本文介绍使用 Python 的 urllib2 和 cookielib，登录哈工大统一身份认证系统。

爬网站的时候，很多都需要先登录。登录分为带验证码和不带验证码，不带验证码很简单，直接 Post 表单即可，带验证码需要获取验证码图片，使用某种手段解析出来，再带验证码提交。下面以登陆哈工大统一身份认证系统为例，解释如何完成一次带验证码的自动登录。对登录其他网站也有借鉴意义。

一．分析网站信息：

【1】哈工大采用统一身份认证，即不论登录任何系统，都会跳转到统一身份认证页面进行认证，成功之后再跳转进对应页面。

研究生院转统一身份认证：

http://ids.hit.edu.cn/authserver/login?service=http%3A%2F%2F219.217.227.152%3A8080%2Fhitgsmis%2FindexActionCAS.do

教务处转统一身份认证：

http://ids.hit.edu.cn/authserver/login?service=http%3A%2F%2Fjwts.hit.edu.cn%2FloginCAS

乐学网转统一身份认证：

https://ids.hit.edu.cn/authserver/login?service=https%3A%2F%2Fcms.hit.edu.cn%2Flogin%2Findex.php%3FauthCAS%3DCAS

可以看到，它是用 service 值来标记从哪来，认证成功之后要到哪去。

【2】网页完整信息见：哈工大统一身份认证.html

【3】使用审查元素，可以找到登录表单，发现它长这样：

```
<form id="casLoginForm" class="fm-v clearfix amp-login-form" role="form" action="
/authserver/login?service=http%3A%2F%2F219.217.227.152%3A8080%2Fhitgsmis%2FindexActionCAS.do" method="post">

        <span id="msg" class="auth_error" style="top:-19px;">请输入验证码</span>
    <p>
        <i class="auth_icon auth_icon_user"></i>
        <input id="username" name="username" placeholder="用户名" class="auth_input" type="text" value="15S003086"/>
        <span id="usernameError" style="display:none;" class="auth_error">请输入用户名</span>
    </p>
    <p>
        <i class="auth_icon auth_icon_pwd"></i>
        <input id="password" name="password" placeholder="密码" class="auth_input" type="password" value=""
        autocomplete="off"/>
        <span id="passwordError" style="display:none;" class="auth_error">请输入密码</span>
    </p>
    <p id="cpatchaDiv">
    </p>
        <p>
            <input type="checkbox" name="rememberMe" id="rememberMe"/> <label
                onmousedown="javascript:$('.iCheck-helper').click();">一周内免登录</label>
        </p>

    <p>
        <button type="submit" class="auth_login_btn primary full_width">登录
        </button>
    </p>

    <a id="getBackPasswordMainPage" href="getBackPasswordMainPage.do" class="auth_login_forgetp">
    <small>登录遇到问题？找回密码？修改密码？</small>
    </a>

    <input type="hidden" name="lt" value="LT-700416-zfoKA7Bq7fri5zBTQOBeP1QwUOPzqy1489647595111-UAfR-cas"/>
    <input type="hidden" name="dllt" value="userNamePasswordLogin"/>
    <input type="hidden" name="execution" value="e4s5"/>
    <input type="hidden" name="_eventId" value="submit"/>
    <input type="hidden" name="rmShown" value="1">
</form>
```

还可以找到验证码输入框，发现它长这样：

```
<div id="hidenCaptchaDiv" style="display: none;">
    <i class="auth_icon auth_icon_bar"></i>
    <input type="text" placeholder="验证码" id="captchaResponse" name="captchaResponse"
           class="auth_input captcha-input"/>
    <img id="captchaImg" class="captcha-img" src="captcha.html" alt="验证码" title="验证码"/>
            <span style="cursor: pointer;color: #1dadff;margin-left: 2px;" id="changeCaptcha"
                  class="chk_text">换一张</span>
    <span id="cpatchaError" style="display:none;" class="auth_error">请输入验证码</span>
</div>
```

这里面有三个特别重要的信息，分别是：form 的 action；type 为 hidden 的神秘 input；验证码的 src。其内容稍后分解。

【4】用 python 模拟登录，最好先看看浏览器是怎么登陆的。Ctrl+Shift+I，呼出 Chrome 的强大控制台，点击 Network 选项卡，刷新页面，观察浏览器工作过程。

下面是统一身份认证界面的加载流程，第一项 Login 是我要 open 的页面，因为登录只需要提交表单，其他的 js，css 可以忽略了。

浏览器做了这么几件事：1.获取 login 页面(就是 url 指向的页面)；2.获取 login 引用的各种资源：就是下面那一堆。

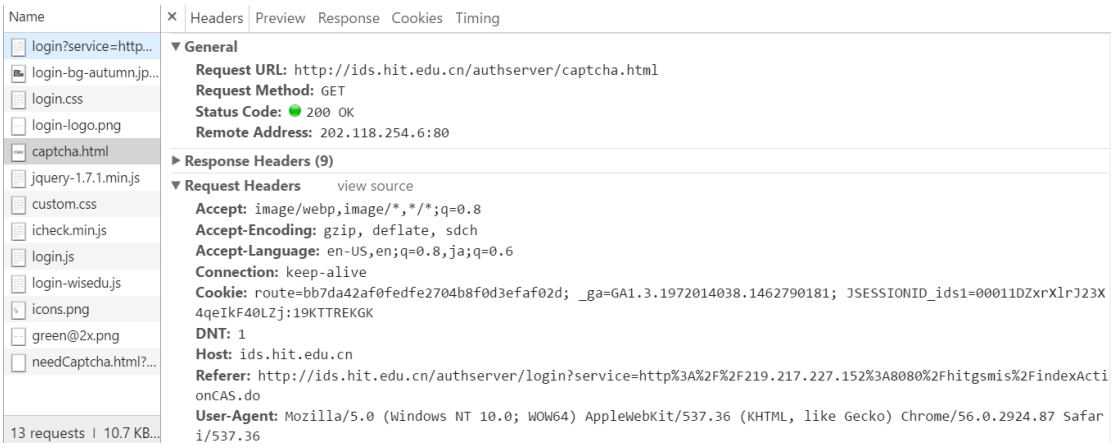
login?service=http%3A%2F%2F219.217.227.152%3A...	200	document	Other	7.9 KB	18 ms
login-bg-autumn.jpg	200	jpeg	login?service=http%3A%2...	(from memory...	0 ms
login.css	200	stylesheet	login?service=http%3A%2...	(from disk cac...	2 ms
login-logo.png	200	png	login?service=http%3A%2...	(from memory...	0 ms
captcha.html	200	jpeg	login?service=http%3A%2...	2.4 KB	24 ms
jquery-1.7.1.min.js	200	script	login?service=http%3A%2...	(from memory...	0 ms
custom.css	200	stylesheet	login?service=http%3A%2...	(from disk cac...	2 ms
icheck.min.js	200	script	login?service=http%3A%2...	(from memory...	0 ms
login.js	200	script	login?service=http%3A%2...	(from memory...	0 ms
login-wisedu.js	200	script	login?service=http%3A%2...	(from memory...	0 ms
icons.png	200	png	jquery-1.7.1.min.js:4	(from memory...	0 ms
green@2x.png	200	png	jquery-1.7.1.min.js:4	(from memory...	0 ms

我们的第一步就是获取 login 页面，下面点开左侧打开详细信息，可以看到 login 的 RequestHeaders。所以我去打开统一认证页面的时候，就使用一样的头（貌似只要有 User-Agent 那项就行，保险我还是都加上了）。

▼ **Request Headers** view source

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,ja;q=0.6
Cache-Control: max-age=0
Connection: keep-alive
Cookie: route=bb7da42af0fedfe2704b8f0d3efaf02d; _ga=GA1.3.1972014038.1462790181; JSESSIONID_ids1=00011DZxrXlrJ23X4qeIkF40LZj:19KTTREKGK
DNT: 1
Host: ids.hit.edu.cn
Referer: http://219.217.227.152:8080/hitgsmis/main.do
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36

captcha.html 是验证码页面，request 的时候头与 login 不同，但使用 login 的头也没关系。captcha.html 非常简单，没有引用任何其他资源，就是简单 Get，返回一张验证码图片。从右侧栏中可以看到，只由 General，Response Headers 和 Request Headers 三项。



下面是点击登录后，浏览器的执行过程。过程很麻烦，第一个是 form 表单提交，302 表示被重定向到了第二个 url。第二项是 Get 了具体的页面，如下图，因此可以推测，在第一个页面完成身份验证后，会根据【1】中的 service 不同重定向到对应页面。

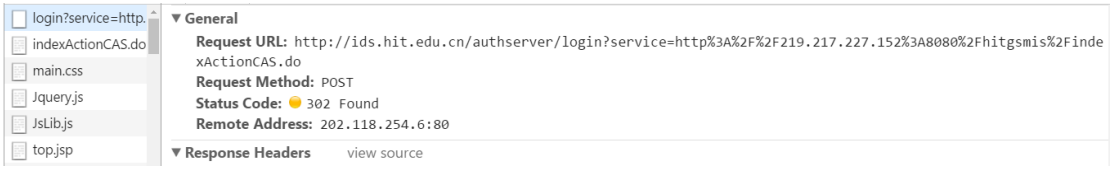
login?service=http%3A%2F%2F219.217.227.152%3A...	302	x-www-form-u...	Other	683 B	55 ms
indexActionCAS.do?ticket=ST-359647-ySq0C69sZZ7...	200	document	http://ids.hit.edu.cn/auths...	1.1 KB	114 ms
main.css	200	stylesheet	indexActionCAS.do?ticket...	(from memory...	0 ms
Jquery.js	200	script	indexActionCAS.do?ticket...	(from memory...	0 ms
JsLib.js	200	script	indexActionCAS.do?ticket...	(from memory...	0 ms
top.jsp	200	document	indexActionCAS.do?ticket...	1.5 KB	8 ms
bottom.jsp	200	document	indexActionCAS.do?ticket...	390 B	9 ms
Left.do	200	document	indexActionCAS.do?ticket...	6.6 KB	47 ms
bottom.jsp	200	document	indexActionCAS.do?ticket...	390 B	21 ms
info.jsp	200	document	indexActionCAS.do?ticket...	2.1 KB	7 ms
bottom.jsp	200	document	indexActionCAS.do?ticket...	390 B	30 ms
bottom.jsp	200	document	indexActionCAS.do?ticket...	390 B	37 ms
main.css	200	stylesheet	top.jsp	(from memory...	0 ms
Jquery.js	200	script	top.jsp	(from memory...	0 ms
JsLib.js	200	script	top.jsp	(from memory...	0 ms

浏览器完整过程

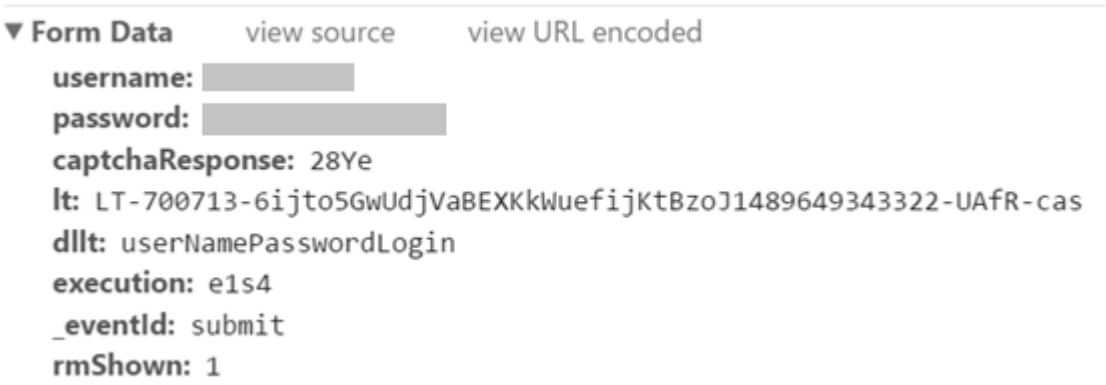
第二项的 General 信息：第一项被重定向到对应的页面（第二项），第二项 Get 了具体页面

第一项的 response headers：这里包含了重定向信息？

可见身份验证工作由第一步完成，与后面无关。打开第一步详细，可见 Post 方法提交表单，request url 是表单里的 action。滚轮向下看一下 FormData 部分，可以看到表单提交的信息。



提交 Form 的 General 信息



Form 的 Data 信息

【5】流程追踪到此结束，总结一下就是：

1. 向 login 界面发送头为 XXX 的 Get 请求，获取页面。
2. 获取 captcha.html，解析验证码。
3. 输入用户名，密码，验证码，Post 给 login 表单中的 action。

看起来很简单吧~

二.实际操作

【1】关于验证码的小秘密

captcha.html 不限制访问，且只返回验证码，每次 Get 内容都不相同。说明这是专门用于申请验证码的页面，但是它是怎么把验证码和登录人的输入关联的？还记得之前奇怪的 type=hidden 的 input 标签吗？仔细看 Form Data 信息图中，它们是不是都在那里。它们是：'lt','dllt','execution','_eventId','rmShown'。

我做实验后发现很可能是这样：

1.浏览器 Get login，服务端返回一个带这些标记字段的页面，标记写在 type=hidden 的 input 标签中。

2.服务端下一个被请求的验证码，会和最近一次返回的 login 页面上的标记字段对应。

3.页面 Post 表单，加入了 label 信息。服务端根据表单中的标记去找对应验证码，再看匹配不匹配。(很可能是 lt 标记，因为 cas 表示身份认证)

缺点是坏人一直刷新验证码，会导致映射错误；网络过慢会导致映射错误。

【2】 解决方案设计

1. 向 login 界面发送头为 XXX 的 Get 请求，获取页面内容 content。
2. 从中解析出'lt','dllt','execution','_eventId','rmShown'它们对应的值，叫做 label_values。
3. 向 captcha.html 发送 Get，获取页面内容，存入本地，并打开。
4. 等待用户输入 username，password，captcha。

5. 根据用户输入和 2 中的 label_values 构造表单，提交给 Form 中 action 指定的 url。
6. 登录成功，查看 cookie，发现多了获取授权的 cookie 了。将登陆成功的 openner 和 cj 返回给调用者，由调用者使用，可以畅通无阻的访问网站了。

【3】代码实现见 LoginHIT.py。