目前借鉴的方法：前端把account和password，提交到服务端的登录api，服务端验证正确后，生成一个token，并把token和userId，存在缓存里（推荐redis数据库），然后把token返回给前端。前端每次的请求头中带token，这样就是简单的token 机制

**15 条评论**

前端每次的请求头中带token，这个如何做到呢？

你这里的前端指的是什么？

前端就是发起ajax请求的端

以angular为例，前端缓存token 然后通过切面中间件加工每次ajax请求的headers

这样会造成一个问题，因为这个token是有 有效期的，假如说是一天的有效期，那么就会出现一种情况，用户正在操作，突然token有效期到期了，前端会跳转到登陆界面。这样用户体验太差了。

**token可以有两种解决方案**： 1、获取token的时候同时携带过期时间，每次ajax请求当前时间（提前10分钟）和过期时间作比对，如果大于那么调用token重新获取接口更新token； 2、token心跳机制，后台服务自动续期； ———————————————— 延伸阅读： [知乎专栏](https://zhuanlan.zhihu.com/p/19920223?columnSlug=FrontendMagazine)

想更加安全性就 http更换为https

那这样岂不是很矛盾？token的有效期还有什么用？如果使用心跳和session岂不是一样的？意义何在？前后端分离就没有意义了？实在是头疼啊！

赞查看对话回复踩举报

token主要是解决前后端分离后无session 安全性的痛点，比如说没有token的api如果被第三方恶意使用对服务器和数据都很危险，通过https（证书）+token很大程度上能够预防该问题，过期时间就是为了防止token长久有效。心跳主要用于app，在用户登录使用app后很长一段时间内是不需要考虑token过期问题，但是当用户很久未使用app，那么从安全性考虑就需要用户重新登录。

今天看到**一个token解决方案**：登录后得到token 和刷新token（均有过期时间返回）然后做本地缓存，然后在快过期前使用刷新token 来获取新的token 和刷新token

更全面的就是需要设备id和操作设备类型和操作设备版本

非常感谢你的建议。我**目前的做法**是只要用户使用任何一个接口，token有效期顺延一天，这样就不会出现token有效期到了，跳转到登陆界面重新获取token，如果用户超过token有效期的话，就需要重新登录获取新的token。

**如果我直接去访问登陆外的其他页面呢？如何做拦截**

这个可以用前端的sessionstorage，把token存在session里面，在每个页面判断有无session，有的话不做处理，没有跳转至登录界面

前端把account和password，提交到服务端的登录api，服务端验证正确后，生成一个token，并把token和userId，存在缓存里（推荐redis数据库），然后把token返回给前端。前端每次的请求头中带token，这样就能够轻松的实现

安全性：

楼上说的基本上现在都是这么做的吧，至少我没有见过其他更多的更好的方法  
restapi这边基本都是用token来做的  
如果要更安全的话  
就是前后端的数据加密了  
1.HTTPS是一方面，保证数据在传输的过程中不被篡改  
2.另外可以做的可以考虑对参数加密，比如我前端传参的时候ABC三个参数，是按照你既定的规则来加密的，最后传输到后台的实际上是一个加密的字符串，后台再根据规则来解密得到参数，响应的json也一样处理。  
这样一定程度可以防止

居然用token的被采纳了, 难道每一个url都加上token吗? 呵呵  
说说我的:   
其实还是用cookie的啊, 跟原来没啥区别:

1. client发送username和password到server
2. server验证成功后, 写cookie到client,然后返回ok的json, 其中cookie的key要存储在redis中,value就是用户信息, 并且要设置key的超时时间,如:60分钟
3. client收到ok后, 进行相应的业务操作, 以后每次请求server都会自动带上cookie, 不用你写代码
4. server端的filter(你肯定用filter来实现)中会每次验证传过来的cookie的key在redis中是否存在, 有就代表登录成功过可以操作, 没有就返回错误标识注意: 在登录成功后, 每次调用服务器接口时候, 都要为redis的key进行续期,如60分钟
5. 当redis的key超过60分钟, 自己会删除这个key, 那么再次请求server时, 就会收到需要登录的返回值
6. 当用户主动退出系统的时候, 也要在server中删除redis的key

再来说说token:token一般用来做显示的身份验证, 比如手机验证码之类的, 这些都需要手动传递token到后台

1. 在client点击验证码
2. server发送验证码,返回给client一个token
3. 在client对验证码进行验证,传递token + 验证码到server
4. server对token+验证码进行验证

cookie怎么解决跨域的问题？

[**馒头see**](https://segmentfault.com/u/mtsee) · 2016年09月08日

nginx代理啊, 任何域名都可以转换成你自己的想要的域名, 比如你是 www.app.com , 想代理苹果官网 www.apple.com , 在你自己的nginx里面配置一下就可以了, 比如配置成 www.app.com/apple

[**yuechen323**](https://segmentfault.com/u/yuechen323) · 2016年09月08日

你这个方案对于浏览器这样的ok，但是楼主的标题是rest api啊，目前只能token，token可以作为参数，也可以放到head里就行了

[**monolithic**](https://segmentfault.com/u/monolithic) · 2016年12月08日

回复 [monolithic](https://segmentfault.com/u/monolithic)：

什么啊??? nginx放在前面做反向代理, 后面一堆java的微服务的rest接口这是很正常的架构, 跟浏览器没有关系