中国科技云通行证网站接入说明

（更新时间：2021 年 12 月 31 日）

1 关于中国科技云通行证

中国科技云通行证是基于中国科技云的统一账号系统，可以用于登录各类科研应用服务，包括：科研在线、文档库、国际会议服务平台、科研主页、中科院邮件系统等，以及今后将逐步扩展的更多应用服务。

UMT 即用户管理工具，是一套能实现用户管理以及单点登录统一身份认证的软件系统，是协同工作环境套件 Duckling 的核心组件之一。

UMT 使用 OAuth 2.0 标准协议实现的单点登录和服务授权的身份认证接口，为其它应用提供用户身份认证、服务授权、单点登录等功能。其最大的特点是安全和集成简单，协议标准，编程语言支持广泛。中国科技云通行证使用 UMT OAuth 2.0 提供应用的接入。

中国科技云通行证是基于 UMT 的统一账号系统。通行证账号包括中科院邮件系统账号以及原 Duckling 通行证账号共计约 20 多万用户账号。

中国科技云通行证为第三方应用提供接入服务，支持三种不同类型的应用接入：

- Web 网站接入
- 移动应用接入
- LDAP 应用接入

本文档为 Web 网站接入的说明文档。

2 Web 网站接入

Web 网站接入是中国科技云通行证针对第三方应用网站提供的网络接入方案。接入中国科技云通行证让您的网站支持用中国科技云通行证账号单点登录，免除用户在您网站注册、登录的烦琐程序，快速为网站增加用户，提升用户体验。

3 接入流程

您的网站要接入中国科技云通行证大体需要四个步骤：

- 提交 Web 网站的接入申请
- 网站接入审核
- 网站接入开发
- 网站部署上线

3.1 提交 Web 网站接入申请

提交 Web 网站接入申请的流程如下：

- 使用通行证账号登录通行证网站（http://passport.escience.cn/）
- 从通行证首页底部的【应用接入->管理应用】进入应用申请与管理页面，其直接访问的链接为：https://passport.escience.cn/user/developer.do?act=display
- 点击【添加新的应用接入】后，在弹出框中填写网站应用相关信息，相关界面为：

点击“添加新的应用接入”按钮，在如下弹出框中填写应用接入信息

- 保存应用接入参数，等待通行证管理员审核。

3.2 网站接入审核

用户提交应用接入的相关参数后，通行证管理员将会收到相应的邮件通知并给予审核，审核成功后，将会返回应用接入所需要的两个参数： client_id 与 client_secret，这两个参数将会在 OAuth 接入开发过程中用到。

您可以在登录通行证后直接通过【应用管理】界面相应的参数信息，如下图所示：

如上图所示：App Key === client_id，App secret === client_secret

3.3 Web 网站接入开发

Web 网站接入开发的基本流程为：

用户的登录过程整体是在中国科技云通证中完成的。其基本原理是将用户重定向到通行证的登录框中，用户在此登录框中完成登录过程。

具体来讲，可按如下步骤进行接入开发：

第一步：

应用在需要用户认证时，将登录链接重定向到通行证的用户登录界面，其 URI 地址如下：

```
https://passport.escience.cn/oauth2/authorize?response_type=code&redirect_uri=YOUR_REGISTERED_REDIRECT_URI&client_id=YOUR_CLIENT_ID&theme=YOUR_THEME
```

其中 repsonse_type=code ； redirect_uri 为下面接受 authorization code 的 servlet 的 URI；

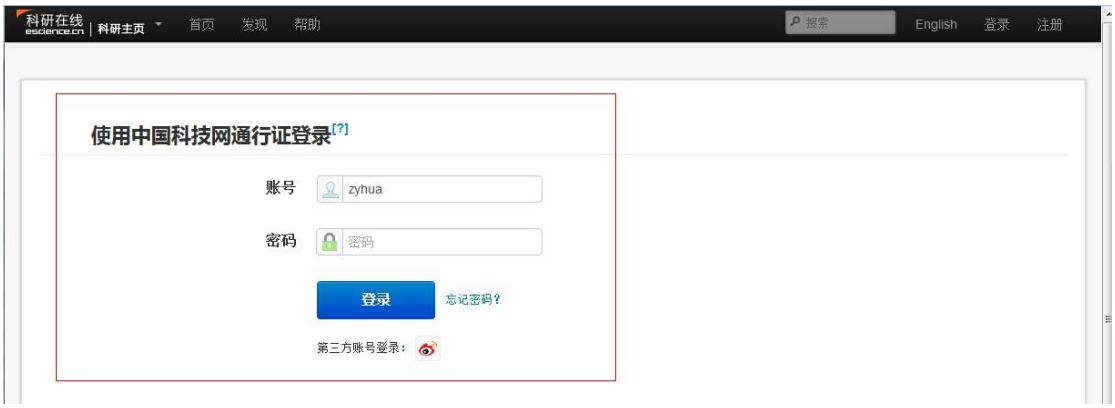
第二步：

用户在通行证登录界面输入通行证账号与密码进行认证。

目前通行证默认支持两种不同类型的接入方式：内嵌式与跳转式

1）内嵌方式

这种方式下通行证只提供输入帐号、密码的部分页面内容。Web 应用拥有对登录界面最大的控制力。这种方式适合那些对登录界面有强烈的显示需求的 Web 应用使用。下图是科研主页使用这种方式接入后的效果图：



下图是经过自定义登录界面后，中科院邮件系统使用这种方式集成以后的效果图：

通行证提供一个简洁的 iframe，应用需将其内嵌到需要登录认证的应用页面中。当用户需要登录认证时，直接在应用页面显示通行证提供的登录界面，并让用户完成登录认证过程。

在应用页面中加入 iframe 具体内容如下：

```
<IFRAME ID="Frame1"
SRC="https://passport.escience.cn/oauth2/authorize?response_type=code&redirect_uri=YOUR_REGISTERED_REDIRECT_URI&client_id=YOUR_CLIENT_ID&theme=embed" >
</IFRAME>
```

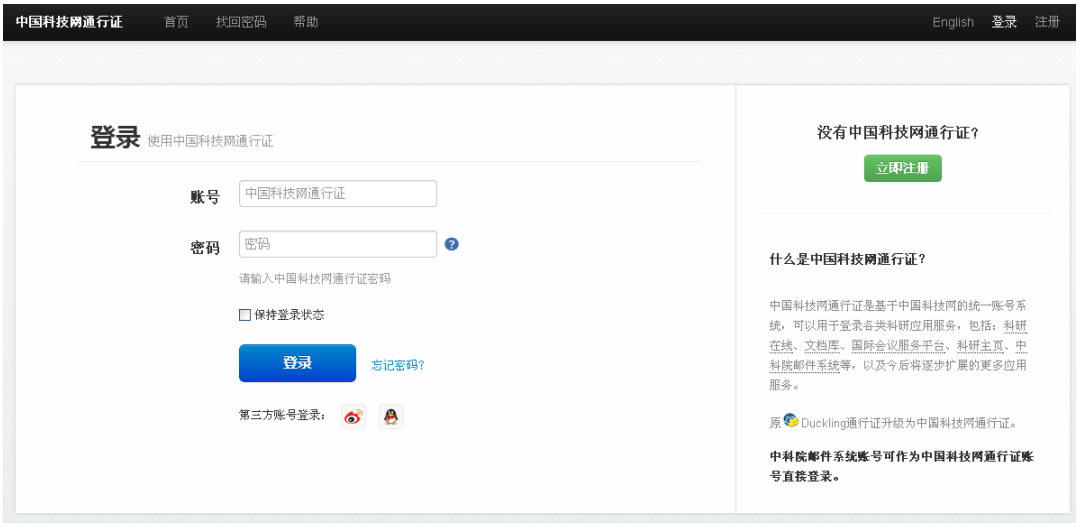
2）跳转方式

这种方式下，登录界面将跳转到通行证来完成。接入的 Web 应用可以很简单的就完成接入开发，而不需要单独的再开发登录界面。

当用户需要认证时，页面需跳转到通行证页面中进行用户身份登录。当登录完成后跳回应用的页面中。为完成上述过程应用需开发一个引导的 servlet 将用户浏览器重定向到如下 URI 中：

https://passport.escience.cn/oauth2/authorize?response_type=code&redirect_uri=YOUR_REGISTERED_REDIRECT_URI&client_id=YOUR_CLIENT_ID&theme=full

下图为通行证的登录界面：



在如下的 URI 中，如将参数：theme=full 换成 theme=simple 时，其登录界面如下图所示 ：

第三步：

用户在通行证登录界面完成账号的验证后，通行证会将浏览器重定向跳转至 YOUR_REGISTERED_REDIRECT_URI/?code=CODE，即配置的 redirect_URI 用于接受 authorization code 的 servlet。接着认证应用就会获取 authorization code。获取 code 后，在应用后台根据下面 URI 去通行证换取 Access_Token：

提交方法：POST
Conten-Type: application/x-www-form-urlencoded
提交参数如下:
client_id: YOUR_CLIENT_ID//客户端 ID
client_secret: YOUR_CLIENT_SECRET
grant_type:authorization_code//固定值
redirect_uri: YOUR_REGISTERED_REDIRECT_URI//申请表单中的回调地址
code:code//第二步中回传 code 的值
提交 URL:
https://passport.escience.cn/oauth2/token

（其中， grant_type=authorization_code； client_secret 为申请参数 code 为第二步中回传 code 的值。）

注意：为了安全，这里的 Code 只能使用一次。应避免在同一次登录中使用同一个 Code 多次访问这个链接。

完成认证并获取认证返回值。返回参数值示例为：

```
{
    "access_token":   "SIAV32hkKG",
    "expires_in":   3600,
    “refresh_token:   ”ASAEDFIkie876”,
    ”userInfo”: {
                    “umtId”:   12,
                    “truename”:   ”yourName”,
                    ” type”:   ”umtauth”,
                    ” securityEmail”: ” securityEmail”,
                    ”cstnetIdStatus”: ”cstnetIdStatus”,
                    ”cstnetId”: ”yourEmail”,
                    “passwordType”:” password_umt”,
                    ”secondaryEmails”:[“youremail1”, “youremail2”]
                }//注意，此处为方便展示，userInfo 值展示为 json 结构，实际返回值为 json 字符串（由双引号""进行包裹）类型，需转换成 json 对象后使用。
}
```

参数说明：

- ✓ umtId：对应 umt 里面的 id 号
- ✓ truename：用户真实姓名
- ✓ type：账户所属范围 umt、coremail、uc
- ✓ securityEmail：密保邮箱
- ✓ cstnetIdStatus：主账户激活状态，即邮箱验证状态， 可选值：active-已激活，temp-未激活。***应用可根据此状态判断是否允许该用户登录***。
- ✓ passwordType: 登录的密码类型
- ✓ cstnetId：用户主邮箱
- ✓ secondaryEmails：辅助邮箱邮箱，暂不开放设置辅助邮箱的 api

根据客户端需求存储 access_token、refresh_token 和用户信息等值。access_token 认证的标识信息可用其访问 UMT 中的其他资源。资源 API 列表请看其他页。refresh_token 当 access_token 过期时可拿此 token 去重新换取 access_token。userInfo 后面进行说明。到此为止登录过程已经完成。

passwordType 是登录时所用的密码类型，接入应用可根据密码类型判断是否允许该登录进入本应用，具体类型如下：

- password_core_mail：已开通中科院邮箱账号的中国科技云通行证账号
- password_umt：在中国科技云通行证平台上自助注册的用户，没有开通中科院邮箱账号。
- Password_third_party_xxx：中国科技云通行证平台已支持与中国科技云通行证绑定的第三方登录账号，xxx 为对应的应用代码，目前支持的第三方登录账号有：
  - geo（password_third_party_geo）：来自于第三方国家地球系统科学数据共享平台的账号；
  - uaf（password_third_party_uaf）：来自于科技云认证联盟（UAF）的账号；
  - cas_hq（password_cas_hq）：来自于第三方院机关统一认证平台的账号。

具体的实现方面可参考《UMT-oauth2-SDK 使用说明（JAVA 版）》或者《UMT-oauth2-SDK 使用说明（PHP 版）》中的实现完成服务器端的实现。

3.4 网站部署上线

在完成通行证网站接入开发后，部署上线，通行证用户即可无缝使用您的网站应用了。

4 桌面客户端接入说明

桌面和移动客户端的开发目前支持内嵌浏览器方式。这种方式下，通过以下几步完成接入：

- 在客户端需要登录的时候弹出一个内嵌的浏览器访问与客户端配合的 Web 端的登录 URL。
- 该 Web 服务器的登录 URL，在用户在访问时，跳转到通行证简洁风格的登录页面上。

提供一个风格简洁的跳转方式的页面实现。只是在登录的时候显示现在认证的应用名称。其提供的 URI 和 Web 应用接入中的跳转方式基本一致，唯一的区别是此时的 theme=simple。

```
https://passport.escience.cn/oauth2/authorize?response_type=code&redirect_uri=YOUR_REGISTERED_REDIRECT_URI&client_id=YOUR_CLIENT_ID&theme=simple
```

简单跳转方式的登录页面：

- 用户登录完成以后，通行证会将重定向回客户端配合的 Web 服务器端，并携带一个一次性的 Code。Web 服务器端可以用这个 Code 获取登录用户的信息。具体的获取方式，请参见用户登录后流程部分所描述的内容。
- Web 服务器端完成登录以后，可以给客户端返回提供一个登录信息的凭证(可以不是通行证的凭证)，用于客户端和 Web 端的通信使用。客户端可以截取内嵌浏览器的返回分析后获得这个凭证。
  这里的实现方式是没有规定的，客户端可以根据自己的需要进行定制。

5 其它

5.1 错误信息说明

授权服务器会按照 OAuth2.0 协议对本请求的请求头部、请求参数进行检验，若请求不合法或验证未通过，授权服务器会返回相应的错误信息，包含参数：error：错误码和 error_description 错误信息描述。

错误码描述：

- ✓ invalid_scope：申请权限错误
- ✓ access_denied：用户中途取消授权
- ✓ unauthorized_client：客户端不识别
- ✓ redirect_uri_mismatch：配置 redirect_URI 和传送 URI 不一致
- ✓ server_error：服务器错误
- ✓ invalid_request：错误的获取 token URL
- ✓ invalid_client：获取 token 是 client_id 错误
- ✓ unsupported_grant_type：授权服务器不支持该类型授权

错误信息的返回方式有两种：

1) 当请求授权 code：https://passport.escience.cn/oauth2/authorize 时出现错误，返回方

式是：跳转到 redirect_uri,并在 uri 的加上 error=errorcodestatus 中附带错误的描述信息。

2) 当请求 access token endpoing: https://passport.escience.cn/oauth2/token 时出现错误，在返回参数中加入 error=errorcodestatus。

5.2 如何退出通行证？

第三方接入应用后，在应用自身退出的同时，也需要退出通行证登录。要退出通行证只需重定向到通行证的退出地址即可。退出地址为：

https://passport.escience.cn/logout

如果需要在退出后返回应用自身的某个链接，则加上 WebServerURL 参数，WebServerURL 参数应该用 UTF-8 编码 encode。

格式：

https://passport.escience.cn/logout?WebServerURL=返回地址

示例：退出以后，需要跳到 vmt 的首页（http://vmt.escience.cn），则可以写成：

https://passport.escience.cn/logout?WebServerURL=http://vmt.escience.cn

5.3 如何判断当前用户登录状态？

在应用开发过程中，需要判断用户在通行证的登录状态，以作相应的处理。要实现此功能，请下载或者直接包含通行证提供的 passport.js 文件，其链接为：

https://passport.escience.cn/js/passport.js

Passport.js 的具体调用形式为：

```
var passport = new Passport(option);
passport.xxx();
```

对外公布的方法及参数说明如下：

```
/**
 * 构造函数
 *@param option: {
 *         umtUrl:'http://passport.escience.cn', //umt 的地址，必填
 *         viewPort:$("#testDiv"),              //显示 message 的地方，可不填
 *         message:'haha,runing...',                 //提示信息的内容，可不填
 *         loginclass:'miaomiao'                //提示信息加载后的，class，可不填
 *         }
 * */
function Passport(option);

/**
* 去验证 cookie 信息，是否已登录
*@return cookie 里面时候有已登录的值
* */
function hasSsoLoginFlag();

/**
* 只判断登陆，不刷新页面,验证 cookie 和 session
* @param callback(flag) 回调函数，flag 为 true 或 false，如果直接 return 会有异步返回
问题
* */
function checkLogin(callback);

/**
* 如果已登录，就去 umt 调一下 login，回刷
* @param returnUrl      登陆成功以后的回调地址
* @param params         是一个 json 对象，里面的参数有
*                       {
*                            target:'none',       //回调时 form 表单提交的 target
*                            appname:'dct',           //应用名称
*                            theme:'ddl'         //除非在 umt 里面有定制页面，否则无意义
*                       }
function checkAndLogin(returnUrl, params)
```