

# 美多商城项目

## 1. QQ登录-预备工作

- 1) 成为QQ开发者
- 2) 创建开发者应用
- 3) 查询QQ登录开发文档

## 2. QQ登录-开发关键点

获取QQ登录用户的唯一身份标识(openid), 然后根据openid进行处理。

判断该QQ用户是否绑定过本网站的用户, 如果绑定过, 直接登录, 如果未绑定过, 先进行绑定。

QQ用户绑定: 将openid 和 用户user\_id 对应关系存下来。

id	user_id	openid
1	2	Akdk19389kDkdkk99939kdk
2	2	AKdkk838e8jdiafkdkkFKKKFf

注: 一个用户可以绑定多个qq账户。

## 3. QQ登录API

- 1) 获取QQ登录网址API

```
API: GET /oauth/qq/authorization/?next=<url>
响应:
{
  "login_url": "QQ登录网址"
}
```

- 2) 获取QQ登录用户openid并进行处理API

API: GET /oauth/qq/user/?code=<code>

参数: code

响应:

- 1) 如果openid已经绑定过本网站的用户, 直接签发jwt token, 返回  

```
{
  'user_id': <用户id>,
  'username': <用户名>,
  'token': <token>
}
```
- 2) 如果openid没有绑定过本网站的用户, 先对openid进行加密生成token, 把加密的内容返回  

```
{
  'access_token': <token>
}
```

### 3) 绑定QQ登录用户的信息API

API: POST /oauth/qq/user/

参数:

```
{
  "mobile": <手机号>,
  "password": <密码>,
  "sms_code": <短信验证码>,
  "access_token": <access_token>
}
```

响应:

```
{
  'id': <用户id>,
  'username': <用户名>,
  'token': <token>
}
```

## 4. 相关模块的使用

```
# 将python字典转化为查询字符串
from urllib.parse import urlencode

# 将查询字符串转换成python字典
from urllib.parse import parse_qs

# 发起网络请求
from urllib.request import urlopen

# itsdangerous: 加密和解密
from itsdangerous import TimedJSONWebSignatureSerializer
```