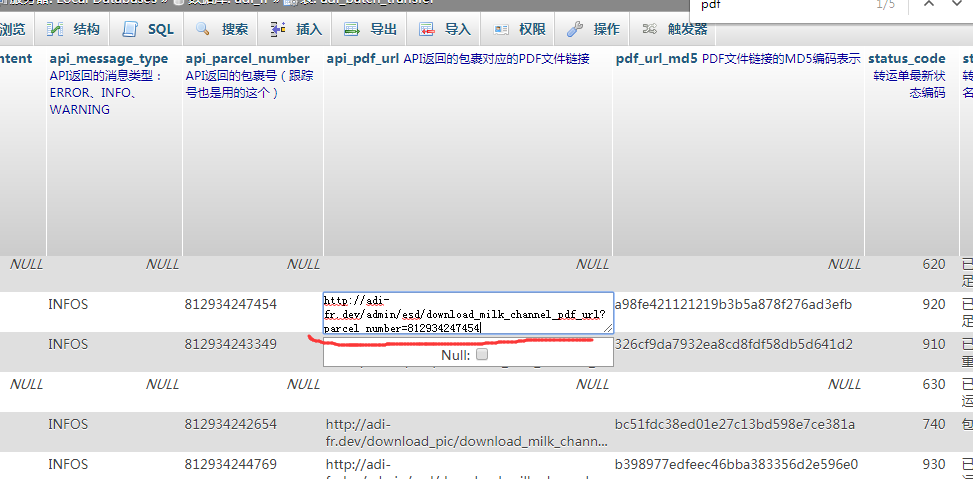
**关于fr面单下载流程**



面单下载形式都是指向一个download函数，然后带一个token值。

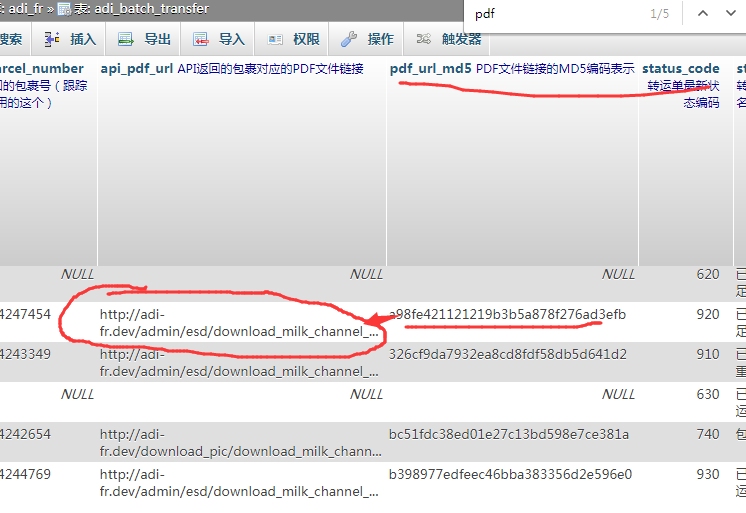
为什么要这么做？

1. 为了安全性。防止我们系统的各种ID或者关键参数暴露给用户，万一有不怀好意的使用者。增加了下载地址的请求难度，如果只是简单的带上order\_id那么有些人利用ID自己拼凑可能会直接得到争取的面单（别人的订单）



例如：<http://adi-fr.dev/admin/esd/download_milk_channel_pdf_url?parcel_number=812934247454>

1. 统一页面逻辑。每种业务的开单方式不一样，然后获取面单的方式也不同，有些是直接外部地址，有些是内部地址，有些可能是文件地址，或者访问时候生成。所以为了，不每次都要去判断或者修改页面上的很多下载地方，采用统一的token验证,只需要在跳转函数做修改，页面不用管。
2. 设置双层地址更加灵活。



不管这边地址以后做任何调整，下载部分程序依然是按照token来匹配，真实的下载地址。

只需要管实际获得图片地址的处理就行了。

以上是个人见解。

**注意事项**

第二层地址的权限问题！

例子：

法邮地址：<https://ws.colissimo.fr/sls-ws/GetLabel?parcelNumber=EY139789573FR&includeCustomsDeclarations=true&x=0&y=0&signature=012c3296b204f2293d680d766bf48c65db68f3162ba779c47e4a47547176f625>

(指向外部，所以没有权限问题产生。只要在访问一层地址token验证函数的时候有权限判断就行了)

奶粉地址：

<http://adi-fr.dev/admin/esd/download_milk_channel_pdf_url?parcel_number=812908064729>

（指向内部，并且esd 有后台登陆验证）

<http://adi-fr.dev/biz/download_milk_channel_pdf_url?parcel_number=812908068407>

（指向内部，并且biz有前台登陆验证）

以上地址是前两次使用的二层地址，均指向内部

在这里坑就出现了，当需要验证的时候。有些情况就会出问题。由于二层地址是同一个函数.

前台有单独下载自己面单的情况，所以下载地址在esd内的时候。前台用户要单独下载自己面单，则会跳转到后台登陆页面，提示登陆。后台管理者，可以正常下载面单

-----------------之后迁移到biz前台---------------------

Biz带前台验证，当后台管理员没有登陆前台用户账号的时候。会跳转到前台提示登陆（这些情况都是不合理的）

发生上面情况的原因：

1. 写代码的时候马虎，没有查看控制器的构造函数。发现是带有登陆验证的
2. 没有仔细思考，下载地址的使用场景。
3. 测试的时候一般都是前台账号登陆，后台也登陆了的，没有做单独的登陆测试。

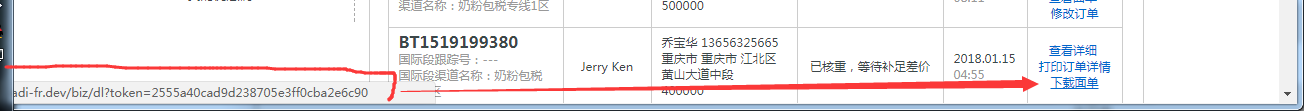
最后调整到了不用验证的下载地址。（验证放在第一步token 验证地址中去）

<http://adi-fr.dev/download_pic/download_milk_channel_pdf_url?parcel_number=812908069871>

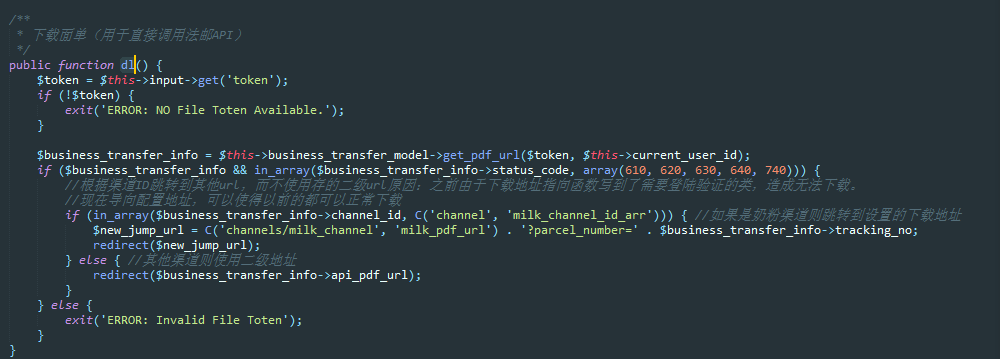
（指向内部，download\_pic没有验证）

**业务流程**

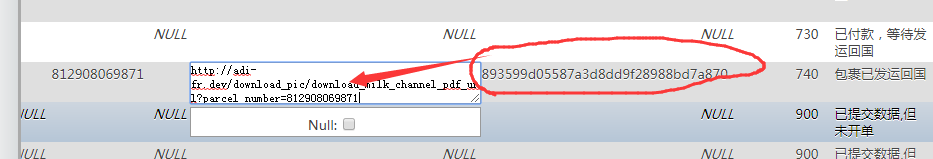
一级跳转地址(token验证地址)



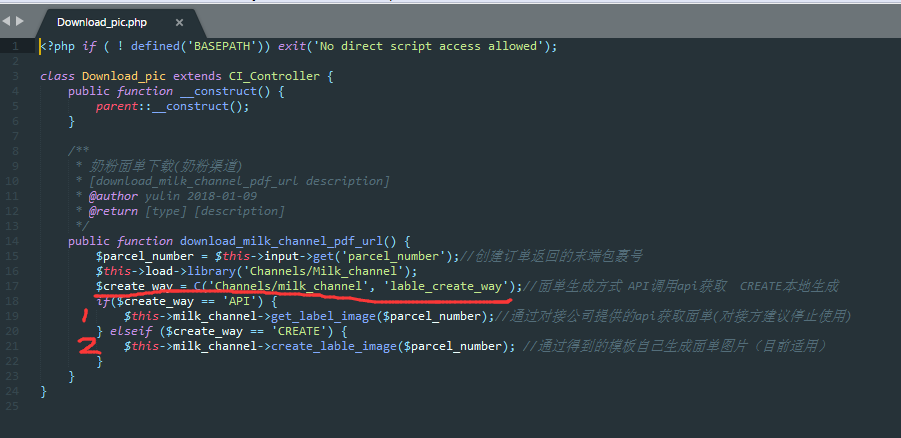
访问到了



根据token查找二级跳转地址



二级地址是实际上下载图片的地址



二级地址按照业务逻辑灵活设定，这里展示的是指向内部的跳转地址情况

跳转到此处之后就可以完成得到图片的动作