

项目开发

一、开发模式

- 瀑布流
- 敏捷开发

二、团队构成

- 后端
 - 后端人员一般是前端人员的两倍
- 前端
 - web
 - 移动
- UI
- 测试
- 项目经理、产品经理

三、开发流程

- 敏捷开发过程

四、接口文档

- 简要描述
- 请求地址
- 请求方式
- 请求参数
- 返回参数
- 案例：

用户注册：

- 简要描述：
 - 用户注册接口

- 请求地址：
 - `http://xx.com/api/user/register`
- 请求方式
 - post
- 请求参数：

参数名	必选	类型	默认值	说明
username	是	string	无	用户名，唯一
password	是	string	无	密码
email	是	string	无	邮箱

- 返回示例：

```
{
  "error_code": 0,
  "data": {
    "uid": "1",
    "username": "12154545",
    "name": "吴系挂",
    "groupid": 2 ,
    "reg_time": "1436864169",
    "last_login_time": "0",
  }
}
```

- 返回参数说明
 - groupid int型，用户组id，1：超级管理员；2：普通用户
 - error_code int型，错误代码，0表示无错误

五、django中跨域问题

- 一个网站不能使用ajax来访问另一个网站的数据，这样的话会被浏览器给阻止，这个问题我们通常叫做跨域问题。

哪怕是同一个域名下的二级域名也不行。

- ajax不允许跨域访问
 - 域名组成 `http://www.google:8080/script/jquery.js`

- http:// (协议号)
- www (子域名)
- google (主域名)
- 8080 (端口号)
- script/jquery.js (请求的地址)
- 当协议、子域名、主域名、端口号中任意一个不不同时，都算不同的“域”。
- 不同的域之间相互请求资源，就叫“跨域”。

跨域访问：

```
Failed to load http://www.test.com/index.py?
rand=0.000729056685777163&: No 'Access-Control-Allow-Origin'
header is present on the requested resource. Origin
'http://localhost' is therefore not allowed access.
```

- django中解决方案：

- 1.安装 `pip install django-cors-headers`
- 2.修改配置文件settings.py

```
#在INSTALLED_APPS里添加“corsheaders”
INSTALLED_APPS = [
    ...
    'corsheaders',
    ...
]
#在MIDDLEWARE_CLASSES添加
MIDDLEWARE_CLASSES = (
    ...
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.common.CommonMiddleware',
    ...
)
#在配置文件末尾添加：
# 处理跨域
CORS_ALLOW_CREDENTIALS = True
CORS_ORIGIN_ALLOW_ALL = True
CORS_ORIGIN_WHITELIST = (
    'http://10.0.108.*',
    "http://localhost"
)
```

```
CORS_ALLOW_METHODS = (  
    'DELETE',  
    'GET',  
    'OPTIONS',  
    'PATCH',  
    'POST',  
    'PUT',  
)
```

```
CORS_ALLOW_HEADERS = (  
    'XMLHttpRequest',  
    'X_FILENAME',  
    'accept-encoding',  
    'authorization',  
    'content-type',  
    'dnt',  
    'origin',  
    'user-agent',  
    'x-csrftoken',  
    'x-requested-with',  
)
```