



链滴

关于 JWT 写登录授权接口的分析

作者: [remixjc](#)

原文链接: <https://ld246.com/article/1575510385122>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



前情概要

目前已经实现了一版逻辑，想要在这个基础上进行更多更多的细节扩展，比如扩展 token 的保存时间这时候就需要增加 refresh_token 来请求新的 token 数据

解决方案

解决方案这里分两种情况说明，当前的解决方案、扩展的解决方案

当前的解决方案

当前的解决方案主要是基于标准的JWT加密的token返回，设置时限(这里以30分钟为例)，利用redis储双倍的时间来进行后续接口请求的验证，如果超过时限但是在2倍时限之内的话，返回对应的授权要重新请求的状态码，当判断获取到状态码之后，用现有的token去请求刷新接口，这个接口验证token时验证是否在时限和双倍时限之间，如果是，重新返回新的token，如果token超过了双倍时限，那则拒绝请求，必须重新请求登录授权获取。

这里的redis中存储的键值可以做一些处理，来验证token信息的可靠性

防止token被利用，所以时间可以尽可能的少一些

扩展的解决方案构思

首先是按照正常方式来授权获取token，同时不带access token及refresh token，两种token中得数类型不同，其中得加密方式也可以不同，在拦截器中判断传入的方式从而实现不同的业务逻辑。

在刷新token时，会作废掉refresh token，这样可以保证长期登录，如果超过一个月不登陆，那么在edis里面的数据就会消失，因为在redis里面保存了最长30天的数据，而access_token则只保存2小时

最终接口形态

1. 登录授权

请求方式: POST

数据格式: JSON

请求参数:

参数	描述
username	登录用户名
smscode	短信验证码

响应结果:

```
{
  "code": 200,
  "data": {
    "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOiE1NzU1NDAxMDk5DgslbWVxvYWQlOiJ7XCJtZXJjaGFudGlkXCI6MTEwMixclm5hbWVcljpcuWFqOaZr-inhuinieKoOebn1wiLFwiaWRcljoxMTAyMDAwMSxclmFnZW5jeW5vXCI6XCllwMDMwMDFclixclmVtYWlXCI6XCllwMjNAMTIzLmNuXCJ9In0.TfnSxoXbHs4ckal-M1i8AmmDCTWWp52x3BTcYdqNNBA",
    "refresh_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOiE1NzU1NDAxMDk5ZkslnbWVxvYWQlOiJ7XCJ1c2VybmFtZVwiOlwiMTUwMTEuNDUyMDBcln0ifQ.Aao1veTc_DEQLh2EmxjBf2B80LO4b4zgA6lgxloR_4",
    "expires_in": 7200,
    "info": {
      "merchantid": 1102,
      "name": "name",
      "id": 11020001,
      "agencyno": "003001",
      "email": "123@[123.cn](http://123.cn/)"
    }
  },
  "message": "Ok"
}
```

2. 刷新授权

请求方式: POST

数据格式: JSON

| :-- | :-- | :-- |

| **app-refresh-token** | Header | 登录授权时获取到的refresh token值, 正常可以保存30天, 当使刷新授权后, 该refresh token失效, 再次请求刷新请使用最新获取到refresh_token, 如果请求都不用, 则需要用户重新登录, 该内容需要客户端严格保存 |

| **app-access-token** | Header | 登录授权时获取到的access_token值, 正常保存2小时, 2小时后效, 失效后需要调用该接口用refresh_token获取最新并保存到客户端 |

响应结果:

```
{
  "code": 200,
  "data": {
    "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOjE1NzU1NDAxMDk5DgslbBheWxvYWQiOiJ7XCJtZXJjaGFudGlkXCI6MTEwMixlcm5hbWVcljpccluWFqOaZr-inhuinieKoOebn1wiLFwiaWRcljoxMTAyMDAwMSxclmFnZW5jeW5vXCI6XCiwMDMwMDFclixclmVtYWlXCI6XCixMjNAMTlzLmNucXJ9In0.TfnSxoXbHs4ckal-M1i8AmmDCTWWp52x3BTcYdqNNBA",
    "refresh_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOjE1NzgxMjQ5MDk5zkslnBheWxvYWQiOiJ7XCJ1c2VybmFtZVwiOlwiMTUwMTExNDUyMDBcIn0ifQ.Aao1veTc_DEQLh2EmxjBf2B80LO4b4zgA6lgxloR_4",
    "expires_in": 7200
  },
  "message": "Ok"
}
```