

# 致谢

感谢金智卞工提供帮助完成此文档编写工作

# 介绍

统一身份认证是一套完整独立、高效稳定、安全可靠的集中身份认证和分级授权管理、单人多角色管理平台。其必须提供广泛而灵活的认证服务与接口、信息采集与信息查询功能，实现单点登录服务以确保用户身份认证的准确性和便利性，实现各应用系统用户名和口令的统一；

# LDAP

使用的判断用户是否合法方式为用户和密码验证法，在用户与服务器的通信过程中，采用一种方式去确认对方用户的身份或权限，然后再对用户的不同身份进行分配其访问控制权限。

结合我校现有系统，用 LDAP 技术可以解决不同子系统统一身份认证的问题，利用 LDAP 目录服务集中存储用户身份数据，实现 VPN、上网、无线认证以及应用等认证。

## LDAP 认证过程

LDAP 是轻量目录访问协议 (Lightweight Directory Access Protocol) ；  
LDAP 认证是通过 WSS3.0 加上轻量目录 LDAP 协议搭建的种认证方式，使用

https 加密传输，主要用于做文档管理。LDAP 认证就是把用户数据放在 LDAP 服务器上，通过 LDAP 服务器上的数据对用户进行认证处理。

LDAP 实现原理：每一个登陆，连接请求去发送本地的用户、密码给 LDAP 服务器，然后在 LDAP 服务器上进行匹配，然后判断是否可以通过认证。

LDAP 优点：

1). LDAP 数据库对读操作进行优化的种数据库，在读写比例大于 7 比 1 的情况下，LDAP 会体现出高的性能。

2). 更灵活添加数据类型，LDAP 是根据 schema 的内容定义各种属性之间的从属关系及匹配模式的。

例如：在传统的结构化数据库 mysql 中添加一个字段，就需要在用户表中添加一个字段。但是在数据量

极大的时候是很耗时间的，效率低，用户体验差，但是 LDAP 只需要在 Schema 中加入新的属性，不会

由于用户的属性增多而形象查询性能。

3). LDAP 是个开放的标准协议，不同于一般的 SQL 数据库，LDAP 的客户端是跨平台的，方便简洁。

4). 在存储上 LDAP 是以树形结构存储数据，任何一个分支都可以单独在服务器中进行分布式管理，

不仅有利于服务器的负载均衡，还方便做跨区域的服务器部署。

5). LDAP 支持强认证方式，可以达到很高的安全级别，根据 UTF-8 编码。

LDAP 接口主要面向瞬时认证并发要求非常高的应用系统，最典型的就是高校的 VPN、无线网认证。利用 LDAP 接口实现应用系统认证集成以后，可实现高效的统一认证。

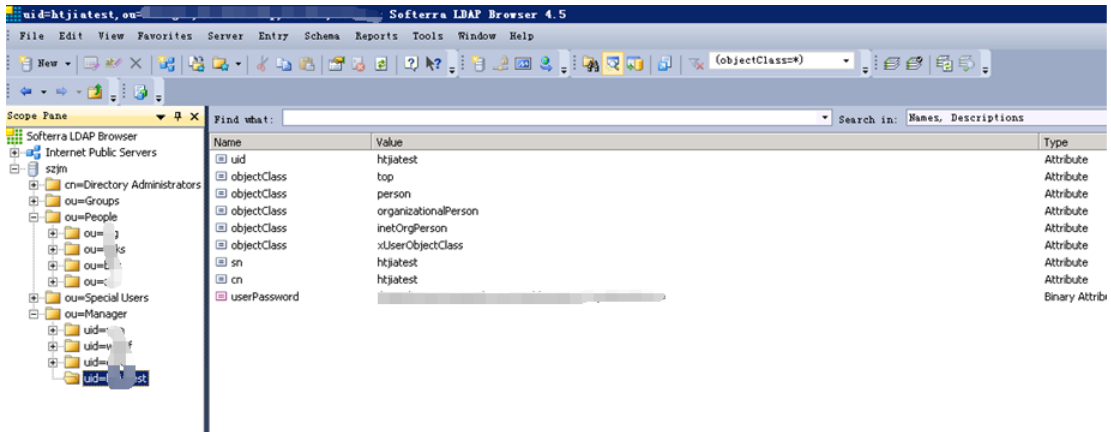
`uid=htjiatest,ou=Manager,dc=authldap,dc=edu,dc=cn`

ldap://XXX.XXX.XXX.XXX:389/uid=htjiatest,ou=Manager,dc=authldap,dc=edu,dc=cn

DC (Domain Component):所在控制域

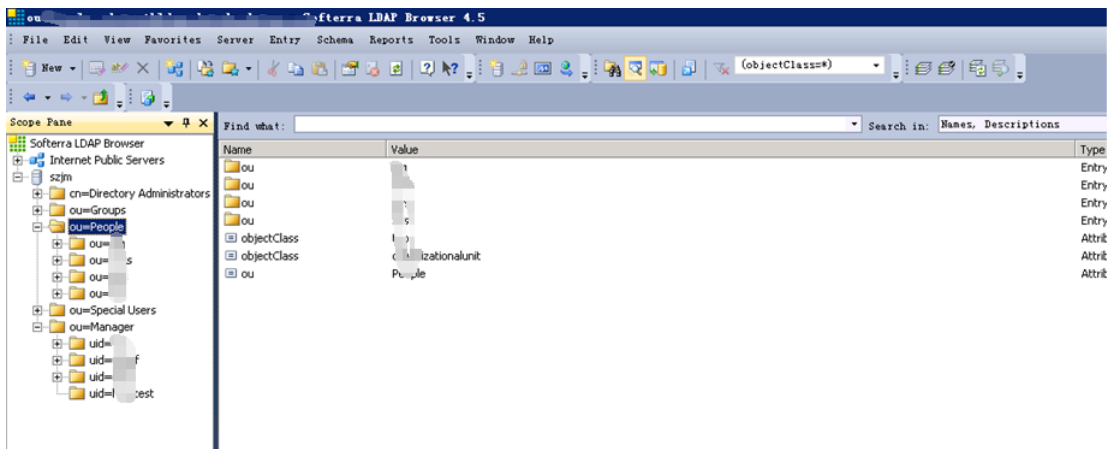
OU (Organizational Unit):组织单元

CN (Common Name):通用名称



在校生成

ldap://XXX.XXX.XXX.XXX:389/ou=zxs,ou=People,dc=authldap,dc=edu,dc=cn



系统后台维护部分



概况 帐号管理 认证管理 授权管理 审计管理 监控管理 系统管理

操作日志 同步任务日志 历史日志 配置管理 应用帐号 平台版本展示 API调用日志

### 应用帐号

添加应用帐号信息，您可以 [返回](#) 列表页面

应用帐号\*：

应用名称\*：

应用权限：

密码：

密码确认：

密码策略： [?](#)

## LDAP 教职工同步过程

select \* from V\_JZG --差异视图

(基本表和差异表比对以后的表)

```
CREATE OR REPLACE VIEW V_JZG
(id, name, password, sfzx, actiontype)
AS
SELECT t.id, t.name, t.password, '1', 'add'
FROM (SELECT a.gh AS id,
             a.XM AS name,
             '88888888' AS password
      FROM t_hr_jzg a
      where a.gh <> 'ampadmin') t
WHERE NOT EXISTS (SELECT c.id FROM V_JZG_BASIC c WHERE t.id = c.id)
```

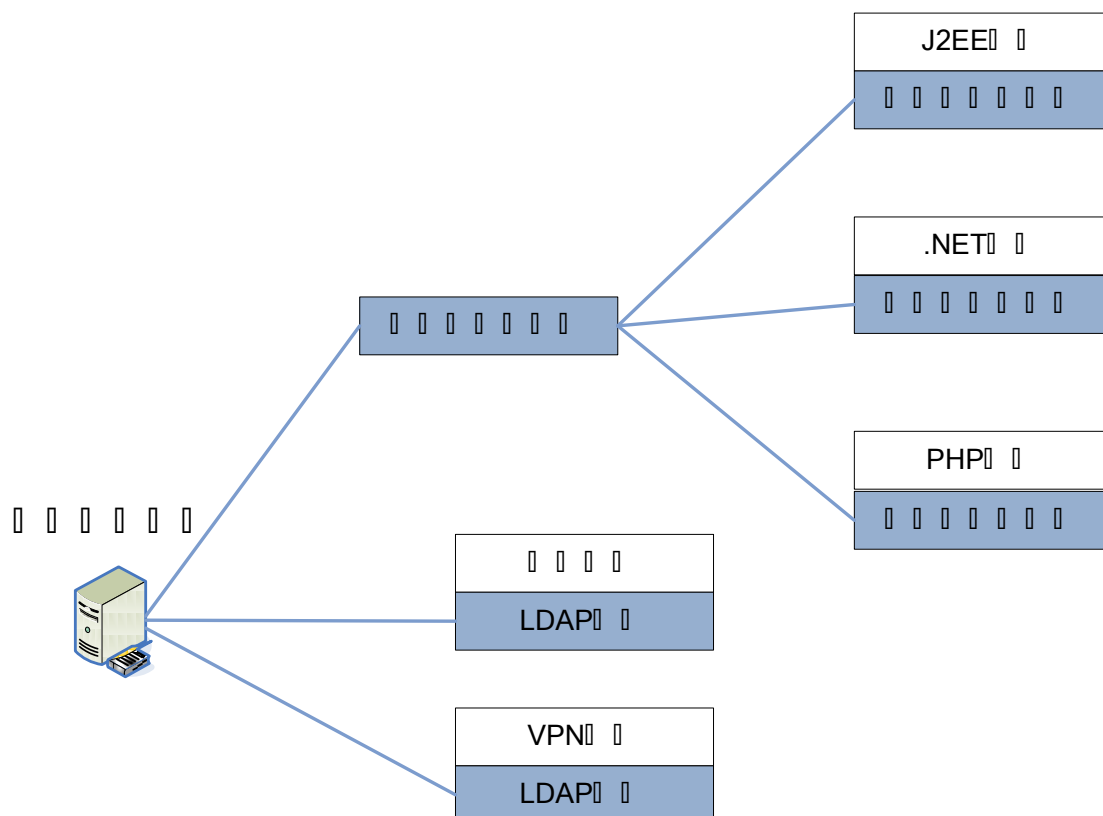
select \* from V\_JZG\_BASIC --基本表

(ldap 里面一致的表)

select \* from t\_hr\_jzg --标准表(数据标准):

(人事系统同步过来的标准表)





代理认证配置完后，均需至统一身份管理平台授权访问。先登录身份认证管理平台，在认证管理的认证应用模板，添加需代理认证的应用；添加完应用后，需给相应的组或者帐号授权，允许其访问该应用。

管理界面首先，认证服务添加应用，如下图所示：

The screenshot shows the '认证应用' (Authentication Application) configuration page. The page includes a navigation bar with tabs like '概况', '帐号管理', '认证管理', '授权管理', '审计管理', '监控管理', and '系统管理'. The '认证管理' tab is selected, and the '认证应用' sub-tab is active.

The main content area is titled '认证应用' and contains a form for adding a new application. The form fields include:

- 应用名称**: testjhtchina
- 应用URL**: http://127.0.0.1:8080/\*
- SecretKey**: (empty field with a note: \*只有secretKey非空的情况下，才可以进行重置，否则会带来第三方集成不便)
- 属性**: 用户编码, 姓名, 密码 (with a note: (点击添加属性))
- 是否用于代理**: 是 (selected), 否
- 是否激活**: 是 (selected), 否
- SAML 版本**: 不支持, 1.1 (selected)
- 描述**: (empty text area)
- 所属厂商**: test1 (with a note: \*可以输入厂商简短语，比如：金智，方正，学校自建等，可以自行输入)
- 客户端语言**: java (with a note: \*可以输入客户端语言简短语，比如：java，donet，php等，可以自行输入)
- 语言版本**: jdk1.8 (with a note: \*可以输入版本的简短语，比如：jdk1.5, .NetFramework4.0，php5等，可以自行输入)
- 应用ip地址**: (empty field with a note: 多个以逗号分隔例如 127.0.0.1, 127.0.0.2)
- 校验地址**: (empty field with a note: 可选，需要支持第三方自有用户校验方式的，需要填写此url，校验并且获取用户信息，实现此功能，请参阅“第三方自由用户集成文档”)

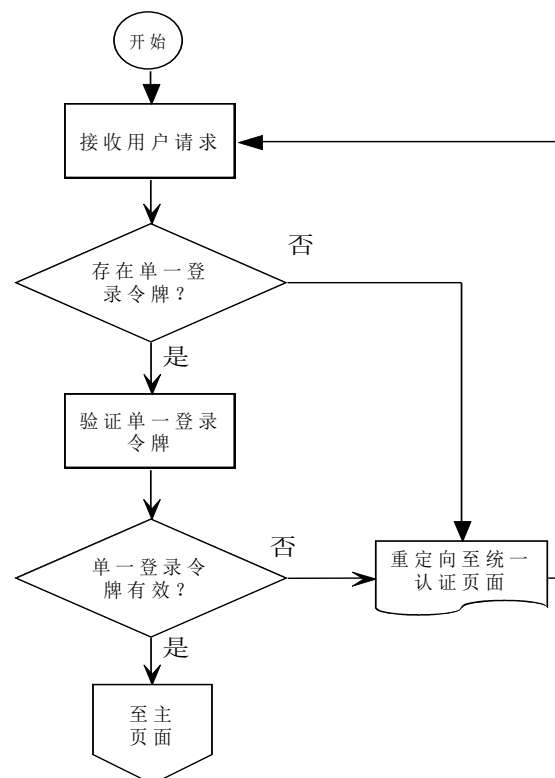
At the bottom of the form, there are buttons for '提交' (Submit) and '返回' (Return).

然后添加用户授权



代码部分实现原理:

- (1) 拷贝提供的 jar
- (2) 修改 web.xml 文件
- (3) 获取用户信息
- (4) 集成应用退出
- (5) 认证接口工作过程



# OAuth 2.0 协议

## sample

### (1) Oauth 注册应用

概况

帐号管理

认证管理

授权管理

审计管理

监控管理

系统管理

认证应用

认证统计

应用日志管理

OAuth应用注册

OAuth服务管理

OAuth注册应用管理

编辑OAuth应用信息，您可以返回列表页面

应用名称：

test123

客户端ID：

客户端秘钥：

客户端授权回调地址：

http://192.168.1.100:8080/oauth2/callback

基础授权范围：

获取个人基本信息 + (点击添加组)

扩展授权范围：

获取用户的手机号码 + (点击添加组)

提交

返回

### (2)认证应用，添加应用

概况

帐号管理

认证管理

授权管理

审计管理

监控管理

系统管理

认证应用

认证统计

应用日志管理

OAuth应用注册

OAuth服务管理

认证应用

您可以添加应用添加移动应用，还可以输入：应用名称：应用URL：应用类型：请选择是否激活：请选择

应用ID	应用名称	应用URL	应用类型	是否激活	描述	编辑	删除
661034742	test	http://192.168.1.100:8080/oauth2/callback	Web应用	是	科研		
7899	汇	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
355	汇	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
7	OAuth	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
9	教	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
78	vp	https://192.168.1.100:8080/oauth2/callback	Web应用	是			
57	管	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
42	tes	http://192.168.1.100:8080/oauth2/callback	Web应用	是	测试		
58	vp	https://192.168.1.100:8080/oauth2/callback	Web应用	否			
48	云	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
37	管	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
35	管	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
58	管	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
73	管	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
56	管	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
2	管	http://192.168.1.100:8080/oauth2/callback	Web应用	是			
4	管	http://192.168.1.100:8080/oauth2/callback	Web应用	是	修改密码		



概况 帐号管理 认证管理 授权管理 审计管理 监控管理 系统管理

认证应用 认证统计 应用日志管理 OAuth应用注册 OAuth服务管理

### 认证应用

编辑应用信息，您可以 [返回](#) 列表页面

应用名称:

应用URL:

SecretKey: 86S7rL9dm1UdycJnSIHlUxhA3snEk \*只有secretKey泄露的情况下，才可以进行重置，否则会带来第三方集成不便

属性: 用户编码 ☒ 姓氏 ☒ 姓名 ☒ 密码 ☒ + (点击添加属性)

是否用于代理: 是 ☐ 否 ☒

是否激活: 是 ☒ 否 ☐

SAML 版本: 不支持 ☒ 1.1 ☐

描述:

所属厂商:  \*可以输入厂商简短语，比如：金智，方正，学校自建等，可以自行输入

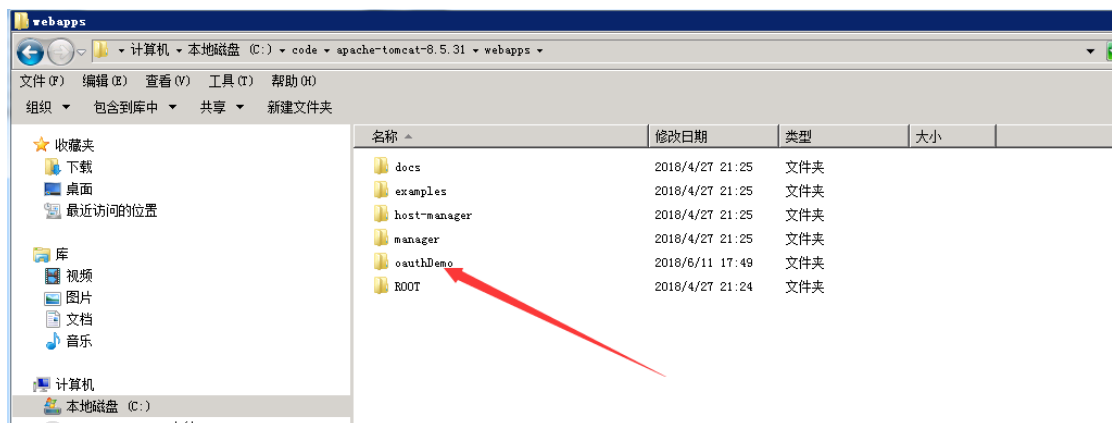
客户端语言: java \*可以输入客户端语言简短语，比如：java，donet，php等，可以自行输入

语言版本: jdk1.6 \*可以输入版本的简短语，比如：jdk1.5，NetFramework4.0，php5等，可以自行输入

应用ip地址:  多个以逗号分隔例如: 127.0.0.1, 127.0.0.2

校验地址:  可选，需要支持第三方自有用户校验方式的，需要填写此url，校验并且获取用户信息，实现此功能，请参考集成文档

## 首先部署项目



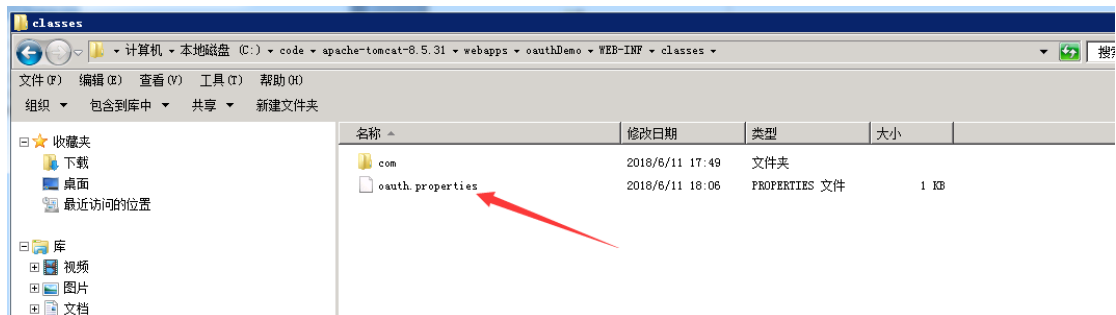
## 修改配置文件

Web.xml ( /oauthDemo/WEB-INF/web.xml )



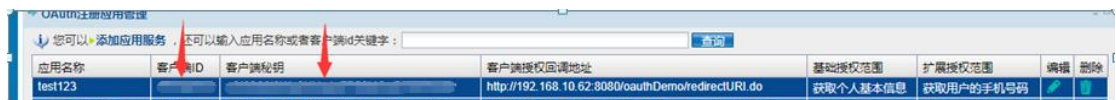
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns="http://java.sun.com/xml/ns/javaee"
3         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
5         http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
6         version="2.5">
7
8     <context-param>
9         <param-name>logoutUrl</param-name>
10        <param-value>http://authserver.szjw.edu.cn/authserver/logout?service=http://192.168.10.62:8080/oauthDemo/index.do</param-value>
11    </context-param>
12
13    <servlet>
14        <servlet-name>index</servlet-name>
15        <servlet-class>com.wisedu.oauth.IndexServlet</servlet-class>
16    </servlet>
17
18    <servlet>
19        <servlet-name>redirectURI</servlet-name>
20        <servlet-class>com.wisedu.oauth.RedirectUIServlet</servlet-class>
21    </servlet>
22
23    <servlet-mapping>
24        <servlet-name>index</servlet-name>
25        <url-pattern>/index.do</url-pattern>
26    </servlet-mapping>
27
28    <servlet-mapping>
29        <servlet-name>redirectURI</servlet-name>
30        <url-pattern>/redirectURI.do</url-pattern>
31    </servlet-mapping>
32
33    </web-app>
34
35
36
```

修改 oauth.properties ( oauthDemo/WEB-INF/classes/ oauth.properties )



```
1 #回调地址
2 redirect_uri=http://
3
4 #OAuth服务端的配置
5 oauth_server=http://
6 client_id=
7 client_secret=
8
9 #第二个OAuth服务配置
10 cas.oauth_server=http://
11 cas.client_id=
12 cas.client_secret=
13
```

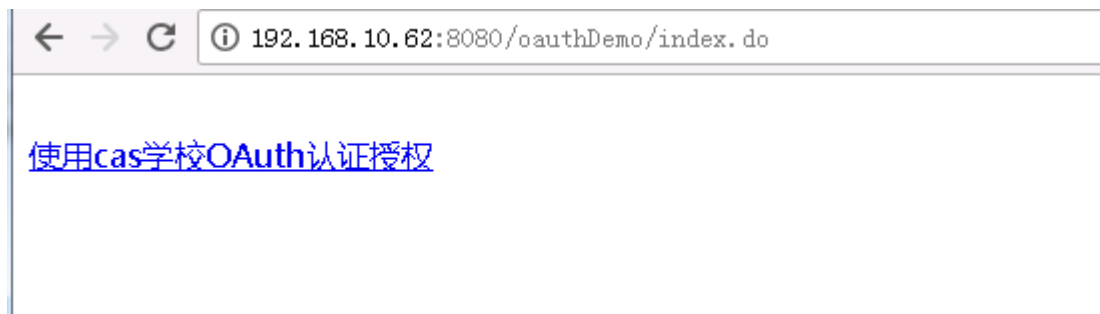
这里配置文件的 client\_id, cas.client\_id 和 client\_secret , cas.client\_secret



启动 tomcat 服务

运行效果如下：

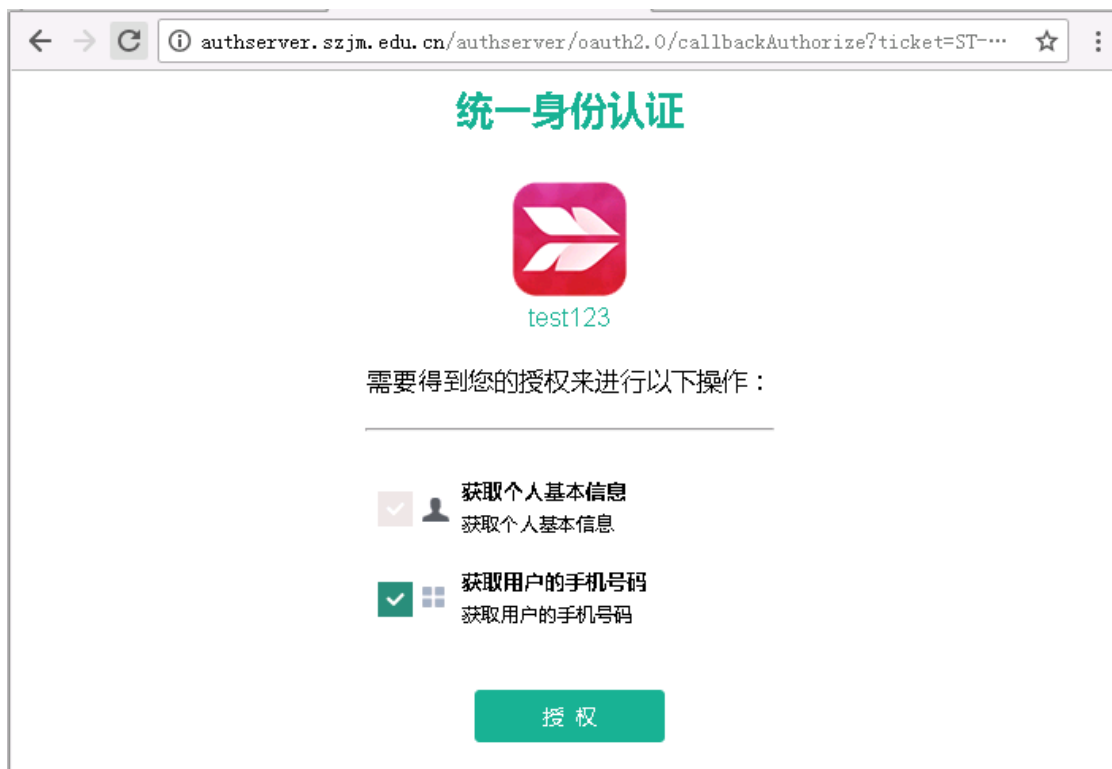
<http://192.168.10.62:8080/oauthDemo/index.do>



进入登陆界面



输入用户名和密码



## 获取授权信息



## 名词定义

- (1) Third-party application: 第三方应用程序，本文中又称“客户端”（client），即本例程序“Hello World”。
- (2) HTTP service: HTTP 服务提供商，本文中简称“服务提供商”，即例子中的 IDS。
- (3) Resource Owner: 资源所有者，本文中又称“用户”（user）。
- (4) User Agent: 用户代理，本文中就是指浏览器。
- (5) Authorization server: 认证服务器，即服务提供商专门用来处理认证的服务器。

(6) Resource server: 资源服务器，即服务提供商存放用户生成的资源的服务器。它与认证服务器，可以是同一台服务器，也可以是不同的服务器。

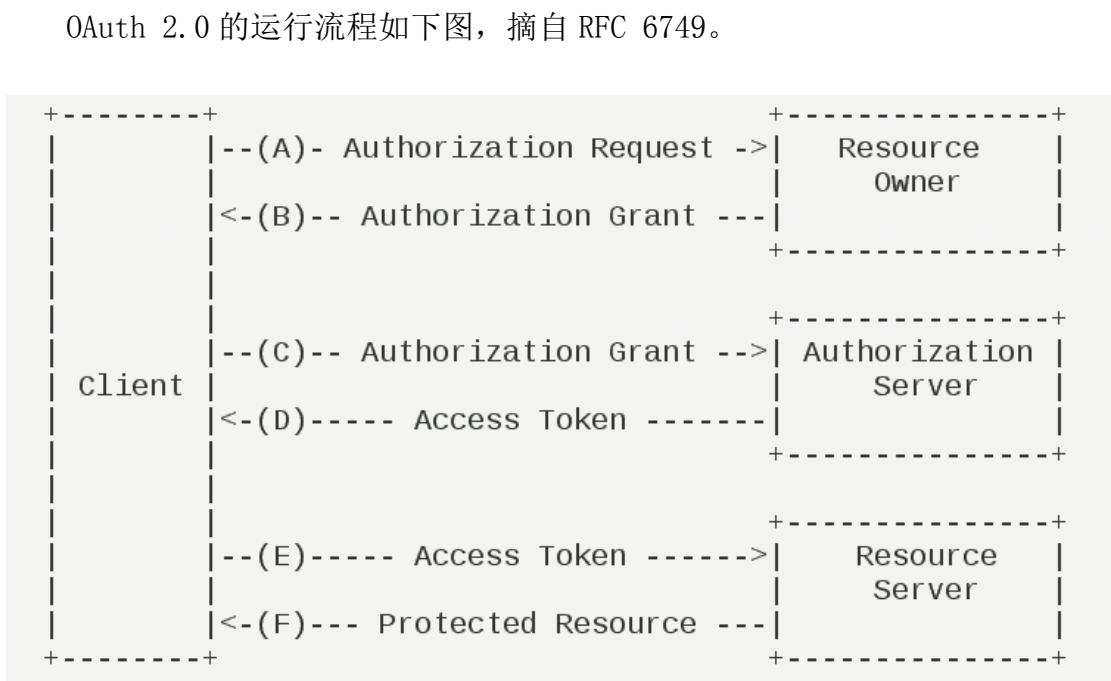
OAuth 的作用就是让“客户端”安全可控地获取“用户”的授权，与“服务商提供商”进行互动。

## OAuth 的思路

OAuth 在“客户端”与“服务提供商”之间，设置了一个授权层（authorization layer）。“客户端”不能直接登录“服务提供商”，只能登录授权层，以此将用户与客户端区分开来。“客户端”登录授权层所用的令牌（token），与用户的密码不同。用户可以在登录的时候，指定授权层令牌的权限范围和有效期。

“客户端”登录授权层以后，“服务提供商”根据令牌的权限范围和有效期，向“客户端”开放用户储存的资料。

## 运行流程



Resource Owner:资源拥有者

Authorization Servicer:验证服务器

Resource Service :资源服务器

- (A) 用户打开客户端以后，客户端要求用户给予授权。
- (B) 用户同意给予客户端授权。
- (C) 客户端使用上一步获得的授权，向认证服务器申请令牌。
- (D) 认证服务器对客户端进行认证以后，确认无误，同意发放令牌。
- (E) 客户端使用令牌，向资源服务器申请获取资源。
- (F) 资源服务器确认令牌无误，同意向客户端开放资源。

## 参考文献

OAuth2.0 参考

[http://www.ruanyifeng.com/blog/2014/05/oauth\\_2\\_0.html](http://www.ruanyifeng.com/blog/2014/05/oauth_2_0.html)

<https://www.cnblogs.com/flashsun/p/7424071.html>