前言：公司的业务主要是对接财务系统做单据传输或者凭证处理的，难免少不了和各大财务软件做数据对接，其中当然是必须通过接口来传递数据了。于是乎，用友T+的版本来了，对接的工作自然是我来做，可没想到就是这样一个T+接口，搞得我快吐血了。

1.先简单描述下项目的运行环境，.net core
2.2。这也是导致后来一直卡在接口对接上的一大原因。接口不太支持.net core的运行环境。因为自己之前也处理过财务接口，金蝶K3，KIS账务平台，K3wise，U8 Cloud等等，说实话，T+还是第一次接触，当然第一件事肯定是看官方文档了。直通车T+开发平台，不过打开之后，浏览器一直在刷新页面，后来问了社区的服务人员人家说换个浏览器试试，换了T+的浏览器，结果还是一直刷新，到底是啥问题，咱也不知道，咱也不敢问,后来在官网的文档中给了解释，说是最好使用IE和360浏览器，估计是客户使用的比较多吧，所以才不兼容主流的Google浏览器嘛？

2.OK，那么首要任务就是先熟悉接口定义以及相关的请求方式和请求参数等等，当然返回参数也是需要的。接着就开始将接口封装到项目中了，T+ OpenAPI v2 接口需要引入鉴权机制，简单的来说就是需要在每次请求业务接口时，请求的Header需要带上Authorization参数，那么Authorization怎么来获取呢？见下图，即 对appKey,authInfo,orgId做base64位的加密。那么authInfo的值又如何获取呢？请看官方文档的注释，即 对 appkey,orgid,appsecret这三个参数做一次签名算法，那么还有两个问题需要解决。第一，这三个参数怎么获取？ISV申请 http://tplusdev.chanjet.com/enterprise 官网注册后会有ISV的申请，点击申请填写好资料后系统会有人员审核资质并发放具体的加密密钥。好了，第一个问题解决了，那么第二个问题呢？签名算法怎么做。不要着急，官网也有提供，这里仅标注C#版本，因为使用的环境是.net 的 签名1的下载地址签名1下载。

二、

1、使用云企业ID（其实就是增加access_token），

Httpurl：http://localhost:8088/tplus/api/v2/Inventory/Create

（其中 http://localhost:8088/ 需要对应更换为自己的服务器地址）

Headers:

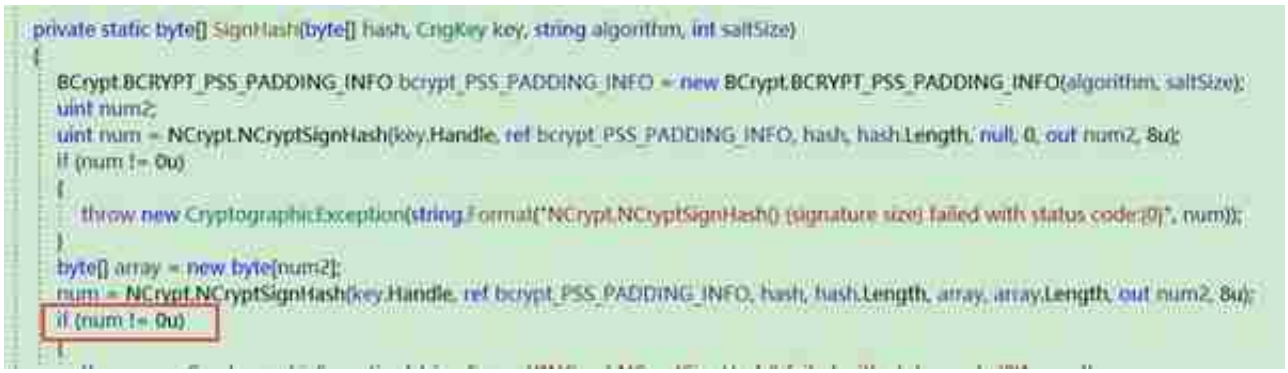
```
{
    Authorization:base64{
    "appKey":"138750dc-aa1a-419a-8c92-7e1966fcd6fe",
      "authInfo": 签名2{
              {"appkey": "138750dc-aa1a-419a-8c92-7e1966fcd6xx",
                                  "orgid": "90006696642",
                                  "appsecret": "ifsaxx"},
                          "私钥的文件全路径"
        "access_token": "一.1或一.2中获取到的token"
           },
    "orgId":"90006696642"
    }
}
```

PostBody:

```
{
    _args:{
    业务参数与之前v1一致,
    type:"Asyn"//如果需要异步，那么传递此参数
```

签名算法2

4.好了，坑终于来了，下载完demo后，将demo中用到的dll拷贝到自己的项目中去。主要用到的dll 有这么多



jose-jwt源码签名加密

5.至此，终于是将T+的接口给搞定了，也不枉自己在这上面耽误这么多时间，感觉自己的头发又掉了好多。最后的最后，附上.net core
环境下T+的接口调用，给有需要的朋友，也免得大家入坑。Tplus.net
core接口下载

总结一下，在处理接口问题的时候，其实最重要的还是要彻底理解接口的定义，然后做起来才能得心应手，不过，也很感谢这次经历，起码有了对接T+接口的经验，下次遇到同样的项目就可以直接使用了，也免去了后续的麻烦。也希望自己在以后的工作中能更加有经验。加油。