知乎模拟登录

## 一、分析登录请求

在Chrome中新建一个隐身窗口进行分析，隐身窗口中没有之前的cookie信息和其他访问记录。

1. 在请求之前请先打开‘开发者控制台’（F12），然后选中Network->preserve log，这样就能不漏掉中间重定向的请求日志，如图1所示。

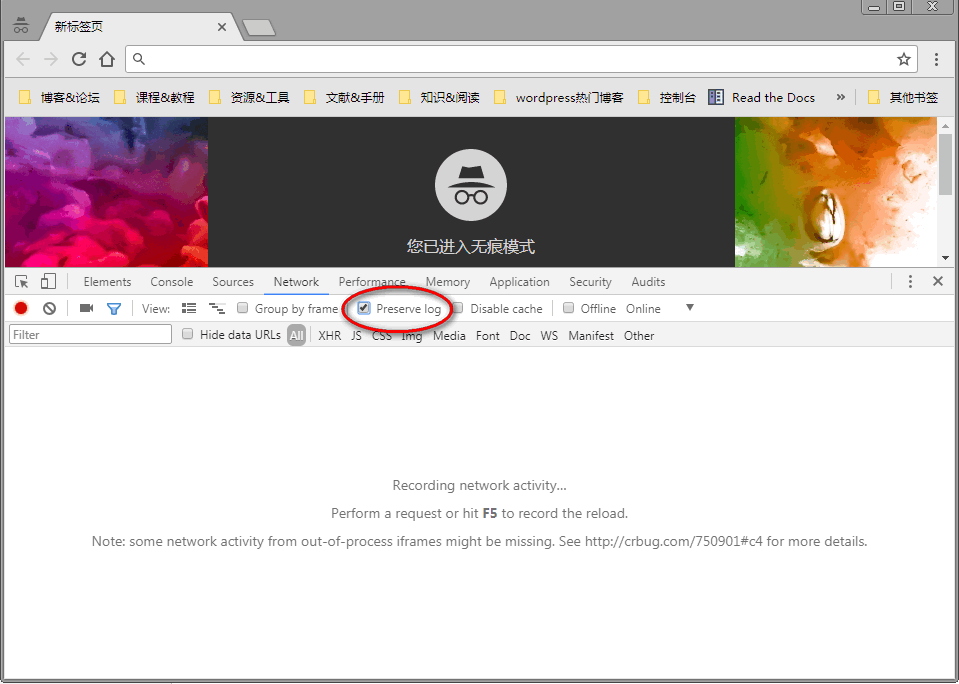


图1 打开控制台，准备抓包

1. 如果直接请求知乎官网<https://www.zhihu.com>，就会被重定向到<https://www.zhihu.com/signup>，即注册界面，虽然没有明确的地址显示登录URL，但是可以大胆猜测应该是<https://www.zhihu.com/signin>，请求后果然直接到了登录页面，如图2所示。

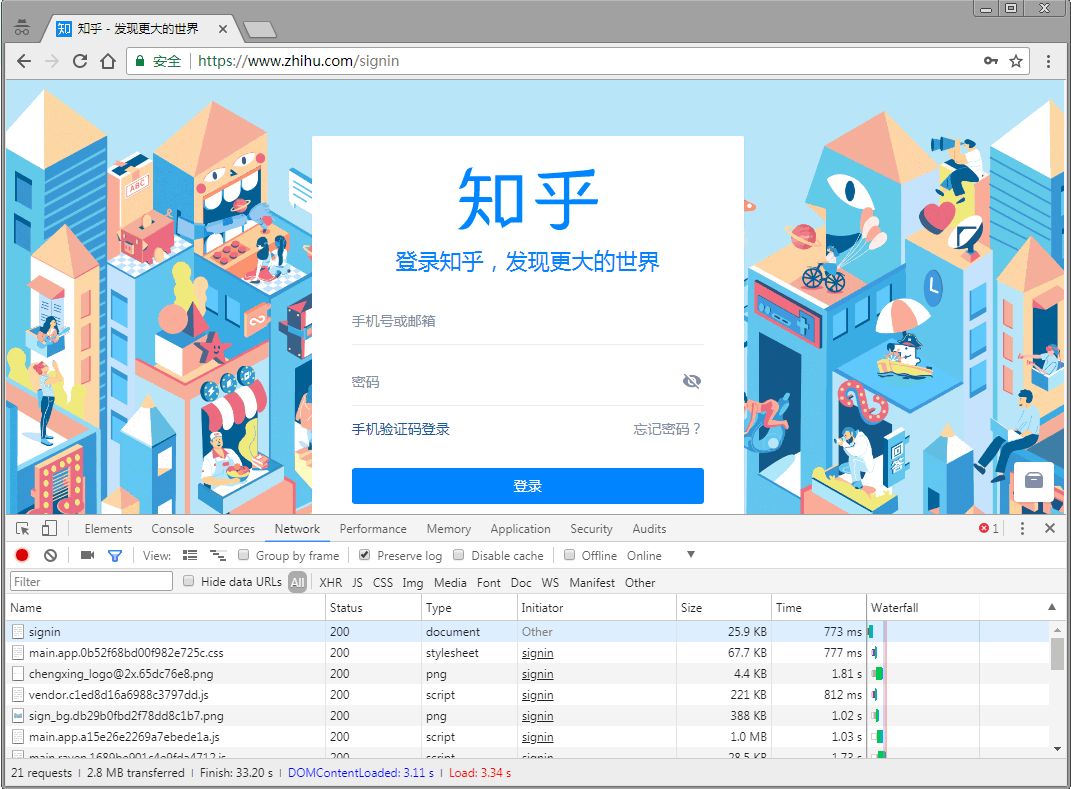


图2 请求登录页面

1. 然后登录知乎，观察控制台的请求日志，可以找到用户名密码被POST到了这个API，如图3所示。

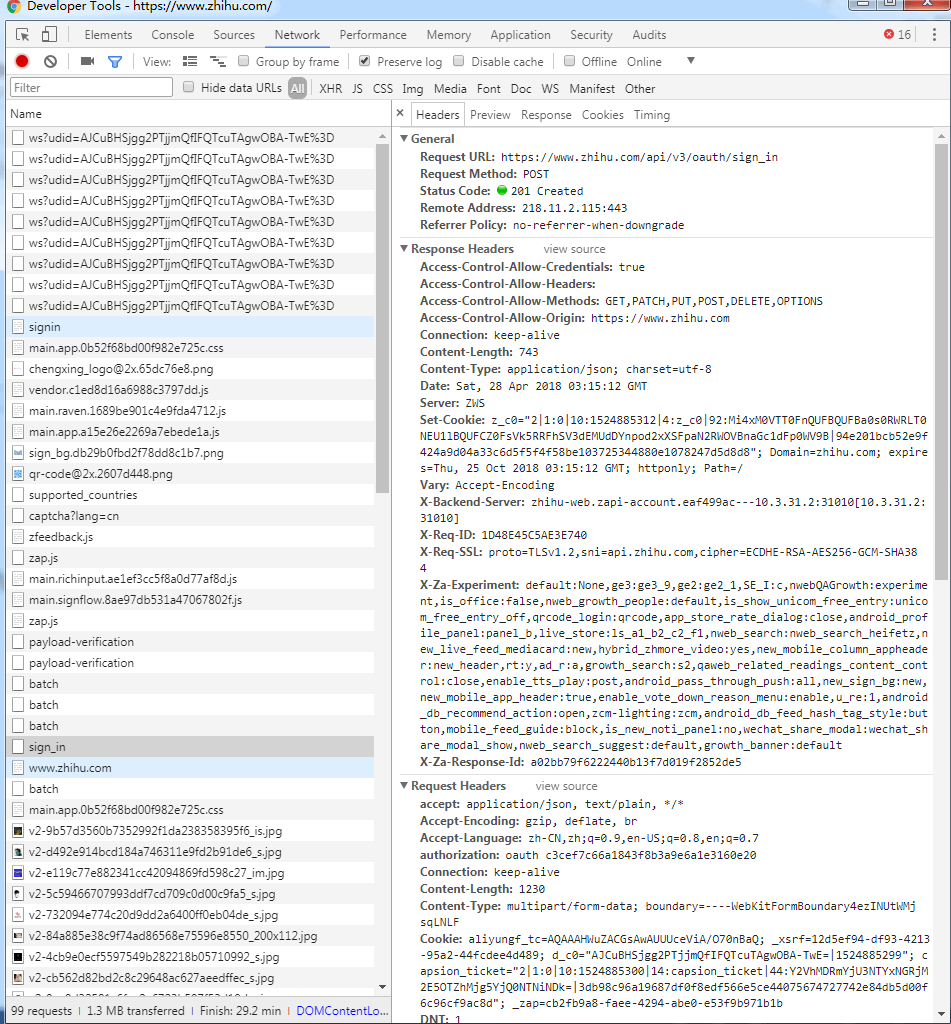


图3 知乎登录认证API

1. 有四种常见的POST请求方式，参考<https://imququ.com/post/four-ways-to-post-data-in-http.html>，可以在控制台中看到请求头中Content-Type就是multipart/form-data，所以会在Request Payload看到这么特殊的数据格式，刚刚登录时输入的用户名密码都会明文方式显示在这里，如图4。

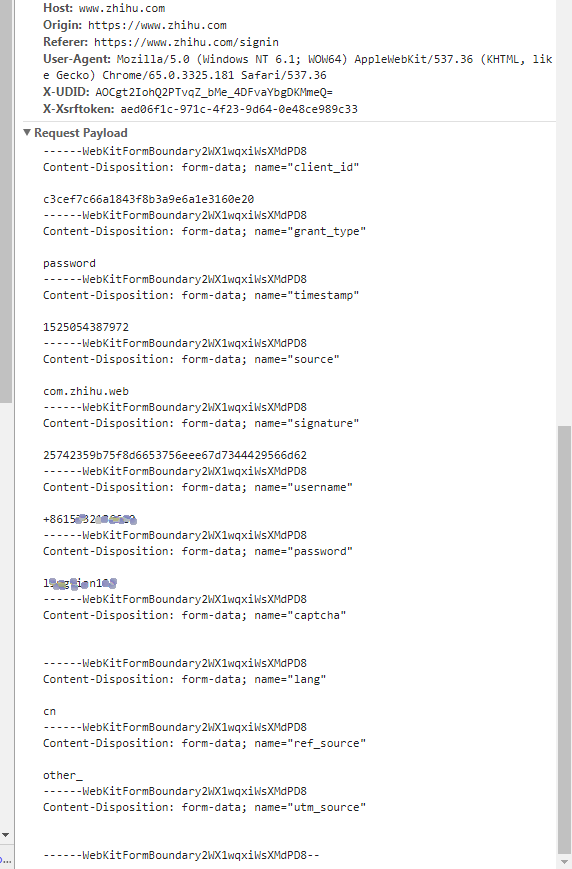


图4 POST数据

1. 图4中的数据就是我们在请求该接口的时候发送的东西，client\_id是不变的，timestamp是时间戳（毫秒级别），signature待会再看，username和password分别是知乎账户的用户名和密码，captcha即验证码，验证码有数字字母组合和点击倒立文字（7个字中有1~2个倒立），分别对应的captcha内容形式如图5。

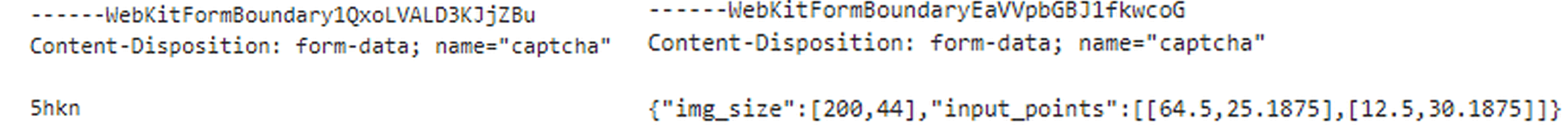


图5 左：字母数字组合，右：倒立文字

如果是数字字母组合的验证码，那么captcha内容就是输入的验证码；如果是倒立文字，由于是服务器生成验证码图片，大小固定是200x44，input\_points为用户点击的x,y坐标。

lang应该是验证码的语种，分为en和cn两种。

至于其他的字段不是很重要。

1. 通过在控制台中搜索signature关键字【Ctrl+Shift+F】，可以搜索到在js中存在关于signature的生成算法，如图6所示。

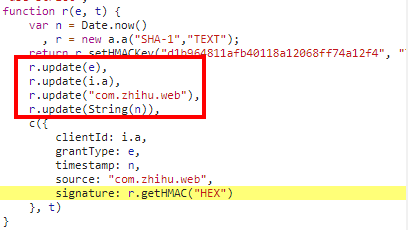


图6 signature生成算法

可以知道这是一个Hmac算法，一定要按照框起来的顺序进行加密生成signature，使用python中内置的库hmac可以很轻松的还原这个算法，代码如下：

*def \_get\_signature(grant\_type =’password’, client\_id , source=’com.zhihu.web’,timestamp):*

*ha = hmac.new(b'd1b964811afb40118a12068ff74a12f4', digestmod=hashlib.sha1)*

*ha.update(bytes((grant\_type + client\_id + source + timestamp), 'utf-8'))*

*return ha.hexdigest()*

1. 最后一个大问题就是验证码的问题，在控制台中可以看到是否需要验证码取决于第一次请求验证码接口返回的结果，如图7。如果为True那么在第二次请求时会返回图片的base64编码，如图8。

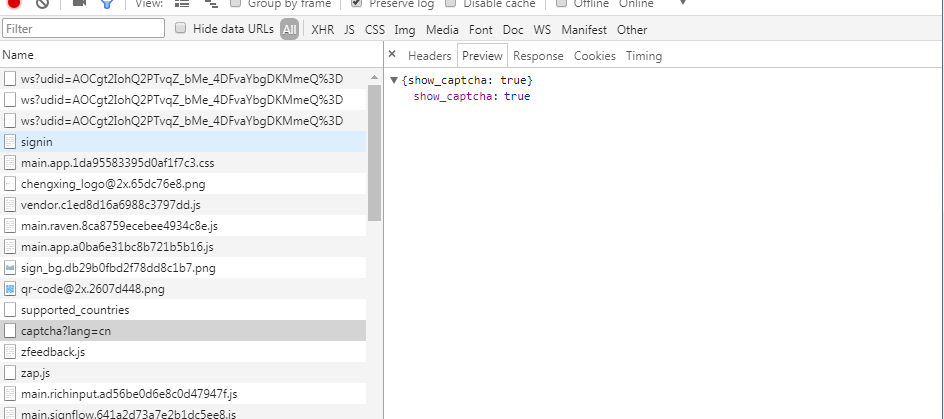


图7 第1次请求

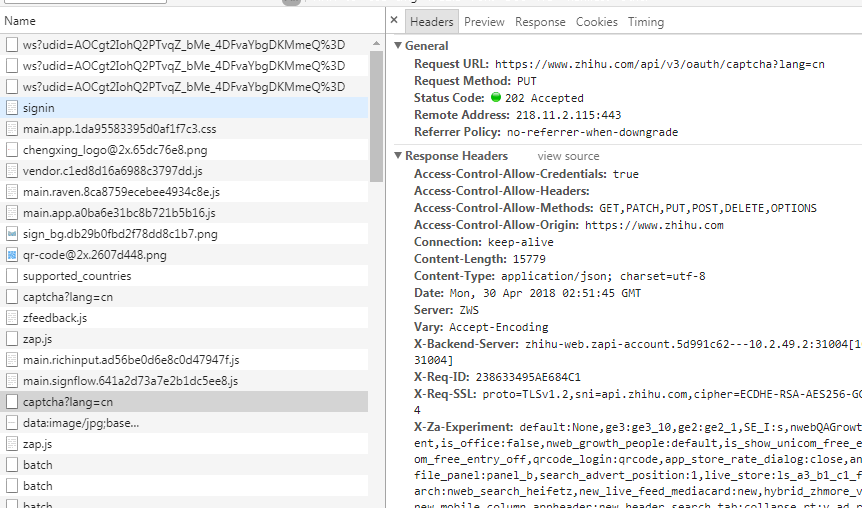


图8 第2次请求

其实，如果出现验证码刷新一次页面就可以避过验证码，当然也可以将图片保存到本地，然后显示出来进行手动输入。

1. 请求所有的URL的时候，都要使用同一个Headers。而且请求头中需要包含的字段有User—Agent、authorization、X-Xsrftoken，其他的酌情添加，其中X-Xsrftoken就是第一次请求页面set-cookie中的xsrf，而authorization是固定的，直接复制过来就可以。

## 二、模拟登录步骤