安全鉴权双token无感刷新

使用双Token进行服务接口安全鉴权，现在已经是标配，但是在实践中发现了一个对用户不友好的体验问题。

最开始，我们在使用refresh_token实现access_token的续期时，只返回了新的access_token，当双token都过期时，会产生一个不好的结果：

上一秒我在C端操作，下一秒access_token过期了，再发一个请求，服务侧判定为401。这时C端对401拦截处理，尝试拿refresh_token去续access_token。

然而，refresh_token也过期了，C端只能跳转到登录页，让用户通过登录重新获取双token，此时是“有感”的，而且是很不好的体感。。。

比如上一秒我在编辑很重要的表单，下一秒提交这个表单的时候，出现了401且refresh_token过期，C端强制跳转到登录页，重新登录之后，发现我之前填写的重要资料信息都不见了，必须重新编辑一次。。。WTF！！！

有一种做法是C端在跳转登录页前，将表单信息保存在本地，登录完成后，跳转回原来的编辑页面，并且自动填好表单信息。。。如果这种需要处理的表单页面很多的时候，我隐隐地感觉到前端同学的嘴里会不自觉地冒出“WTF！！！”。。。

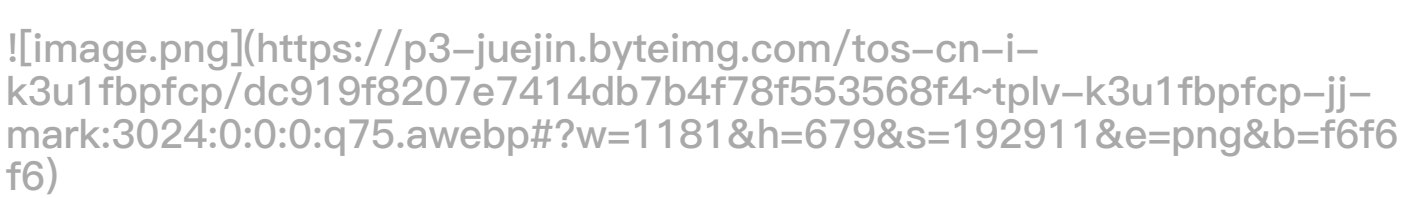
所以，这种让前端同学说“哒咩”的方案不可取。

其实需求的场景主要是在用户体验侧，用户在持续操作的时候，不想突然被要求登录，那导致必须登录的问题源头在于双token同时过期了，怎么避免呢?

如果每次在对access_token续期的时候，同时生成新的refresh_token是不是就可以了？嗯，听起来好像没有问题，但是这是不是科学的做法呢？有没有什么隐患?

由于这是一个大众问题，理论上RFC规格里应该会有相关描述。于是带着重新学习的态度查阅了一遍 [RFC6749 RefreshToken](http://cxyroad.com/

”https://www.rfcreader.com/#rfc6749_line426”) 的规格说明，其中使用refresh_token刷新access_token的交互图如下：



相关说明：

```
(H) The authorization server authenticates the client and validates the refresh token, and if valid, issues a new access token (and, optionally, a new refresh token).
```

从这 第H步 的说明来看，在实现时可以根据需要，返回一个新的refresh_token，这是符合规格的。

所以做到无感刷新，免受突然强势插入登录跳转的困扰，只要在每次更新access_token时，同时返回新的refresh_token就可以了。

那么，refresh_token的过期时间要怎么设置呢?

这是另一个体验决策，如果希望用户连续多天不登录后能够继续体验“无感”，也就是仍旧可以免登录，则将过期时间设置长一些。如果只要保证用户在持续操作时的“无感”体验，则将过期时间设置短一些。理论上，过期时间短些，更安全。

如上。