

实训第二周报告

学号	姓名	导师	研究方向	报告周期
17343105	田皓	余阳	Go-Online 在线编程项目开发	第 2 周 (11.9 - 11.17)

本周内容

开发签到功能

我们对于签到功能的基本想法是：先 **work** 起来，在处理一些没考虑到的细节，对于一些问题的多种解决方案，先模糊选择一种用起来。

- 设计 api

从学姐给出的产品原型中提取粗略的 api（前端大哥完成）：

__web 学生端：__

- 对于现有的查看课时列表的 api, 能够返回每个课时的签到情况(三个 flag, 分别表示未发起>签到, 已签到和未签到)

- __web 教师端：__

- 指定课程的指定课时, 发起签到
- 指定课程的指定课时, 结束签到
- (websocket 或轮询) 为指定课程的已经发起签到的指定课时生成随机签到口令(纯数字 13>位?), 且每 13(暂定)秒重新生成
- 签到已经结束的时, 查看指定课程的指定课时的签到结果, 返回包含学生学号姓名和是否签到的 flag 的列表
- (websocket 或者轮询) 签到正在进行时, 查看指定课程的指定课时的签到结果(能实时更新是否签到的 flag)

__小程序端：__

- 用户名密码登录(应该可以用现成的)
- POST 签到口令, 返回签到成功与否的 respond

- 修改/新增数据库模型：

在 mongodb 中新增一个集合 CheckCollection 来存放每个课时中每个学生的签到情况。

对于每个课时，增加字段：签到状态（0-未发起签到 1-签到中 2-结束签到）。

部分代码如下：

```
# Model-CheckCollection
```

```
class CheckCollection(mongoengine.Document):  
    class_id = mongoengine.IntField(required=True, default=0)  
    students_check = mongoengine.ListField(mongoengine.ListField(mongoengine.StringField()))
```

```
# Model-Class
```

```
# 课时签到状态 0-未发起签到 1-签到中 2-结束签到
```

```
status = models.IntegerField(default=0)
```

每个课时刚创建的时候 status 为 0，老师发起签到之后改为 1，学生签到时会输入课程码-课时码-随机数组成的签到码，在 mysql 中查找对应课时的状态是否正在签到，如果在签到中则更新 CheckCollection。

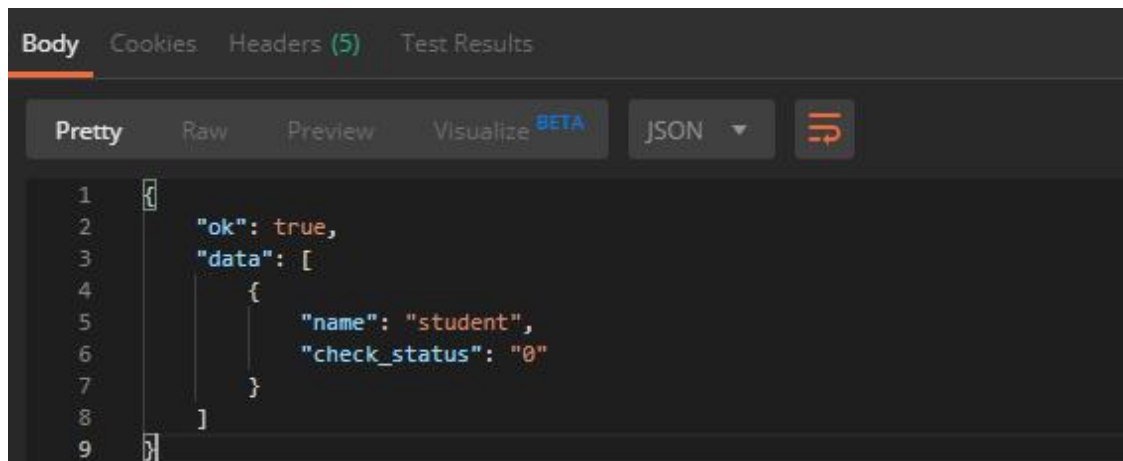
老师查看签到结果只需要查 mongodb 的 CheckCollection 即可，暂时先返回学生的用户名，可以通过用户名再查 mysql 的 USER 表得到学号姓名。

学生查看自己所有课时的签到状态稍麻烦，需要多重数据库查询。

另一种方案是：在 mysql 中建立新表，字段为课程 id，课时 id，学生 id（学号），签到状态（未签到，已签到），这样的缺点是可能课时和学生的多对多关系会让表比较大。

目前正在使用第一种方法，先 work 再说~·

测试老师获得课时签到结果，向对应 url 发送请求结果如下：



其他内容

参加了两次分享会，关于 goonline 使用的微服务架构有了模糊的认识，关于 redis 的基本概念；听了 docker 的相关内容，准备在虚拟机上做一做，建立了仓库用于存放每一次分享会的资料，虽然大部分都不太听得懂 X

开会讨论了实训相关内容，即签到功能的实现，两位前端大哥对我非常好 =w=

学习收获

这周只是做了一小步，算是开了个头，其中也遇到了很多困难，比如数据库崩了，服务器崩了，数据库不知道怎么设计，选择困难症，等等等，还是感觉自己比较弱小，希望在实训的过程学习进步，继续努力·~·

存在问题

- 发现自己数据库很菜，分不清什么时候该用关系型什么时候该用非关系型，没错就是 mysql 和 mongodb。
- 上周的计划太多！

下周计划

- 基本实现签到功能