实训第三周报告

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 1 周（11.17 - 11.24）

本周内容

初步完成签到功能的后端接口

关于 websocket

因为需要实现每隔一定时间变化的签到码，所以我们需要服务端每隔一定时间发送签到码给客户端，方式有 ajax 轮询，long poll 和 html5 的 websocket。

我们比较一下这三种方法：

- 轮询（Polling）

原理非常简单，就是每隔几秒客户端（就是浏览器）发送一个请求，问服务端是否有需要的消息。

举个栗子：

客户端：喂，上次问你的那个，有没有新消息？

服务端：没有。

客户端：喂，上次问你的那个，有没有新消息？

服务端：莫得。

客户端：喂，上次问你的那个，有没有新消息？

服务端：来咯来咯！（返回新消息）

客户端：喂，上次问…~~我是一个没有感情的复读机~~

服务端：……

一直循环

- 长轮询（long poll）

就是阻塞的轮询而已。

客户端：今天没有我要的消息我是不会走的。

服务端：等等，等等，等等，等等，有了，给你，走吧。

客户端：我又来了，没有我要的消息我是不会走的。

一直循环

可以看出，这两种方式，都是在不断地建立 HTTP 连接，然后等待服务端处理，体现 HTTP 协议被动性，就是服务端不能主动联系客户端，只能有客户端发起请求，服务端响应。

简单地说就是，服务器是一个很懒的冰箱（这是个梗）（不会、不能主动发起连接），但是上司有命令，如果有客户来，不管多么累都要好好接待。

这样的缺点是什么呢？就是非常消耗资源。ajex 轮询需要服务器有很快的处理速度，long poll 需要有很高的并发能力。并且，HTTP 是一个状态协议，每一次请求结束不会保存任何信息，下一次发送请求还得带上头部告诉服务器你是谁等等。

所以 websocket 出现了。

- websocket

websocket 只需要发送一次请求，将 http 协议升级为 websocket 协议，然后就可以进行全双工的通信。

websocket 在 danjgo 中的使用

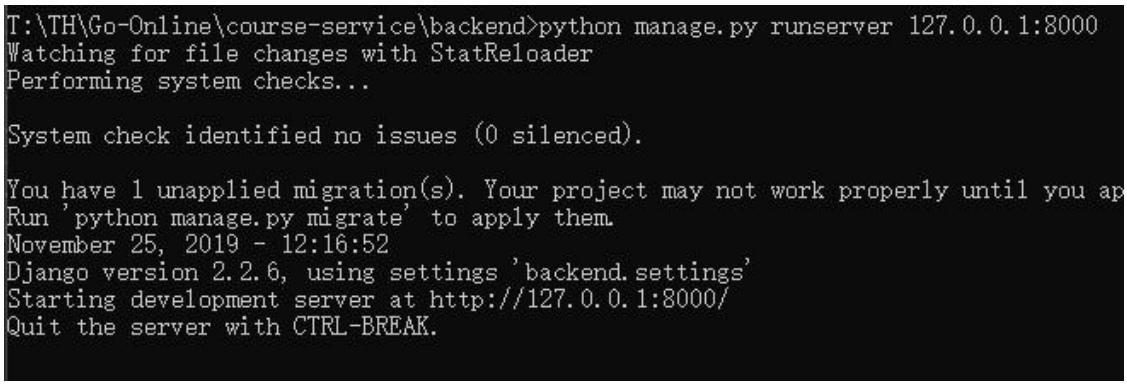
使用 dwebsocket 包，可以很方便的在 danjgo 中使用 websocket。

```
# github example
import threading
from dwebsocket import accept_websocket,require_websocket

clients = []

@accept_websocket
def test_websocket(request):
    if request.is_websocket:
        lock = threading.RLock()
        try:
            lock.acquire()
            clients.append(request.websocket)
            for message in request.websocket:
                if not message:
                    break
                for client in clients:
                    client.send(message)
        finally:
            clients.remove(request.websocket)
            lock.release()
```

启动服务端。



再找一个在线测试 websocket 的网站，连接测试即可，这个例子是客户端发送什么消息服务端回什么消息。

你 12:19:22
等待服务器Websocket握手包...

你 12:19:22
收到服务器Websocket握手包.

服务器 12:19:22
Websocket连接已建立，正在等待数据...

你 12:19:36
abck

服务器 12:19:36
abck

你 12:19:41
sfdsfkelytk

服务器 12:19:41
sfdsfkelytk

你 12:19:51
哈喽哈喽楼啊和hhhhhh

服务器 12:19:51
哈喽哈喽楼啊和hhhhhh

简单实现返回签到码：

你 12:36:16
等待服务器Websocket握手包...

你 12:36:16
收到服务器Websocket握手包.

服务器 12:36:16
Websocket连接已建立，正在等待数据...

服务器 12:36:18
{\code\: \22338\}

服务器 12:36:20
{\code\: \58293\}

服务器 12:36:22
{\code\: \48822\}

服务器 12:36:24
{\code\: \45233\}

服务器 12:36:26
{\code\: \20734\}

服务器 12:36:28
{\code\: \74741\}

服务器 12:36:30
{\code\: \18585\}

服务器 12:36:32
{\code\: \55516\}

完成了其他 api

- 老师得到某课时的签到结果
- 学生签到

学习收获

- 学习了 websocket 原理，以及使用 dwebsocket 在 danjgo 中实现 websocket 连接。

存在问题

- 不知道怎么测试 websocket 接口，因为测试网站没法设置 header，postman 又不支持 ws 协议。

下周计划

- 需要与前端成员测试 websocket 的使用