**品优购电商系统开发**

**第15章**

**单点登录解决方案-CAS**

传智播客.黑马程序员

# 课程目标

目标1：搭建单点登录服务端，开发单点登录客户端

目标2：完成CAS用户登录信息从数据库

目标3：完成CAS编写品优购登录页面

目标4：完成用户中心单点登录

# 单点登录CAS入门

## 1.1 什么是单点登录

**[单点登录](http://baike.baidu.com/item/%E5%8D%95%E7%82%B9%E7%99%BB%E5%BD%95" \t "http://baike.baidu.com/_blank)（Single Sign On），简称为 SSO**，是目前比较流行的企业业务整合的解决方案之一。SSO的定义是在多个应用系统中，用户只需要登录一次就可以访问所有相互信任的应用系统。

我们目前的系统存在诸多子系统，而这些子系统是分别部署在不同的服务器中，那么使用传统方式的session是无法解决的，我们需要使用相关的单点登录技术来解决。



## 1.2 什么是CAS

CAS(**Central Authentication Service中央身份认证服务**)是 Yale 大学发起的一个开源项目，旨在为 Web 应用系统提供一种可靠的单点登录方法，CAS 在 2004 年 12 月正式成为 JA-SIG 的一个项目。CAS 具有以下特点：

【1】开源的企业级单点登录解决方案。

【2】CAS Server 为需要独立部署的 Web 应用。

【3】CAS Client 支持非常多的客户端(这里指单点登录系统中的各个 Web 应用)，包括 Java、.Net、PHP、Perl、Ruby 等。

从结构上看，CAS 包含两个部分： CAS Server 和 CAS Client。CAS Server 需要独立部署，主要负责对用户的认证工作；CAS Client 负责处理对客户端受保护资源的访问请求，需要登录时，重定向到 CAS Server。

**TGT（Ticket Grangting Ticket）**：TGT是CAS为用户签发的登录票据，存入Session

**ST（Service Ticket）**：ST是CAS为用户签发的访问某一服务的票据，回传给CAS客户端(只使用一次，10秒过期)，作为该服务在CAS服务器的唯一标识符。

**TGC（Ticket Grangting Cookie）**：TGC是CAS在浏览器存储用户登录凭证的Cookie

**CAS 最基本的协议过程：**

**第一种情况：不存在ST服务票据**

**CAS访问流程说明：**

1. **访问服务：用户发送请求访问CAS客户端(应用系统)受保护的服务资源。**
2. **获取服务票据：CAS客户端从CAS服务端获取ST服务票据。**

**3. 重定向认证：没有ST服务票据，说明用户还没有进行身份认证，于是，重定向用户请求到CAS服务端进行身份认证，并把用户此次访问的请求URL作为参数(参数名为service)传递给CAS服务端。**

**4. 用户认证：CAS服务端接收到身份认证请求后转向登录页面，用户提供认证信息后进行身份认证。身份认证成功后，CAS服务器往浏览器写入TGC的Cookie，再在Session中存储TGT登录票据。**

**5. 发放票据：CAS服务端产生一个随机的ST，携带ST重定向到CAS客户端。**

**6. 验证票据：CAS客户端收到ST后，向CAS服务端验证服务票据ST的合法性。**

**7. 用户信息：CAS服务端验证ST服务票据通过后，回传登录用户名给客户端。**

**第二种情况：存在ST服务票据**

**CAS访问流程说明：**

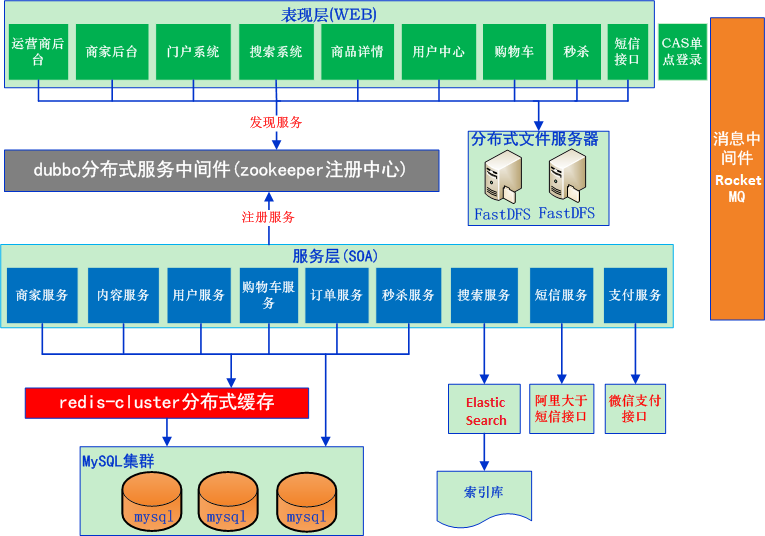
1. **访问服务：用户发送请求访问CAS客户端(应用系统)受保护的服务资源。**
2. **获取服务票据：CAS客户端从CAS服务端获取ST服务票据。**

**3. 用户认证：CAS服务端从用户浏览器中获取TGC的Cookie，再根据该Cookie的值从Session中获取TGT登录票据，如果有代表该用户已登录，产生ST服务票据，并返回给CAS客户端。**

**4. 验证票据：CAS客户端收到ST后，向CAS服务端验证服务票据ST的合法性。**

**5. 用户信息：CAS服务端验证ST服务票据通过后，回传登录用户名给客户端。**

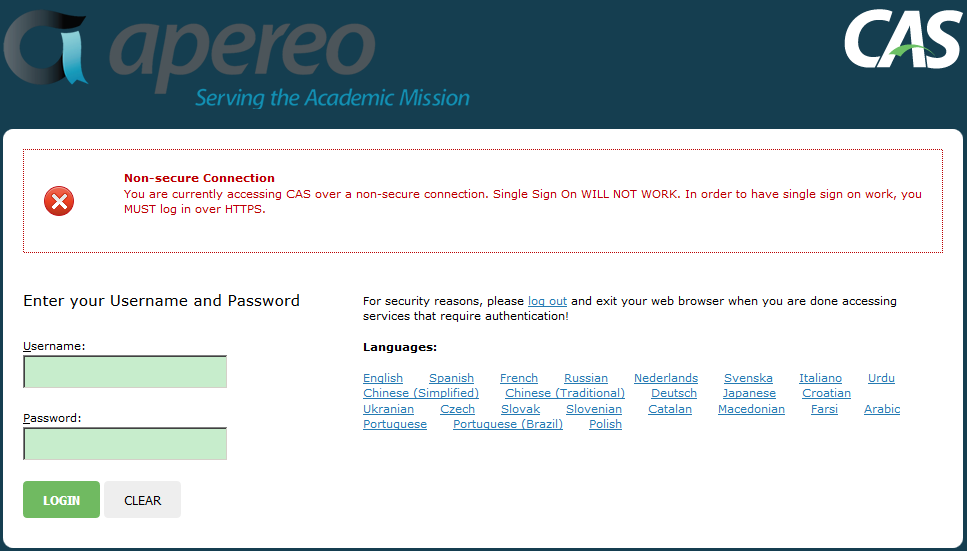
# 搭建单点登录服务端



## 2.1 CAS服务端部署

CAS服务端其实就是一个war包。

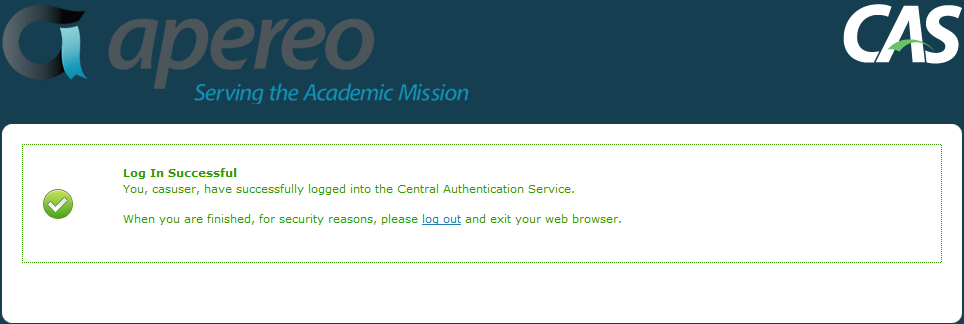
在资料\CAS单点登录\cas-server-webapp-4.2.7.war将其改名为cas.war放入tomcat目录下的webapps下。启动tomcat自动解压war包，改名为ROOT。浏览器输入http://localhost:8080/，可看到登录页面：



不要嫌弃这个页面丑，我们后期可以再提升它的颜值。暂时把注意力放在功能实现上。

这里有个固定的用户名和密码：casuser / Mellon。

登录成功后会跳到登录成功的提示页面：



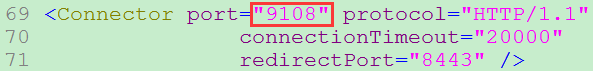
## 2.2 CAS服务端配置

CAS核心配置文件：/WEB-INF/cas.properties

### 2.2.1修改tomcat端口

如果我们不希望用8080端口访问CAS, 可以修改端口，打开tomcat 目录 conf\server.xml找到下面的配置

http端口9108、服务端口：8006、AJP连接端口：8010







修改WEB-INF/cas.properties(第1行)

|  |
| --- |
| server.name=http://localhost:9108 |

### 2.2.2配置用户名与密码

修改WEB-INF/cas.properties(第270行)

|  |
| --- |
| accept.authn.users=admin::123456 |

### 2.2.3去除https认证

CAS默认使用的是HTTPS协议，如果使用HTTPS协议需要SSL安全证书（需向特定的机构申请和购买） 。如果对安全要求不高或是在开发测试阶段，可使用HTTP协议。我们这里讲解通过修改配置，让CAS使用HTTP协议。

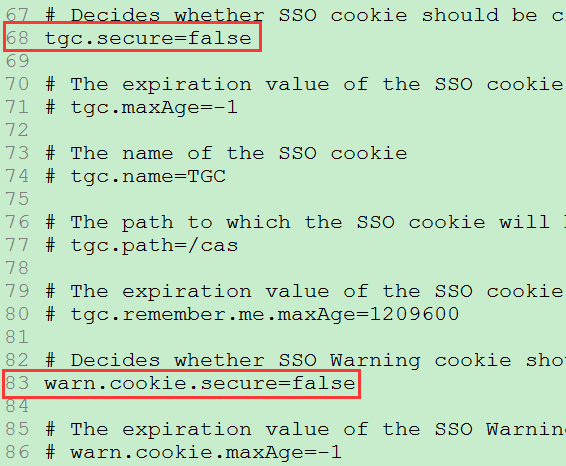
1. 修改WEB-INF/cas.properties(第68行)

|  |
| --- |
| # 决定是否只在安全连接下创建SSO cookie  tgc.secure=false |

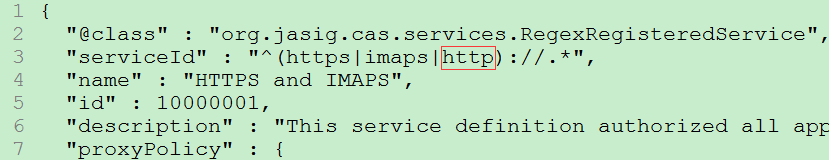
1. 修改WEB-INF/cas.properties(第83行)

|  |
| --- |
| # 决定是否应该仅在安全连接下创建SSO警告cookie  warn.cookie.secure=false |

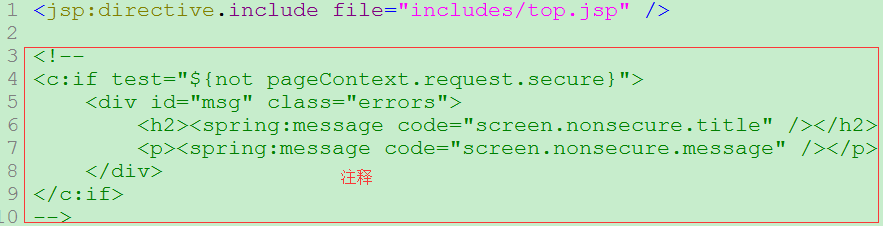
修改后效果如下：



1. 修改WEB-INF\classes\services\HTTPSandIMAPS-10000001.json



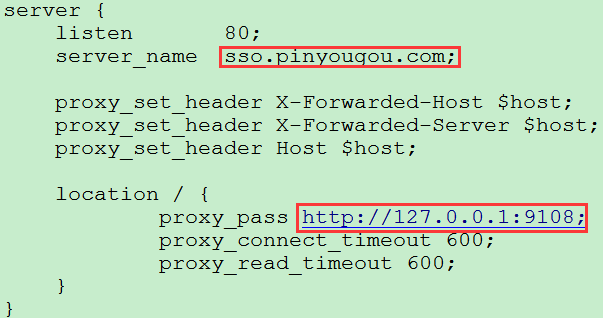
1. 修改WEB-INF\view\jsp\default\ui\casLoginView.jsp



### 2.2.4配置域名访问

修改hosts：127.0.0.1 sso.pinyougou.com

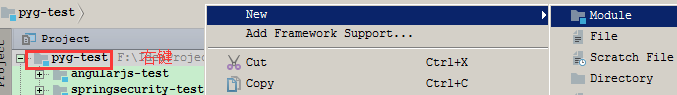
修改nginx.conf：

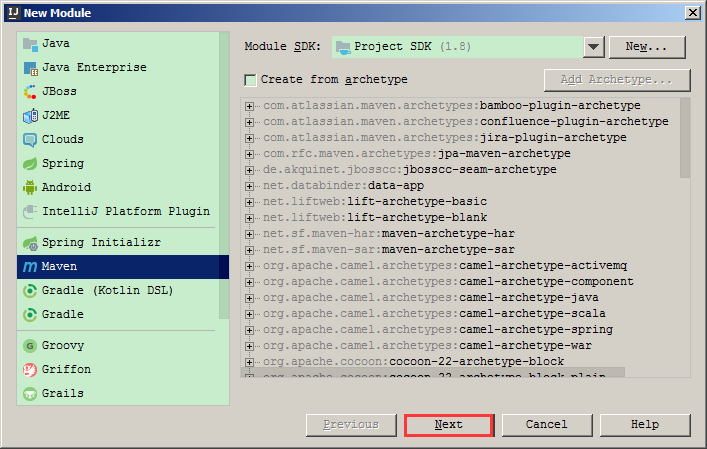


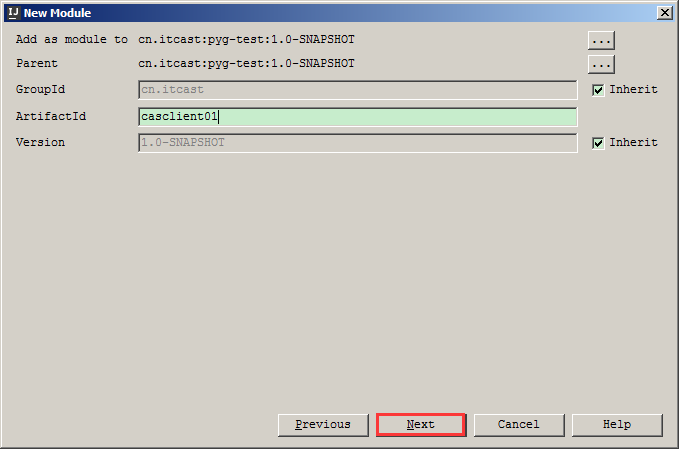
# 开发单点登录客户端

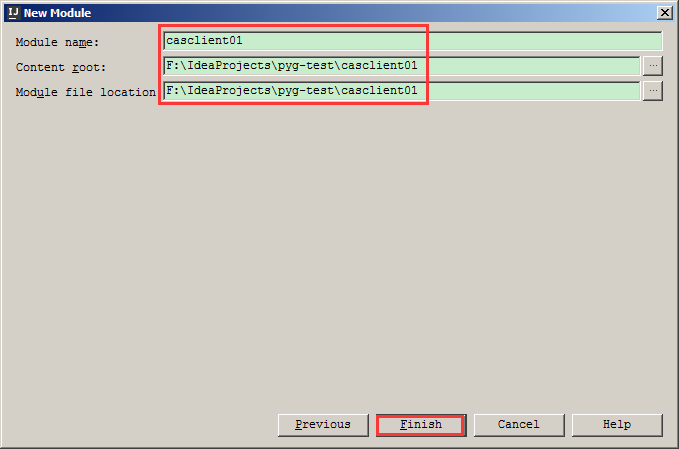
## 3.1客户端工程1搭建

切换到pyg-test项目，创建casclient01模块(war)：









注意：需转化成Web项目

casclient01/pom.xml，配置依赖：

|  |
| --- |
| <**project xmlns="http://maven.apache.org/POM/4.0.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  http://maven.apache.org/maven-v4\_0\_0.xsd"**>  <**parent**>  <**artifactId**>pyg-test</**artifactId**>  <**groupId**>cn.itcast</**groupId**>  <**version**>1.0-SNAPSHOT</**version**>  </**parent**>  <**modelVersion**>4.0.0</**modelVersion**>  <**artifactId**>casclient01</**artifactId**>  <**packaging**>war</**packaging**>  <**name**>casclient01</**name**>  *<!-- 配置依赖关系 -->* <**dependencies**>  *<!-- cas-client-core -->* <**dependency**>  <**groupId**>org.jasig.cas.client</**groupId**>  <**artifactId**>cas-client-core</**artifactId**>  <**version**>3.4.1</**version**>  </**dependency**>  *<!-- servlet-api、jsp-api、el-api-->* <**dependency**>  <**groupId**>org.apache.tomcat.embed</**groupId**>  <**artifactId**>tomcat-embed-jasper</**artifactId**>  <**version**>8.5.16</**version**>  <**scope**>provided</**scope**>  </**dependency**>  </**dependencies**>  <**build**>  <**plugins**>  <**plugin**>  <**groupId**>org.apache.tomcat.maven</**groupId**>  <**artifactId**>tomcat7-maven-plugin</**artifactId**>  <**configuration**>  *<!-- 指定端口 -->* <**port**>9001</**port**>  *<!-- 请求路径 -->* <**path**>/</**path**>  </**configuration**>  </**plugin**>  </**plugins**>  </**build**> </**project**> |

**CAS客户端四个核心过滤器：**

**SingleSignOutFilter：单点退出过滤器(用户退出)**

**AuthenticationFilter：身份认证过滤器(登录才能访问的资源)**

**Cas20ProxyReceivingTicketValidationFilter：票据验证过滤器(验证ST)**

**HttpServletRequestWrapperFilter：请求包裹过滤器(获取登录用户名)**

casclient01/src/main/webapp/WEB-INF/web.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0"**>  *<!-- 配置单点退出过滤器 (可选配置) -->* <**filter**>  <**filter-name**>ssoutFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.session.SingleSignOutFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>ssoutFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!-- 配置身份认证过滤器(必须启用) -->* <**filter**>  <**filter-name**>authenticationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.authentication.AuthenticationFilter</**filter-class**>  <**init-param**>  *<!-- 配置CAS服务端登录地址 -->* <**param-name**>casServerLoginUrl</**param-name**>  <**param-value**>http://sso.pinyougou.com/login</**param-value**>  </**init-param**>  <**init-param**>  *<!-- 配置服务名称 -->* <**param-name**>serverName</**param-name**>  <**param-value**>http://localhost:9001</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>authenticationFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!-- 该过滤器负责对ticket的校验工作(必须启用) -->* <**filter**>  <**filter-name**>ticketValidationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  *<!-- 配置服务名称 -->* <**init-param**>  <**param-name**>serverName</**param-name**>  <**param-value**>http://localhost:9001</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>ticketValidationFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!--  配置HttpServletRequest请求包裹过滤器 (可选配置)  说明：通过HttpServletRequest的getRemoteUser()方法获取SSO登录用户名  -->* <**filter**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.util.HttpServletRequestWrapperFilter</**filter-class**>  </**filter**>  <**filter-mapping**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**> </**web-app**> |

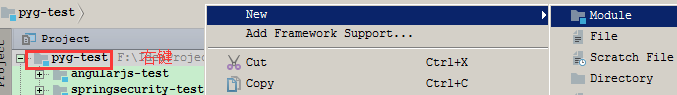
casclient01/src/main/webapp/index.jsp：

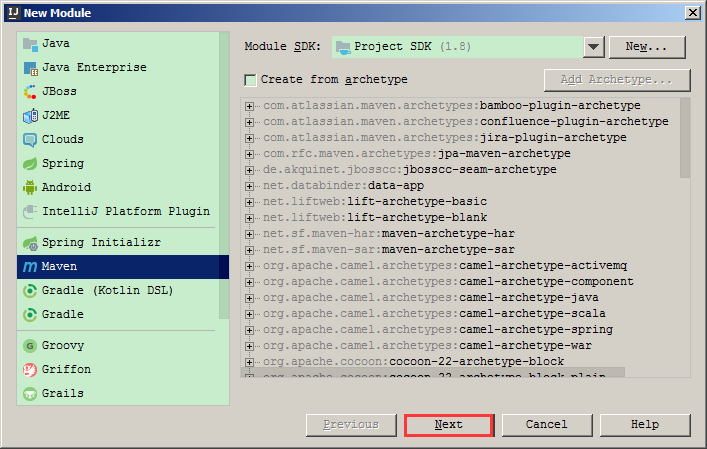
|  |
| --- |
| <%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**"%> <**html**> <**head**>  <**title**>品优购</**title**> </**head**> <**body**>  欢迎来到一品优购：**<%=**request.getRemoteUser()**%>** </**body**> </**html**> |

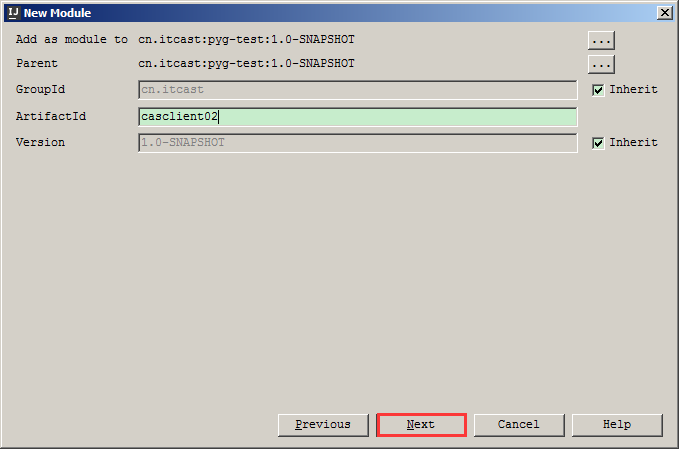
说明：request.getRemoteUser() 获取登录用户名.

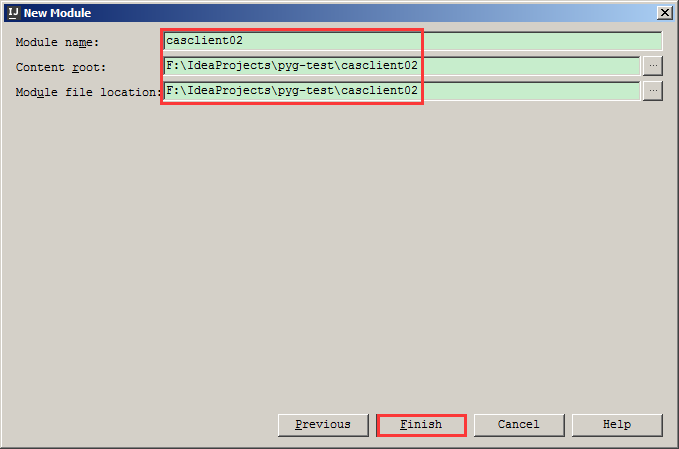
## 3.2客户端工程2搭建

切换到pyg-test项目，创建casclient02模块(war)：









**注意：需转化成Web项目**

casclient02/pom.xml，配置依赖：

|  |
| --- |
| <**project xmlns="http://maven.apache.org/POM/4.0.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  http://maven.apache.org/maven-v4\_0\_0.xsd"**>  <**parent**>  <**artifactId**>pyg-test</**artifactId**>  <**groupId**>cn.itcast</**groupId**>  <**version**>1.0-SNAPSHOT</**version**>  </**parent**>  <**modelVersion**>4.0.0</**modelVersion**>  <**artifactId**>casclient02</**artifactId**>  <**packaging**>war</**packaging**>  <**name**>casclient02</**name**>  *<!-- 配置依赖关系 -->* <**dependencies**>  *<!-- cas-client-core -->* <**dependency**>  <**groupId**>org.jasig.cas.client</**groupId**>  <**artifactId**>cas-client-core</**artifactId**>  <**version**>3.4.1</**version**>  </**dependency**>  *<!-- servlet-api、jsp-api、el-api -->* <**dependency**>  <**groupId**>org.apache.tomcat.embed</**groupId**>  <**artifactId**>tomcat-embed-jasper</**artifactId**>  <**version**>8.5.16</**version**>  <**scope**>provided</**scope**>  </**dependency**>  </**dependencies**>  <**build**>  <**plugins**>  <**plugin**>  <**groupId**>org.apache.tomcat.maven</**groupId**>  <**artifactId**>tomcat7-maven-plugin</**artifactId**>  <**configuration**>  *<!-- 指定端口 -->* <**port**>9002</**port**>  *<!-- 请求路径 -->* <**path**>/</**path**>  </**configuration**>  </**plugin**>  </**plugins**>  </**build**> </**project**> |

casclient02/src/main/webapp/WEB-INF/web.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0"**>   *<!-- 配置单点退出过滤器 (可选配置) -->* <**filter**>  <**filter-name**>ssoutFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.session.SingleSignOutFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>ssoutFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!-- 配置身份认证过滤器(必须启用) -->* <**filter**>  <**filter-name**>authenticationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.authentication.AuthenticationFilter</**filter-class**>  <**init-param**>  *<!-- 配置CAS服务端登录地址 -->* <**param-name**>casServerLoginUrl</**param-name**>  <**param-value**>http://sso.pinyougou.com/login</**param-value**>  </**init-param**>  <**init-param**>  *<!-- 配置服务名称 -->* <**param-name**>serverName</**param-name**>  <**param-value**>http://localhost:9002</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>authenticationFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!-- 该过滤器负责对ticket的校验工作(必须启用) -->* <**filter**>  <**filter-name**>ticketValidationFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  *<!-- 配置服务名称 -->* <**init-param**>  <**param-name**>serverName</**param-name**>  <**param-value**>http://localhost:9002</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>ticketValidationFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!--  配置HttpServletRequest请求包裹过滤器 (可选配置)  说明：通过HttpServletRequest的getRemoteUser()方法获取SSO登录用户名  -->* <**filter**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.util.HttpServletRequestWrapperFilter</**filter-class**>  </**filter**>  <**filter-mapping**>  <**filter-name**>httpServletRequestWrapperFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**> </**web-app**> |

casclient02/src/main/webapp/index.jsp：

|  |
| --- |
| <%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**"%> <**html**> <**head**>  <**title**>品优购</**title**> </**head**> <**body**>  欢迎来到二品优购：**<%=**request.getRemoteUser()**%>** </**body**> </**html**> |

## 3.3单点登录测试

第一步：启动cas部署的tomcat

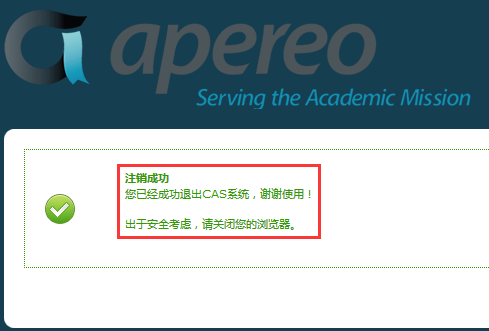
第二步：启动客户端工程1和 客户端工程2

第三步：地址栏输入http://localhost:9001/ 和 http://localhost:9002/，地址均会跳转到CAS登录页

第四步：输入用户名和密码后，页面跳转回9002，再次访问9001也可以打开主页面

## 3.4单点退出登录

地址栏输入：[http://sso.pinyougou.com/logout](http://localhost:9000/cas/logout)



我们可以将这个链接添加到index.jsp：

|  |
| --- |
| <**br**/><**a href="http://sso.pinyougou.com/logout"**>退出登录</**a**> |

但我们更希望退出登录后，能自动跳转到某个页面，那如何处理呢？

修改cas系统的配置文件WEB-INF/cas.properties(第111行)：

|  |
| --- |
| cas.logout.followServiceRedirects=true |

改为true后，可以在退出时跳转页面到目标页面，修改index.jsp的退出链接：

|  |
| --- |
| <**a href="http://sso.pinyougou.com/logout?service=http://www.taobao.com"**>退出登录</**a**> |

# 4.CAS服务端数据源设置

## 4.1需求分析

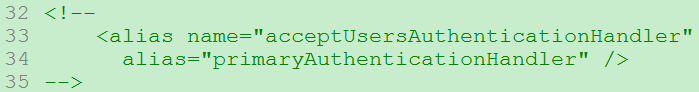
我们现在让用户名密码从我们的品优购的tb\_user表里做验证

## 4.2配置数据源

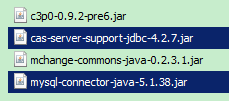
修改cas服务端中WEB-INF/deployerConfigContext.xml，最后添加如下配置：

|  |
| --- |
| *<!-- 配置数据源 -->* <**bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"  p:driverClass="com.mysql.jdbc.Driver"  p:jdbcUrl="jdbc:mysql://localhost:3306/pinyougou\_db"  p:user="root"  p:password="root"  p:maxPoolSize="10"  p:minPoolSize="2"  p:initialPoolSize="2"**/>  *<!-- 配置密码加密方式 -->* <**bean id="passwordEncoder"  class="org.jasig.cas.authentication.handler.DefaultPasswordEncoder"  c:encodingAlgorithm="MD5"  p:characterEncoding="UTF-8"**/>  *<!-- 配置身份认证处理器 -->* <**bean id="primaryAuthenticationHandler"  class="org.jasig.cas.adaptors.jdbc.QueryDatabaseAuthenticationHandler"  p:dataSource-ref="dataSource"  p:sql="select password from tb\_user where username = ?"  p:passwordEncoder-ref="passwordEncoder"**/> |

注释掉默认身份认证处理器(deployerConfigContext.xml配置文件第32行)：



将下面二个jar包拷贝到webapps\ROOT\WEB-INF\lib目录下：



说明：这四个jar包在【资料\CAS单点登录\lib】目录下，c3p0相关jar包cas项目中已自带，用数据库中的用户名和密码进行测试。

# 5.CAS服务端登录界面改造

## 5.1需求分析

我们现在动手将CAS默认的登录页更改为自己的品优购登录页面

## 5.2登录页面

### 5.2.1拷贝资源

* 将品优购的【资料\登录静态原型】登录页login.html拷贝到系统WEB-INF\view\jsp\default\ui 目录下。
* 将【资料\登录静态原型】css、img、plugins三个文件夹拷贝到ROOT目录下。
* 将原来的WEB-INF\view\jsp\default\ui\casLoginView.jsp 改名（可以为之后的修改操作做参照），将login.html改名为casLoginView.jsp

### 5.2.2修改页面

拷贝WEB-INF\view\jsp\default\ui\includes\top.jsp页面中的指令到WEB-INF\view\jsp\default\ui\casLoginView.jsp：

|  |
| --- |
| <%@ **page pageEncoding**="**UTF-8**"%> <%@ **page contentType**="**text/html; charset=UTF-8**"%> <%@ **taglib prefix**="**c**" **uri**="**http://java.sun.com/jsp/jstl/core**"%> <%@ **taglib prefix**="**spring**" **uri**="**http://www.springframework.org/tags**"%> <%@ **taglib prefix**="**form**" **uri**="**http://www.springframework.org/tags/form**"%> <%@ **taglib prefix**="**fn**" **uri**="**http://java.sun.com/jsp/jstl/functions**"%> |

casLoginView.jsp页面，修改form标签：

commandName: 数据封装模型对象

htmlEscape:是否转义html标识符，如<是否转义为&lt;

|  |
| --- |
| <**form:form method="post" id="fm1"   commandName="${**commandName**}"   class="sui-form"**>  ...... </**form:form**> |

casLoginView.jsp页面，修改用户名框：

accessKey: 接收的快捷键

path:相当于name属性

autocomplete:自动填充

placeholder:预提示信息

tabindex:获取焦点顺序

|  |
| --- |
| <**form:input id="username"  path="username"  placeholder="邮箱/用户名/手机号"  class="span2 input-xfat"**/> |

casLoginView.jsp页面，修改密码框：

|  |
| --- |
| <**form:password id="password"  path="password"  placeholder="请输入密码"  class="span2 input-xfat"**/> |

casLoginView.jsp页面，修改登录按钮：

|  |
| --- |
| <**input type="hidden" name="execution" value="${**flowExecutionKey**}"**/> <**input type="hidden" name="\_eventId" value="submit"**/> <**input class="sui-btn btn-block btn-xlarge btn-danger"  value="登&nbsp;&nbsp;录"  type="submit"** /> |

修改后效果如下：



## 5.3错误提示

在casLoginView.jsp页面，表单<**form:form>**内加入错误提示框：

|  |
| --- |
| <**form:errors path="\*" id="msg" cssClass="errors"  element="div" htmlEscape="false"**/> |

测试：输入错误的用户名和密码，提示是英文。这个提示信息是在WEB-INF\classes目录下的messages.properties文件中(55、56、64、65行)

|  |
| --- |
| username.required=Username is a required field.  password.required=Password is a required field.  authenticationFailure.AccountNotFoundException=Invalid credentials.  authenticationFailure.FailedLoginException=Invalid credentials. |

设置国际化为zn\_CN，修改WEB-INF\cas.properties(145行)：

|  |  |
| --- | --- |
| locale.default=zh\_CN |  |

修改WEB-INF\classes\messages\_zh\_CN.properties，改为中文提示68、69行（**native2ascii**转码）第一个：用户名不正确！ 第二个：用户登录失败！

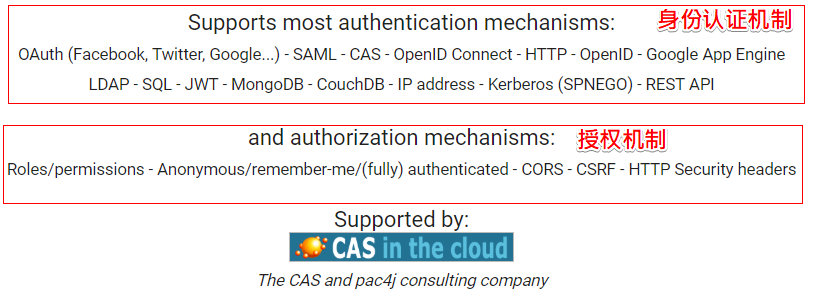
|  |
| --- |
| authenticationFailure.AccountNotFoundException=\u7528\u6237\u540d\u4e0d\u6b63\u786e\uff01  authenticationFailure.FailedLoginException=\u7528\u6237\u767b\u5f55\u5931\u8d25\uff01 |

# Pac4j安全引擎

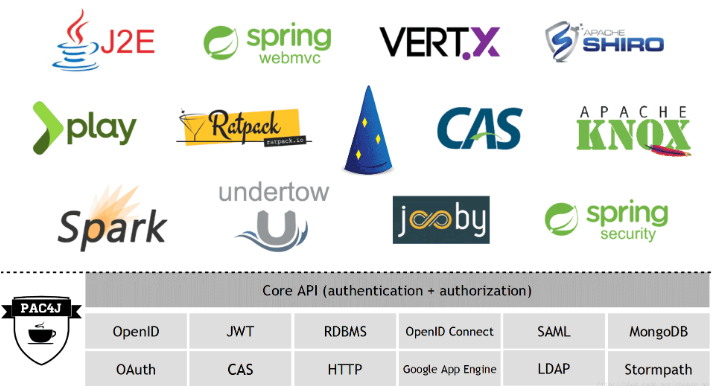
Pac4j是一个简单而强大的安全引擎，用于Java对用户进行身份验证、管理授权，以确保web应用程序安全。它提供了一套完整的概念和组件。它基于Java 8，并在Apache 2许可下使用。它可用于大多数框架/工具和支持大多数认证/授权机制。

官方网站：<http://www.pac4j.org/>





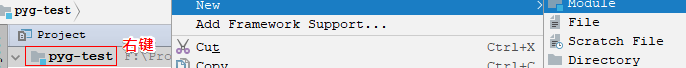
**已经集成的框架：(身份验证机制与授权机制)**

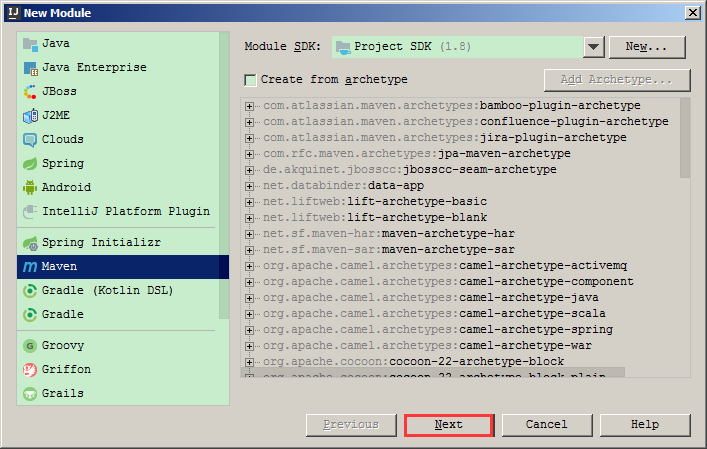


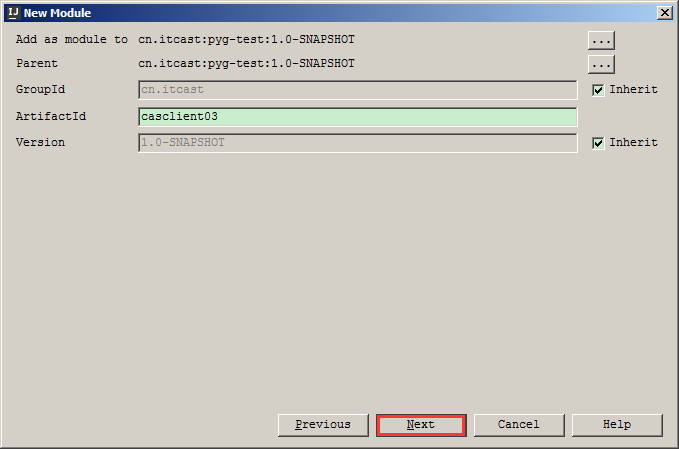
# 7.Pac4j整合CAS

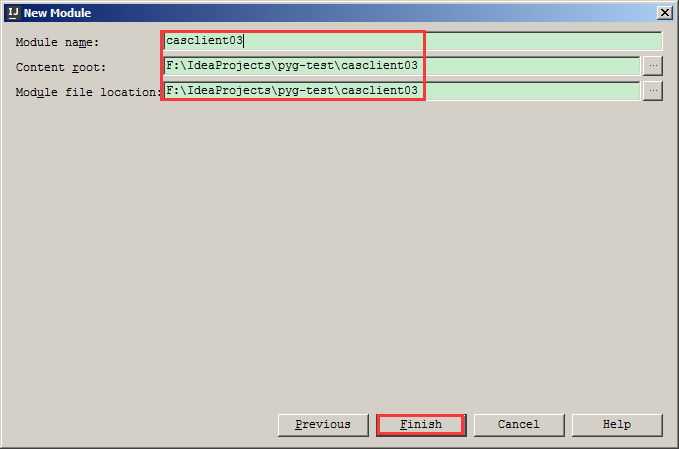
## 7.1 Shiro测试工程

在pyg-test项目中，创建casclient03模块(war)









**注意：需转化成Web项目**

casclient03/pom.xml，添加配置：

|  |
| --- |
| <**project xmlns="http://maven.apache.org/POM/4.0.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  http://maven.apache.org/maven-v4\_0\_0.xsd"**>  <**parent**>  <**artifactId**>pyg-test</**artifactId**>  <**groupId**>cn.itcast</**groupId**>  <**version**>1.0-SNAPSHOT</**version**>  </**parent**>  <**modelVersion**>4.0.0</**modelVersion**>  <**artifactId**>casclient03</**artifactId**>  <**packaging**>war</**packaging**>  <**name**>casclient03</**name**>    *<!-- 配置依赖关系 -->* <**dependencies**>  *<!-- servelt-api、jsp-api、el-api -->*  <**dependency**>  <**groupId**>org.apache.tomcat.embed</**groupId**>  <**artifactId**>tomcat-embed-jasper</**artifactId**>  <**version**>8.5.16</**version**>  <**scope**>provided</**scope**>  </**dependency**>  *<!-- Spring -->* <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-webmvc</**artifactId**>  <**version**>5.1.3.RELEASE</**version**>  </**dependency**>  *<!-- jackson -->*  <**dependency**>  <**groupId**>com.fasterxml.jackson.core</**groupId**>  <**artifactId**>jackson-databind</**artifactId**>  <**version**>2.9.8</**version**>  </**dependency**>  *<!-- shiro-spring -->* <**dependency**>  <**groupId**>org.apache.shiro</**groupId**>  <**artifactId**>shiro-spring</**artifactId**>  <**version**>1.4.0</**version**>  </**dependency**>  </**dependencies**>  <**build**>  <**plugins**>  <**plugin**>  <**groupId**>org.apache.tomcat.maven</**groupId**>  <**artifactId**>tomcat7-maven-plugin</**artifactId**>  <**version**>2.2</**version**>  <**configuration**>  *<!-- 指定端口 -->* <**port**>9003</**port**>  *<!-- 请求路径 -->* <**path**>/</**path**>  </**configuration**>  </**plugin**>  </**plugins**>  </**build**> </**project**> |

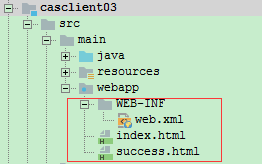
casclient03/src/main/webapp/WEB-INF/web.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0"**>   *<!-- 配置Spring核心监听器 -->* <**context-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:applicationContext-\*.xml</**param-value**>  </**context-param**>  <**listener**>  <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**>  </**listener**>   *<!-- 配置Spring委派过滤器代理(要求Spring容器中有了个bean的id为shiroFilter) -->* <**filter**>  <**filter-name**>shiroFilter</**filter-name**>  <**filter-class**>org.springframework.web.filter.DelegatingFilterProxy</**filter-class**>  </**filter**>  <**filter-mapping**>  <**filter-name**>shiroFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**> </**web-app**> |

提供casclient03/src/main/resources/applicationContext-shiro.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd"**>   *<!-- 配置shiro作为Spring委派过滤器(shiro过滤器工厂) -->* <**bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean"**>  *<!-- 配置注入安全管理对象 -->* <**property name="securityManager" ref="securityManager"**/>  *<!-- 配置登录页面访问URL -->* <**property name="loginUrl" value="/login.html"**/>  *<!-- 配置权限管理URL过滤器链(配置顺序为自上而下) -->* <**property name="filterChainDefinitions"**>  <**value**>  /css/\*\* = anon  /logout = logout  /\*\* = user  </**value**>  </**property**>  </**bean**>   *<!-- 配置安全管理器 -->* <**bean id="securityManager"  class="org.apache.shiro.web.mgt.DefaultWebSecurityManager"**>  *<!-- 配置自定义身份认证域-->* <**property name="realm" ref="authorizingRealm"**/>  </**bean**> </**beans**> |

在casclient03/src/main/webapp/目录下，添加html页面：



## 7.2 Pac4j集成CAS

casclient03/pom.xml引入依赖：

|  |
| --- |
| *<!-- buji-pac4j -->* <**dependency**>  <**groupId**>io.buji</**groupId**>  <**artifactId**>buji-pac4j</**artifactId**>  <**version**>4.1.0</**version**> </**dependency**> *<!-- pac4j-cas -->* <**dependency**>  <**groupId**>org.pac4j</**groupId**>  <**artifactId**>pac4j-cas</**artifactId**>  <**version**>3.6.1</**version**> </**dependency**> |

casclient03/src/main/resources/applicationContext-shiro.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd"**>   *<!-- 配置shiro作为Spring委派过滤器(shiro过滤器工厂) -->* <**bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean"**>  *<!-- 配置注入安全管理对象 -->* <**property name="securityManager" ref="securityManager"**/>  *<!-- 配置过滤器 -->* <**property name="filters"**>  <**map**>  *<!-- 1. 安全过滤器，拦截需要登录的URL -->* <**entry key="security"**>  <**bean class="io.buji.pac4j.filter.SecurityFilter"**>  <**property name="config" ref="config"**/>  </**bean**>  </**entry**>  *<!-- 2. 回调过滤器，完成ticket验证 -->* <**entry key="callback"**>  <**bean class="io.buji.pac4j.filter.CallbackFilter"**>  <**property name="config" ref="config"**/>  *<!-- 验证通过后默认重定向URL -->* <**property name="defaultUrl" value="http://127.0.0.1:9003"**/>  </**bean**>  </**entry**>  *<!-- 3. 退出过滤器，拦截需要退出的URL -->* <**entry key="logout"**>  <**bean class="io.buji.pac4j.filter.LogoutFilter"**>  <**property name="config" ref="config"**/>  *<!-- 中央退出 -->* <**property name="centralLogout" value="true"**/>  *<!-- 本地退出 -->* <**property name="localLogout" value="true"**/>  *<!-- 退出成功后,默认重定向URL -->* <**property name="defaultUrl"  value="http://127.0.0.1:9003/success.html"**/>  </**bean**>  </**entry**>  </**map**>  </**property**>  *<!-- 配置URL过滤器链(配置顺序为自上而下) -->* <**property name="filterChainDefinitions"**>  <**value**>  /success.html = anon  /callback = callback  /logout = logout  /\*\* = security  </**value**>  </**property**>  </**bean**>   *<!-- pac4j配置 -->* <**bean id="config" class="org.pac4j.core.config.Config"**>  <**constructor-arg name="client" ref="casClient"**/>  </**bean**>  *<!-- 配置CAS客户端 -->* <**bean id="casClient" class="org.pac4j.cas.client.CasClient"**>  *<!-- 设置cas服务端信息 -->* <**property name="configuration" ref="casConfiguration"**/>  *<!-- 登录成功后,重定向回来的请求URL -->* <**property name="callbackUrl" value="http://127.0.0.1:9003/callback"**/>  *<!-- 设置客户端名称(client\_name=CasClient)默认 -->* <**property name="name" value="CasClient"**/>  </**bean**>  *<!-- 配置cas服务端信息 -->* <**bean id="casConfiguration" class="org.pac4j.cas.config.CasConfiguration"**>  *<!-- CAS服务端登录请求URL -->* <**property name="loginUrl" value="http://sso.pinyougou.com/login/"**/>  *<!-- CAS服务端请求URL前缀-->* <**property name="prefixUrl" value="http://sso.pinyougou.com/"**/>  </**bean**>   *<!-- 配置安全管理器 -->* <**bean id="securityManager"  class="org.apache.shiro.web.mgt.DefaultWebSecurityManager"**>  *<!-- 配置自定义身份认证域-->* <**property name="realm" ref="pac4jRealm"**/>  </**bean**>  *<!-- 自定义身份认证域 -->* <**bean id="pac4jRealm" class="cn.itcast.casclient.realm.CasPac4jRealm"**/> </**beans**> |

casclient03\src\main\java\cn\itcast\casclient\realm\CasPac4jRealm.java，自定义身份认证域：

|  |
| --- |
| **public class** CasPac4jRealm **extends** Pac4jRealm {  *// 验证用户身份(Cas已认证)* @Override  **protected** AuthenticationInfo  doGetAuthenticationInfo(AuthenticationToken token) {  *// 调用Pac4jRealm实现的认证方法* **AuthenticationInfo authc = super.doGetAuthenticationInfo(token)**;  *// 获取登录用户名* **String username = ((Pac4jPrincipal) authc.getPrincipals()  .getPrimaryPrincipal()).getName();**  System.***out***.println(**"username = "** + username);  **return** authc;  }  *// 设置角色和权限* @Override  **protected** AuthorizationInfo  doGetAuthorizationInfo(PrincipalCollection principals) {  *// 获取登录用户名* String username = ((Pac4jPrincipal)principals  .getPrimaryPrincipal()).getName();  System.***out***.println(**"username = "** + username);  **return null**;  } } |

启动项目访问路径：http://localhost:9003/index.html。

## 7.3获取登录名

### 7.3.1集成SpringMVC

我们在处理后端逻辑需要获得登录名，那么如何获取单点登录的用户名呢? 其实和我们之前获得用户名的方式是完全相同的，我们下面来做个测试。

casclient03/src/main/webapp/WEB-INF/web.xml 添加springmvc配置：

|  |
| --- |
| *<!-- 配置前端控制器 -->* <**servlet**>  <**servlet-name**>casclient</**servlet-name**> <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  <**init-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:casclient-servlet.xml</**param-value**>  </**init-param**> </**servlet**> <**servlet-mapping**>  <**servlet-name**>casclient</**servlet-name**>  <**url-pattern**>/</**url-pattern**> </**servlet-mapping**> |

casclient03/src/main/resources/casclient-servlet.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd"**>  *<!-- 配置组件扫描 -->* <**context:component-scan**  **base-package="cn.itcast.casclient.controller"**/>  *<!-- 配置开启MVC注解驱动 -->* <**mvc:annotation-driven**/>  *<!-- 配置静态资源由默认的servlet处理 -->* <**mvc:default-servlet-handler**/> </**beans**> |

### 7.3.2控制器

casclient03\src\main\java\cn\itcast\casclient\controller\LoginController.java：

|  |
| --- |
| @RestController **public class** LoginController {  @GetMapping(**"/showName"**)  **public** String showName(){  Pac4jPrincipal pac4jPrincipal = (Pac4jPrincipal)SecurityUtils  .*getSubject*().getPrincipal();  **return** pac4jPrincipal.getName();  } } |

地址栏输入：http://localhost:9003/showName 看到登录用户名。

## 7.4退出登录

casclient03/src/main/resources/applicationContext-shiro.xml：

|  |
| --- |
| *<!-- 3. 退出过滤器，拦截需要退出的URL -->* <**entry key="logout"**>  <**bean class="io.buji.pac4j.filter.LogoutFilter"**>  <**property name="config" ref="config"**/>  *<!-- 中央认证退出 -->* <**property name="centralLogout" value="true"**/>  *<!-- 本地退出 -->* <**property name="localLogout" value="true"**/>  *<!-- 退出成功后默认重定向URL -->* <**property name="defaultUrl"  value="http://127.0.0.1:9003/success.html"**/>  </**bean**> </**entry**> |

在casclient03/src/main/webapp/index.html页面上添加链接：

|  |
| --- |
| <**a href="/logout"**>退出登录</**a**> |

casclient03/src/main/resources/applicationContext-shiro.xml中设置success.html为可匿名访问：

|  |
| --- |
| *<!-- 配置URL过滤器链(配置顺序为自上而下) -->* <**property name="filterChainDefinitions"**>  <**value**>  /success.html = anon  /callback = callback  /logout = logout  /\*\* = security  </**value**> </**property**> |

casclient03\src\main\webapp\WEB-INF\web.xml(配置在所有的filter之前)：

|  |
| --- |
| *<!-- 配置SingleSignOutFilter单点退出的过滤器 -->* <**filter**>  <**filter-name**>singleSignOutFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.session.SingleSignOutFilter</**filter-class**>  *<!-- 配置CAS服务端请求URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**> </**filter**> <**filter-mapping**>  <**filter-name**>singleSignOutFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**> </**filter-mapping**> |

**注意：必须在web.xml文件中配置单点退出过滤器，用于判断CAS服务端是否退出登录。**

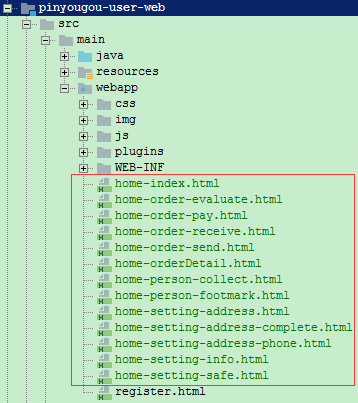
# 8.用户中心单点登录功能

## 8.1需求分析

用户中心实现单点登录。

## 8.2静态资源

拷贝【资料\用户中心原型】至 pinyougou-user-web/src/main/webapp/目录下，效果如下图所示：



## 8.3实现单点登录

### 8.3.1集成CAS

#### **8.3.1.1配置依赖**

在pinyougou-user-web/pom.xml文件中，配置引入shiro-spring、buji-pac4j、pac4j-cas依赖jar：

|  |
| --- |
| <**dependencies**>  *<!-- shiro-spring -->* <**dependency**>  <**groupId**>org.apache.shiro</**groupId**>  <**artifactId**>shiro-spring</**artifactId**>  </**dependency**>  *<!-- buji-pac4j -->* <**dependency**>  <**groupId**>io.buji</**groupId**>  <**artifactId**>buji-pac4j</**artifactId**>  <**version**>4.1.0</**version**>  </**dependency**>  *<!-- pac4j-cas -->* <**dependency**>  <**groupId**>org.pac4j</**groupId**>  <**artifactId**>pac4j-cas</**artifactId**>  <**version**>3.6.1</**version**>  </**dependency**> </**dependencies**> |

#### **8.3.1.2配置web.xml**

修改pinyougou-user-web/src/main/webapp/WEB-INF/web.xml文件：

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0"**>  *<!-- 配置单点退出过滤器 -->* <**filter**>  <**filter-name**>ssoutFilter</**filter-name**> <**filter-class**>org.jasig.cas.client.session.SingleSignOutFilter</**filter-class**>  *<!-- 配置CAS服务URL前缀 -->* <**init-param**>  <**param-name**>casServerUrlPrefix</**param-name**>  <**param-value**>http://sso.pinyougou.com</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>ssoutFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!-- 配置Spring核心监听器 -->* <**context-param**>  <**param-name**>contextConfigLocation</**param-name**> <**param-value**>classpath:applicationContext-\*.xml</**param-value**>  </**context-param**>  <**listener**> <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**>  </**listener**>   *<!-- 配置Spring委派过滤器代理(要求Spring容器中有了个bean的id为shiroFilter) -->* <**filter**>  <**filter-name**>shiroFilter</**filter-name**> <**filter-class**>org.springframework.web.filter.DelegatingFilterProxy</**filter-class**>  </**filter**>  <**filter-mapping**>  <**filter-name**>shiroFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>    *<!-- 配置Spring MVC前端控制器(核心控制器) -->* <**servlet**>  <**servlet-name**>pinyougou</**servlet-name**> <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  <**init-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:springmvc.xml</**param-value**>  </**init-param**>  <**load-on-startup**>1</**load-on-startup**>  </**servlet**>  <**servlet-mapping**>  <**servlet-name**>pinyougou</**servlet-name**>  <**url-pattern**>/</**url-pattern**>  </**servlet-mapping**>   *<!-- 配置全局编码过滤器 -->* <**filter**>  <**filter-name**>characterEncodingFilter</**filter-name**> <**filter-class**>org.springframework.web.filter.CharacterEncodingFilter</**filter-class**>  <**init-param**>  <**param-name**>encoding</**param-name**>  <**param-value**>UTF-8</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>characterEncodingFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!-- 欢迎列表 -->* <**welcome-file-list**>  <**welcome-file**>home-index.html</**welcome-file**>  </**welcome-file-list**> </**web-app**> |

#### **8.3.1.3配置pac4j-cas**

pinyougou-user-web\src\main\resources\applicationContext-shiro.xml，配置pac4j-cas：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd"**>   *<!-- 配置shiro过滤器工厂 -->* <**bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean"**>  *<!-- 配置注入安全管理对象 -->* <**property name="securityManager" ref="securityManager"**/>  *<!-- 配置过滤器 -->* <**property name="filters"**>  <**map**>  *<!-- 1. 安全过滤器，拦截需要登录的URL -->* <**entry key="security"**>  <**bean class="io.buji.pac4j.filter.SecurityFilter"**>  <**property name="config" ref="config"**/>  </**bean**>  </**entry**>  *<!-- 2. 回调过滤器，完成ticket验证 -->* <**entry key="callback"**>  <**bean class="io.buji.pac4j.filter.CallbackFilter"**>  <**property name="config" ref="config"**/>  *<!-- 验证通过后默认重定向URL -->* <**property name="defaultUrl" value="http://user.pinyougou.com"**/>  </**bean**>  </**entry**>  *<!-- 3. 退出过滤器，拦截需要退出的URL -->* <**entry key="logout"**>  <**bean class="io.buji.pac4j.filter.LogoutFilter"**>  <**property name="config" ref="config"**/>  *<!-- 中央退出 -->* <**property name="centralLogout" value="true"**/>  *<!-- 本地退出 -->* <**property name="localLogout" value="true"**/>  *<!-- 退出成功后默认重定向URL -->* <**property name="defaultUrl" value="http://www.pinyougou.com"**/>  </**bean**>  </**entry**>  </**map**>  </**property**>  *<!-- 配置URL过滤器链(配置顺序为自上而下) -->* <**property name="filterChainDefinitions"**>  <**value**>  /css/\*\* = anon  /img/\*\* = anon  /js/\*\* = anon  /plugins/\*\* = anon  /register.html = anon  /user/save = anon  /user/sendCode = anon   /callback = callback  /logout = logout  /\*\* = security  </**value**>  </**property**>  </**bean**>  *<!-- pac4j配置 -->* <**bean id="config" class="org.pac4j.core.config.Config"**>  <**constructor-arg name="client" ref="casClient"**/>  </**bean**>  *<!-- 配置CAS客户端 -->* <**bean id="casClient" class="org.pac4j.cas.client.CasClient"**>  *<!-- 设置cas服务端信息 -->* <**property name="configuration" ref="casConfiguration"**/>  *<!-- 登录成功后重定向回来的请求URL -->* <**property name="callbackUrl" value="http://user.pinyougou.com/callback"**/>  *<!-- 设置客户端名称(client\_name=CasClient)默认 -->* <**property name="name" value="CasClient"**/>  </**bean**>  *<!-- 配置cas服务端信息 -->* <**bean id="casConfiguration" class="org.pac4j.cas.config.CasConfiguration"**>  *<!-- CAS服务端登录请求URL -->* <**property name="loginUrl" value="http://sso.pinyougou.com/login/"**/>  *<!-- CAS服务端请求URL前缀-->* <**property name="prefixUrl" value="http://sso.pinyougou.com/"**/>  </**bean**>   *<!-- 配置安全管理器 -->* <**bean id="securityManager"  class="org.apache.shiro.web.mgt.DefaultWebSecurityManager"**>  *<!-- 配置自定义身份认证域-->* <**property name="realm" ref="pac4jRealm"**/>  </**bean**>  *<!-- 自定义身份认证域 -->* <**bean id="pac4jRealm" class="com.pinyougou.user.realm.CasPac4jRealm"**/> </**beans**> |

#### **8.3.1.4自定义认证域**

pinyougou-user-web\src\main\java\com\pinyougou\user\realm\CasPac4jRealm.java：

|  |
| --- |
| */\*\* 自定义身份认证域 \*/* **public class** CasPac4jRealm **extends** Pac4jRealm{  *// 验证用户身份(Cas已认证)* @Override  **protected** AuthenticationInfo  doGetAuthenticationInfo(AuthenticationToken token) {  *// 调用Pac4jRealm实现的认证方法* AuthenticationInfo authc = **super**.doGetAuthenticationInfo(token);  *// 获取登录用户名* String username = ((Pac4jPrincipal) authc.getPrincipals()  .getPrimaryPrincipal()).getName();  System.***out***.println(**"username = "** + username);  **return** authc;  }  *// 设置角色和权限* @Override  **protected** AuthorizationInfo  doGetAuthorizationInfo(PrincipalCollection principals) {  *// 获取登录用户名* String username = ((Pac4jPrincipal)principals  .getPrimaryPrincipal()).getName();  System.***out***.println(**"username = "** + username);  **return null**;  } } |

### 8.3.2获取登录用户名

#### **8.3.2.1前端代码**

修改pinyougou-user-web/src/main/webapp/home-index.html引入js：

|  |
| --- |
| <**script src="/js/vue.min.js"**></**script**> <**script src="/js/axios.min.js"**></**script**> <**script src="/js/controller/indexController.js"**></**script**> |

pinyougou-user-web/src/main/webapp/js/controller/目录下，创建indexController.js：

|  |
| --- |
| *// 窗口加载完* ***window***.onload = **function** () {  **var** vue = **new Vue**({  **el** : **'#app'**, *// 元素绑定* **data** : { *// 数据模型* **loginName** : **''** *// 登录用户名* },  **methods** : { *// 操作方法  // 获取登录用户名* showName : **function** () {  **axios**.get(**"/user/showName"**).then(**function**(response){  vue.**loginName** = response.**data**.**loginName**;  });  }  },  created : **function** () { *// 创建生命周期* **this**.showName();  }  }); }; |

home-index.html页面，隐藏登录与注册(25行)：

|  |
| --- |
| <**ul class="fl"**>  <**li class="f-item"**>{{**loginName**}}，品优购欢迎您！</**li**>  </**ul**> |

home-index.html页面，显示用户名(158行)：

|  |
| --- |
| <**span class="name"**>{{loginName}}</**span**> |

#### **8.3.2.2后端代码**

pinyougou-user-web/src/main/java/com/pinyougou/user/controller/包下创建LoginController.java：

|  |
| --- |
| */\*\* 登录控制器 \*/* @RestController **public class** LoginController {  @GetMapping(**"/user/showName"**)  **public** Map<String,String> showName(){  */\*\* 获取用户登录名 \*/* Pac4jPrincipal principal = (Pac4jPrincipal) SecurityUtils  .*getSubject*().getPrincipal();  String loginName = principal.getName();  Map<String, String> map = **new** HashMap<>();  map.put(**"loginName"**, loginName);  **return** map;  } } |

### 8.3.3退出登录

home-index.html页面退出登录后，跳转到网站首页(158行)：

|  |
| --- |
| <**span class="safe"**><**a href="/logout"**>退出登录 </**a**></**span**> |