

# 微信公众号接口开发

设计前后端通信方案

# 前后端分离

- 早期的 web 开发总是使用模板机制，前端代码耦合在模板标签中，前端写好的页面要交给后台处理，项目的开发维护非常不便。
- 前后端分离说的是开发过程中以及项目维护中，可以分开进行。前后端通过接口通信。

# API 通信格式

- JSON 是目前普遍采用的方案，一方面是描述信息的额外数据量比较小。另一方面，在 JS 中可以直接解析。
- 此外，部分 API 返回其他格式仍然是可以的，这需要结合实际需求进行具体设计。

# 接口规范

- 在确定了通信结构和通信方式的情况下，就要设计一套简单的接口规范。
- 前后端需要涉及的接口，要根据规范，并生成标准文档。通过标准进行开发。

# 文档的格式和说明

- 接口文档可以用 `apidoc` 等其他 `npm` 扩展根据注释生成，注释一般以扩展要求的格式编写。也可以直接写。
- 接口的文档一般包括以下几项：  
请求类型、 URL、 参数、 成功返回值、 错误返回值

# API 设计示例

- 因为每个项目所涉及的 API 会不同，这里仅给出通用的字段：

```
{  
    status : 0,  
    data : [DATA]  
}
```

- data 是需要的数据， status 描述了请求是否成功。

# API 设计示例

- 错误的返回值：

```
{  
    status : 1002,  
    errmsg:[ERRMSG]  
}
```

- 通常来说，错误的格式比较通用，只有在成功返回时，其数据会根据不同接口而确定。

# 关于接口的状态码

- 一般接口的状态码并不和 HTTP 协议本身相关。
- HTTP 协议本身是针对请求的，而请求成功，但是在实际的业务操作中并不一定是成功的。
- 另一个重要的因素是，每个系统都需要自己的一套逻辑使用接口来描述。



## 其他参考方案

- 状态码不一定是数字才可以，还可以是简短的错误字符串，比如： `ERRNOTFD`、 `ERRSYS` 。