区和工作区中删除)
git diff --cached 或 $ git diff --staged 查看尚未提交的更新
git stash push 将文件给push到一个临时空间中
git stash pop 将文件从临时空间pop下来

-----

git remote add origin git@github.com:username/Hello-world.git
git push origin master 将本地项目给提交到服务器中

-----

git pull 本地与服务器端同步

-----

git push (远程仓库名) (分支名) 将本地分支推送到服务器上去。
git push origin serverfix:awesomebranch

-----

git fetch 相当于是从远程获取最新版本到本地, 不会自动merge
git commit -a -m "log_message" (-a是提交所有改动, -m是加入log信息) 本地修改同步至服务器端 :
git branch branch_0.1 master 从主分支master创建branch_0.1分支
git branch -m branch_0.1 branch_1.0 将branch_0.1重命名为branch_1.0
git checkout branch_1.0/master 切换到branch_1.0/master分支
du -hs

git branch 删除远程branch
git push origin :branch_remote_name
git branch -r -d branch_remote_name

```

4.jenkins搭建

jenkins安装的形式本身有很多种, 咱们采用war包进行部署

下载群建jenkins.war以后, 在d盘下创建jenkins目录, 将war包放进来

如果你是mac, 那么就找一个目录, 把war包放进去

```
java -DJENKINS_HOME=D:\jenkins\jenkins_data -jar D:\jenkins\jenkins.war
```

D:\jenkins\jenkins_data 表示jenkins所有的数据最终会存在该目录下

D:\jenkins\jenkins.war war包的路径

每次粘贴命令比较麻烦，那么可以写一个脚本文件

windows下创建一个txt文件，将启动命令复制进去，然后修改文件后缀为bat，每次启动只需要双击该文件即可

mac下创建一个文本文件，将启动命令复制进去，然后修改文件后缀为sh，启动时终端执行sh xxxx.sh

- 邮件通知注意点

邮件通知

SMTP服务器

smtp.qq.com

用户默认邮件后缀

@qq.com

☒ 使用SMTP认证

用户名

2879897713@qq.com

密码

Concealed

☐ 使用SSL协议

☐ Use TLS

有可能你这里和我一样，你这里可能添加凭据的一个下拉框，那么就点击添加凭据，在凭据中填入邮箱和授权码，然后在这里选择即

5.普通任务配置注意点




- 测试报告无需登录即可查看

测试报告默认情况下需要经过登录才能查看，我们设置一下，不需要登录也可以查看

系统管理--》全局安全配置

授权策略

- ☐ Anyone can do anything
- ☐ Legacy mode
- ☐ Logged-in users can do anything
- ☐ 安全矩阵
- ☒ 项目矩阵授权策略

用户/组	Overall	凭据		Agent				Job				Run	View		SCM	Lockable Resources										
	Administer	Read	Create	Delete	Manage Domains	Update	View	Build	Configure	Connect	Create	Delete	Discover	Move	Read	Workspace	Delete	Replay	Update	Configure	Create	Delete	Read	Tag	Reserve	Unlock
 Anonymous Users	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 Authenticated Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 admin	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Add user or group...

- git拉取之后的代码存储

从git拉取代码之后会将其存在jenkins数据目录中的workspace目录下

6.mock测试

7.selenium环境搭建

- 安装selenium

```
# windows
pip install selenium==3.141.0
# mac
python3 -m pip install selenium==3.141.0
```

- chromedriver下载及配置

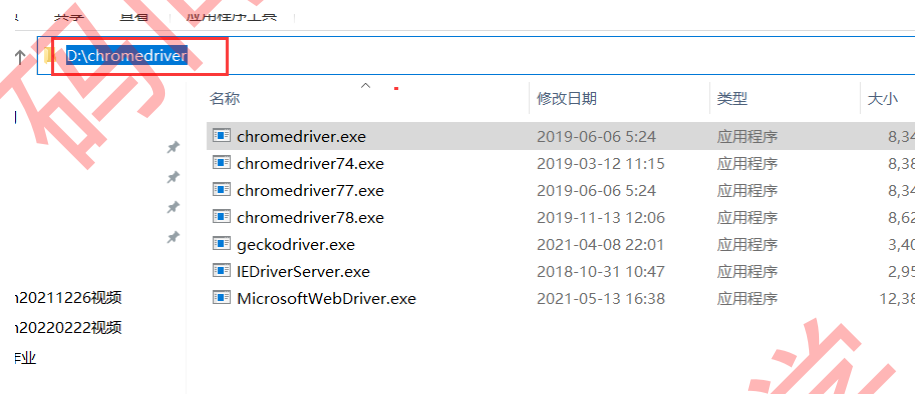
必须根据自己的chrome版本去找对应的



Index of /chromedriver/77.0.3865.40/

Name	Last modified	Size
Parent Directory	-	-
chromedriver_linux64.zip	2019-08-20T18:02:46.906Z	5.17MB
chromedriver_mac64.zip	2019-08-20T18:02:48.324Z	7.05MB
chromedriver_win32.zip	2019-08-20T18:02:49.818Z	4.54MB
notes.txt	2019-08-20T18:02:53.355Z	3.11KB

下载以后放在某个目录下，进行解压，解压后配置环境变量，每个人的路径都是不一样的



加好以后，打开命令行，直接敲chromedriver，回车

```
D:\1码同学\付费课\mock测试>chromedriver
Starting ChromeDriver 76.0.3809.12 (220b19a666554bdcac56dff9f944c300842c933-refs/branch-heads/3809@{#83}) on port 9515.
Only local connections are allowed.
Please protect ports used by ChromeDriver and related test frameworks to prevent access by malicious code.
```

能启动一个服务，看到版本号和下载的一致就可以

如果版本不一致，有可能你之前配置过chromedriver，所以找到之前配置的环境变量，将其删除

- 创建项目

8.前端基础

浏览器页面是由html源码、js、css

```
<!doctype html>
<html data-n-head>
  <head>...</head>
  <body data-n-head class>
    <div id="__nuxt">
      <div class="nuxt-progress" style="width:0%;height:2px;background-color:#29d;opacity:0;"></div> == $0
      <div id="__layout">
        <div id="app">
          <div id="shortcut" data-v-33b2dcbe>...</div>
          <div id="header" data-v-1a9bb128>...</div>
          <div id="wrapper" class="container" data-v-2a183b29>
            <div class="search-bar-container" style="position:fixed;top:-160px;" data-v-331cdfca data-v-2a1
            </div>
            <div class="focus-container" data-v-2a183b29>...</div>
            <!-->
            <div class="floor-container" data-v-2a183b29>...</div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

9.css定位规则

- class属性定位
后跟的class属性的值
当class属性值中有空格时，空格需要用.代替
- id属性定位
#后跟id属性的值
- 其他属性定位
属性是用[]括起来，[xxx="xxxx"]
- 属性的模糊匹配
 - 以某个属性的xxx开头 button[class^="search"]
 - 以某个属性的xxx结尾 button[class\$="goods"]
 - 某个属性包含xxx button[class*="btn"]
- 同级兄弟元素
当我们发现目标元素没有什么特征时，可以看看他同级的上面的元素有没有特征，如果有就可以通过大哥找小弟
.search-input + button
- 使用伪类定位元素
同级元素按照顺序定位
.nav-list>li:last-child

10.xpath定位

- 路径
绝对路径不常用，我们通常是使用相对路径//，以相对路径开始
- 属性定位
//ul[@class="search-hot-keywords"]
多属性同时定位
//button[@type="button" and @class="search-btn goods"]
- 按照顺序找元素
找匹配到的元素的所有中的第一个，相当于找高三年假全级第一
(//li)[1]

找高三的8个班，每个班的第一名

```
//li[1]
```

找最后一个

```
//li[last()]
```

找倒数第2

```
//li[last()-1]
```

找前三个

```
//li[position()<4]
```

- 通过元素文本找

```
//button[text()='搜商品']
```

- 属性包含xxx或者文本包含xxx

```
//button[contains(@class,"goods")] 找属性class包含goods的button标签
```

```
//button[contains(text(),"商品")] 找文本内容包含商品的button标签
```

- *代表任意

```
//*[contains(text(),"我的购物车")] 不管什么标签，只要你的文字包含了我的购物车，就会命中
```

- xpath轴

主要应用在比较复杂元素定位上，特别是当你的目标元素没有任何特征信息时

- 找同级元素的下方元素

```
//input[@class="search-input"]/following-sibling::button[2]
```

- 找同级元素的上方元素

```
//em[@id="cart-num"]/preceding-sibling::span
```

- 返回上一级查找

```
//input[@class="search-input"]/..
```

11.定位如何选择

在企业中通常情况可能会有规定使用统一的定位方式，css或者xpath

css和xpath区别

1. css执行效率比xpath快，相同的元素定位1000次，css会比xpath快十几秒

css定位是按照css自身的节点去定位，不会扫描整个html源码，xpath每次定位都会扫描整个html源码

2. xpath定位比css更加强大

xpath定位时可上可下，但是css只能向下

对于css和xpath实际上都可以在开发者工具中通过右键拷贝拿到表达式，虽然可以拷贝，但是学会了基本规则后可以更加灵活的去处理元素定位