**简答题批改页面**

**技术文档**

**张燕妮**

**整理于：2015.12.20**

目 录

[第一章 简答题批改页面概述 2](#_Toc438393068)

[1.1批改页面功能及操作简介 2](#_Toc438393069)

[1.2 批改页面脚本调用关系 3](#_Toc438393070)

[第二章 系统环境与文件概览 6](#_Toc438393071)

[2.1 系统环境 6](#_Toc438393072)

[2.2 文件概览 6](#_Toc438393073)

[第三章 Edx与gitlab数据同步的实现 7](#_Toc438393074)

[2.1 edx与gitlab 建立ssh连接 7](#_Toc438393075)

[2.2 ssh 方式从gitlab clone仓库到edx服务器 10](#_Toc438393076)

[2.3 编写git pull和git push的脚本 11](#_Toc438393077)

[2.4 xblock中调用脚本 13](#_Toc438393078)

## 第一章 简答题批改页面概述

### 1.1批改页面功能及操作简介

简答题批改页面作为open edx的一个课件组件，教师能够利用该组件实现简答题的批量批改。该组件能够自动生成待批改的简答题列表（已批改过的将不会显示），当教师在下拉菜单中选中某个要批改的简答题题号，即可动态生成与该题号对应的所有已提交答案，教师可一次性批改与该题号对应的所有学生提交的答案，点击保存按钮便会自动保存批改结果，并把结果同步到gitlab仓库。

批改的过程可以随时中断，下次批改时只需再次点击“生成待批改页面”按钮，即可重新生成批改页面，上一次已批改过的题不会重复出现。

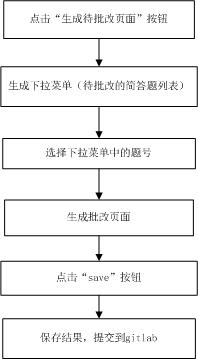


图1－1是简答题批改页面的操作流程图。

图1－1 简答题批改页面操作流程图

### 批改页面脚本调用关系

#### 生成待批改列表

当在edx前台点击“生成待批改页面”按钮时，触发grade.js中的$(“#generateList”,element).click()点击事件, 调用edx后台grade.py中的generate\_list()，这是一个json\_handler。

接着，Generate\_list()通过os.system调用pullFromGitlab.sh脚本，实现从gitlab仓库pull最新的学生提交的答案仓库到edx 服务器（保存在”/var/www/zyni/answer”）。

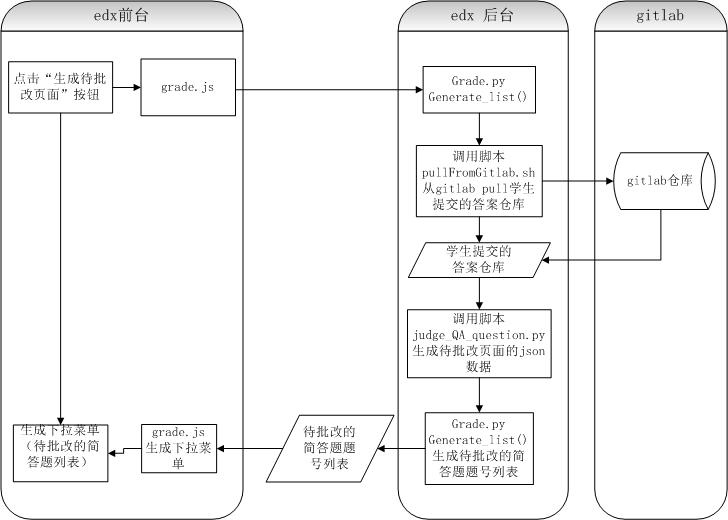
然后，调用judge\_QA\_question.py脚本，实现生成简答题待批改页面的json数据，该数据用于批改页面html的生成，按题号保存json数据（保存在”/var/www/zyni/grade\_xblock\_data/”）。

接下来，Generate\_list()通过遍历/var/www/zyni/grade\_xblock\_data/生成待批改的题号列表。作为返回结果，传递给edx前台。

最后，edx前台由grade.js中的generateOption()方法生成下拉菜单。

上述过程用流程图表示如图1－2所示：

图1－2 生成下拉菜单流程图

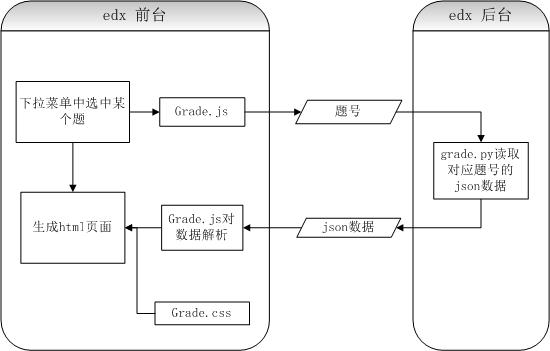


#### 生成html页面

选中1.2.1生成的下拉菜单中的某一个题号时，调用grade.js中的clickSelect()方法，向edx后台传递参数{“value”:题号}。

edx后台，grade.py中的select\_click()接收参数，根据题号从"/var/www/zyni/grade\_xblock\_data/judge\_QA\_question/"读取对应的json数据，返回给edx 前台。

edx前台，调用grade.js中的selseCallBack(result)方法，根据接收到的json数据，动态生成html页面。结合grade.css对html页面进行渲染。



上述过程用流程图表示如图1-3所示：

图1－3 生成html页面流程图

#### 保存批改结果

通过1.2.2中生成的批改页面可以进行批改。点击”save”按钮，实现把批改结果push到gitlab.具体的调用关系如下。

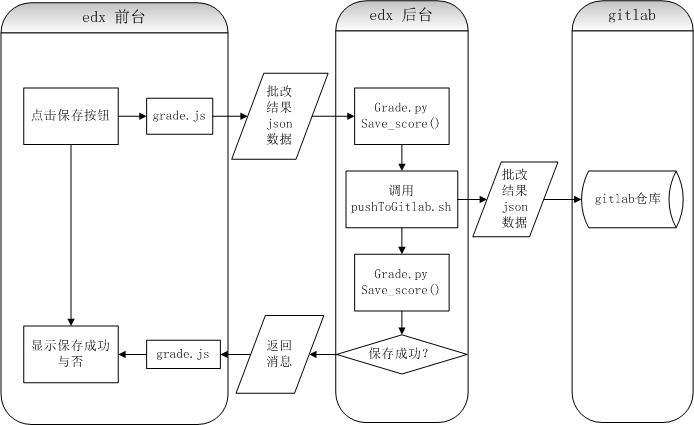
点击“save”按钮，调用grade.js中的click\_btn()方法，向edx后台传送数据，数据格式为{"user\_name":user\_name ,"email":email,"q\_number":q\_number,"score":score}。

edx后台，由grade.py中的save\_score()接收数据，按email建立目录，保存对应题号的批改结果。保存在” /var/www/zyni/answer/score/”目录下。

然后，调用“pushToGitlab.sh”脚本，把结果push到gitlab。

上述过程用流程图表示如图1-4所示：

图1-4 保存批改结果流程图



## 第二章 系统环境与文件概览

### 2.1 系统环境

Openedx平台布署于IBM提供的linux环境中。

表1. 相关访问链接表

|  |  |  |
| --- | --- | --- |
| 类别 | 说明 | 地址 |
| IBM Ubuntu | webshell | https://crl.ptopenlab.com:8800/webshell/XjWEfv67al5HsZUF/ |
| Open edx | 前台 | http://crl.ptopenlab.com:8811/ |
|  | studio | http://crl.ptopenlab.com:18010/ |
| gitlab | 学生提交的答案仓库 | http://172.16.13.236/ |

### 2.2 文件概览

表2. 文件名与实现的功能对应关系表

|  |  |  |
| --- | --- | --- |
| 类别 | 文件名 | 功能 |
| xblock | grade.py | 实现edx后台操作功能 |
| grade.js | 实现edx前台渲染功能 |
| grade.html | edx前台html页面框架 |
| grade.css | 实现edx前台渲染功能 |
| lib\_log.py | 日志脚本 |
| 脚本 | pullFromGitlab.sh | 从gitlab仓库pull到edx服务器 |
| pushToGitlab.sh | edx服务器push到gitlab仓库 |
| judge\_QA\_question.py | 生成待批改页面的json数据源 |

表3. 文件存储位置表

|  |  |  |
| --- | --- | --- |
| 类别 | 目录名/文件名 | 存储位置 |
| xblock | grade\_xblock | /home/zyni/grade\_xblock |
| 脚本 | script | /var/www/zyni/script |
| 日志 | grade\_xblock.log | /var/www/zyni/grade\_xblock.log |
| 数据 | grade\_xblock\_data | /var/www/zyni/grade\_xblock\_data |
| answer | /var/www/answer |

## 第三章 Edx与gitlab数据同步的实现

### 2.1 edx与gitlab 建立ssh连接

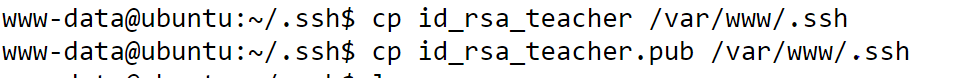
具体实现步骤如下：

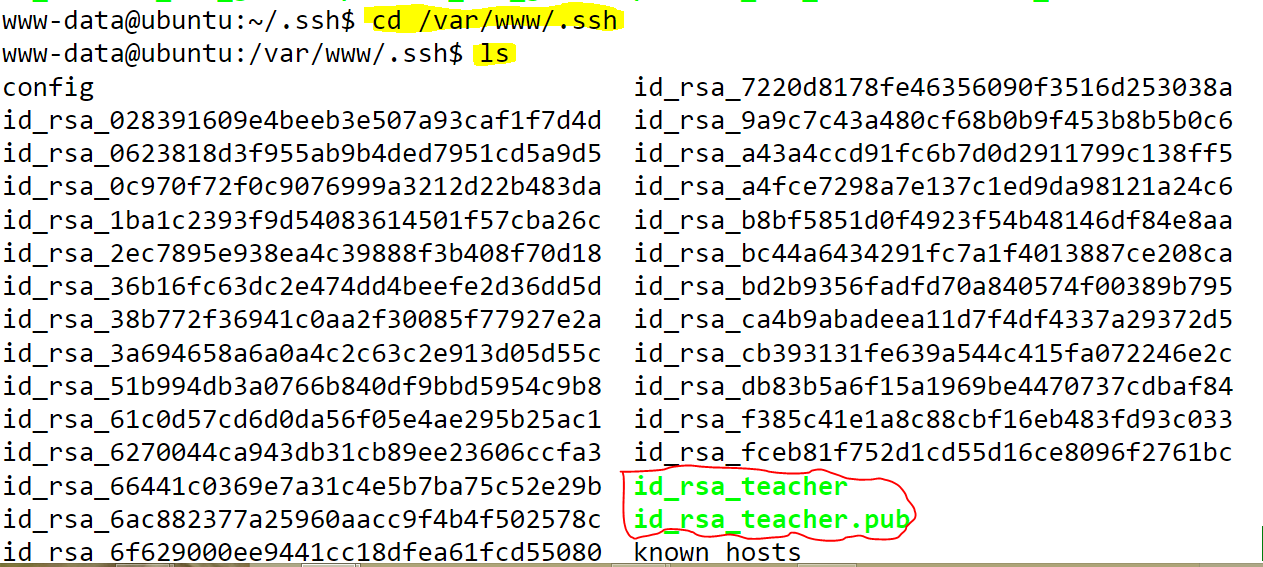
**（1）切换用户到edx。**

**（2） 生成密钥：**

密钥生成方法见：http://blog.csdn.net/jenyzhang/article/details/44703937

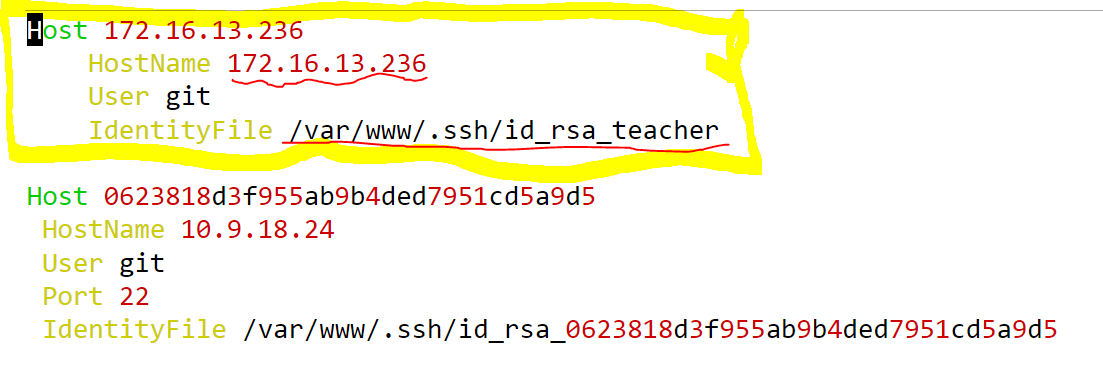
这里只需把在zyni用户下已生成的密钥拷贝到/var/www/.ssh目录下即可

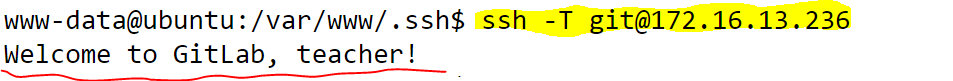
**① 把在zyni用户下已生成的密钥拷贝到/var/www/.ssh目录下**

**②进入到/var/www/.ssh 可以看到id\_rsa\_teacher和id\_rsa\_teacher.pub已被成功拷贝**

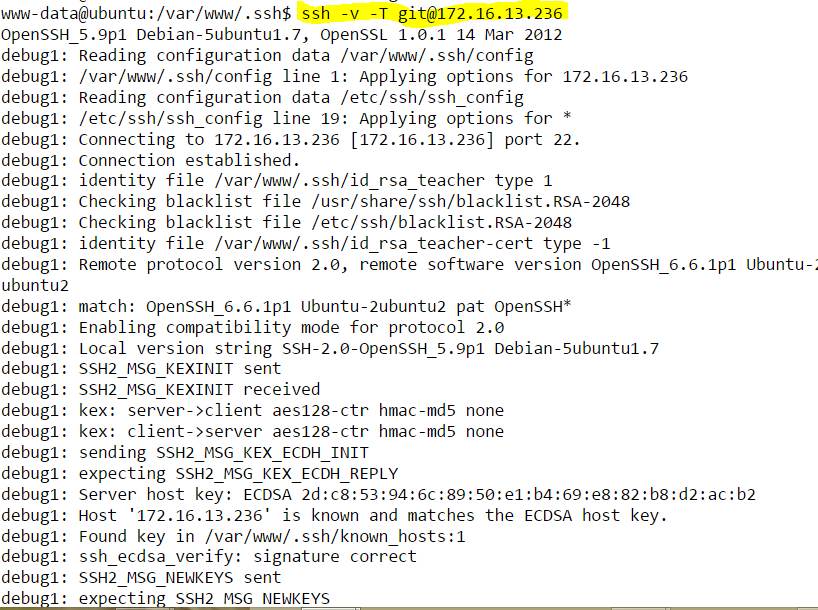
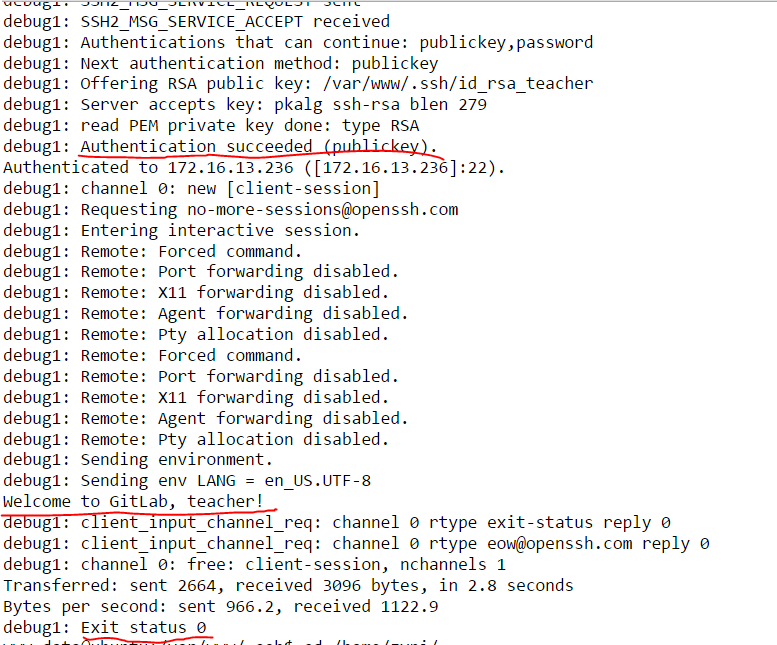
**③ 更改config文件**

输入命令“vim config”

添加如下黄色框内的内容：（:wq 保存更改）

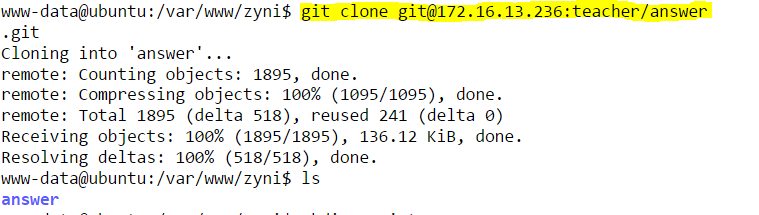
**④ 测试是否成功**

注：如果没有显示上述红线部分的内容（Welcome to gitlab, teacher!）则说明没有成功建立ssh连接。可通过“ssh -v -T [git@172.16.13.236](mailto:git@172.16.13.236)”来调试查看具体的出错.正确的连接应该会出现下面的内容，可以看到有“Authentication succeeded”的提示。如果没有，请检查你的public key是否正确的复制到gitlab 。

### 2.2 ssh 方式从gitlab clone仓库到edx服务器

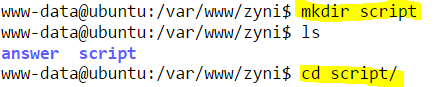
Clone的仓库放在/var/www/zyni/目录下



### 2.3 编写git pull和git push的脚本

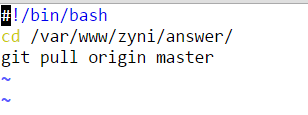
**1. git pull的脚本。**

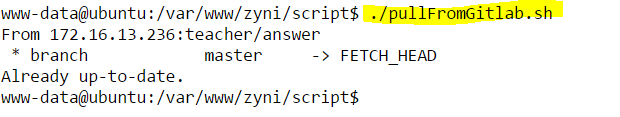
**（1） 在/var/www/zyni/目录下新建文件夹script用来存放脚本**



**（2） 创建pullFromGitlab.sh脚本**



脚本内容如下：(:wq保存)

**（3） 测试pullFromGitalb.sh脚本：**

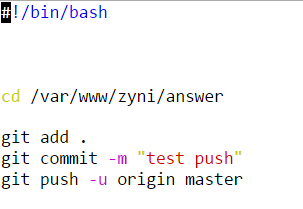
注：如果提示permission dined则更改权限为700

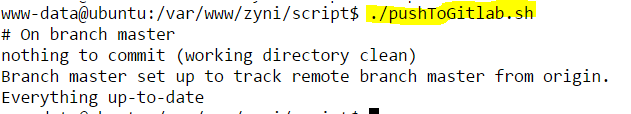


**2. git push的脚本**

**（1） 创建pushToGitlab.sh脚本**



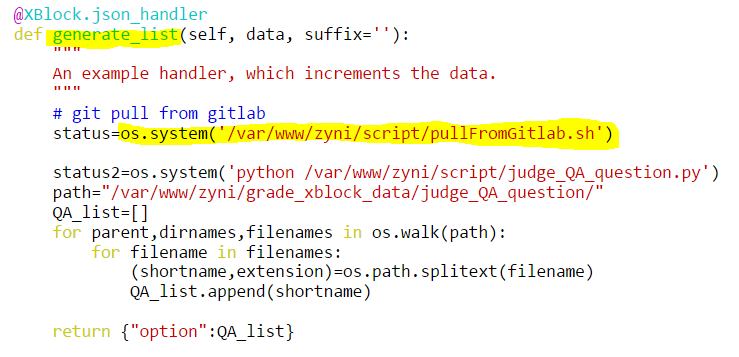
脚本内容如下：(:wq保存)

**（2） 测试pullFromGitalb.sh脚本：**

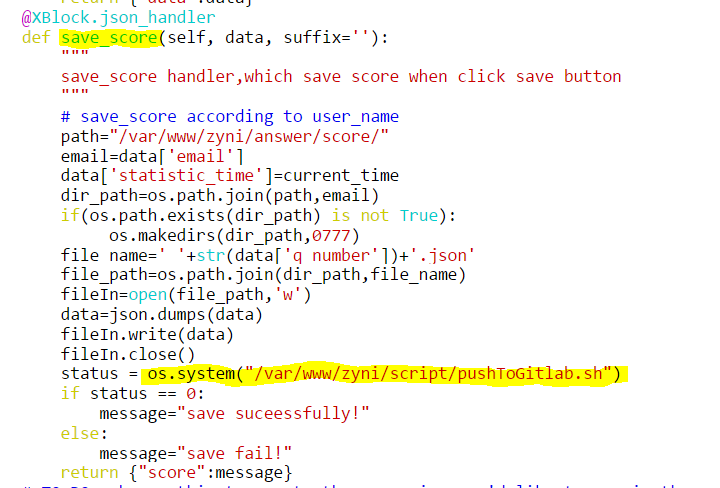
### 2.4 xblock中调用脚本

在xblock中的grade.py文件中调用2.3中创建的两个脚本pullFromGitlab.sh和pushToGitlab.sh

1. **pullFromGitlab.sh的调用**

grade.py中的”generate\_list”方法，通过“os.system(“/var/www/zyni/script/pullFromGitlab.sh”)调用pushToGitlab.sh脚本，实现点击”生成待批改列表”按钮时，从gitlab仓库pull一份最新的学生提交的答案仓库。

1. **pushToGitlab.sh的调用**

grade.py中的”save\_score”方法，通过“os.system(“/var/www/zyni/script/pushToGitlab.sh”)调用pushToGitlab.sh脚本，实现点击保存按钮时，把批改结果push到gitlab仓库中。