

微信公众号接口开发

任务练习：实现本地 token 服务与 API 封装

练习 1： `access_token` 动态更新

- 公众号提供的接口，开发者在网站中接入，就相当于也充当了客户端的角色。
- 不要把 `AppID`、`AppSecret`、`access_token` 放在网站的前端页面，这很不安全。
- 在之前的实现中，动态保存 `access_token` 使用了本地文件保存，这在实际的业务场景中是有一些问题的。

文件保存 `access_token` 的问题

- 如果是在网站后台调用，可能出现多个请求都需要 `access_token`，但是已过期。
- 此时多个请求都会去重新获取 `access_token` 并保存到文件，于是会出现冲突。

解决方案

- 使用单独的线程定期更新 `access_token`。
- 其他请求则仅仅是尝试获取 `access_token`。

实现方式

- 用于保存 `access_token` 的方式可以使数据库，也可以是在内存中。
- 使用定时器定期更新 `access_token`。

设计参考方案 1

- 使用数据库：启用单独的线程 / 进程设置定时器把获取的 `access_token` 写入到数据库。
- 调用其他接口如创建菜单、上传素材等则先去数据库获取 `access_token`。

设计参考方案 2

- 启用一个本地服务，只返回 `access_token`。
- 服务启动后会设置定时器，把 `access_token` 等附加信息保存到一个全局变量。
- 服务接口只是返回全局变量保存的 `access_token`。

要求

- 实现动态获取 `access_token` ，并且在有多个请求时，不会导致 `access_token` 冲突。

练习 2：封装菜单管理、素材管理接口

- 设计一个微信公众号接口开发库，把菜单管理、素材管理进行封装。
- 使用者在调用接口时，只需要传递需要的参数，并获取结果。

示例

- 创建自定义菜单：

```
//menu_data 是菜单数据  
wxapi.menu.set(menu_data);
```

- 或者显式传递 access token：

```
//menu_data 是菜单数据  
wxapi.menu.set(access_token, menu_data);
```

示例

- 上传图片素材：

```
// 文件路径，第二个参数传递选项表明类型  
wxapi.media.upload(image_file, {  
    type : 'image'  
});
```

设计参考

- 公众号提供的接口，就是通过参数构造 URL 并进行调用。
- 重复的内容可以抽离出来进行封装， URL 参数或者 POST 提交的数据都可以作为封装接口的参数。

练习 3：用户和标签管理接口

- 学习过菜单管理和素材管理接口的调用以后，同学们可自行学习如何调用用户与标签管理的接口。