

## tony的专栏



目录视图

摘要视图

RSS 订阅

## 个人资料



achuo

+ 加关注

发私信



访问: 108843次

积分: 1580

等级: BLOG &gt; 4

排名: 千里之外

原创: 8篇

转载: 293篇

译文: 0篇

评论: 5条

## 文章搜索



## 文章分类

java基础学习 (25)  
java之美 (3)  
数据库学习 (10)  
设计模式学习 (16)  
移动开发 (16)  
spring学习 (51)  
hibernate学习 (8)  
项目管理 (13)  
mybatis学习 (15)  
其他相关技术学习 (35)  
开源项目学习 (23)  
单元测试 (3)  
MongoDB (5)  
java面试 (11)  
maven (11)  
webservice (2)  
freemarker (2)  
Redis (7)  
hadoop (14)  
kafka学习 (2)  
apache shrio (6)  
RabbitMQ (5)

【活动】Python创意编程活动开始啦!!! CSDN日报20170423 ——《私活, 永远解救不了自己屌丝的人生!》 程序员4月书讯: Angular来了!

## 转 Jasig cas 单点登录系统Server&amp;Java Client配置

2016-11-22 07:49 581人阅读 评论(0) 收藏 举报

## 分类:

java基础学习 (24)

目录(?)

[+]

Jasig cas (Central Authentication Service) 官方网站: <http://www.jasig.org/cas>. 访问这个站点需要翻墙。cas官网文档访问地址: <https://wiki.jasig.org/display/CASUM/Home>。

我下载的是: 3.5.2版本。我配置好的cas界面如下:

界面一:

JASIG

## Central Authentication Service (CAS)

请输入您的用户名和密码。

用户名:

密 码:

☐ 七天内免登录

登录

重置

界面二:

jenkins (2)  
JPA (2)  
Aerospike (9)  
数据建模 (32)  
centos (1)  
Mahout (6)  
机器学习 (12)  
hive (23)  
网络爬虫 (3)

#### 文章存档

2017年03月 (5)  
2017年02月 (6)  
2017年01月 (10)  
2016年12月 (3)  
2016年11月 (2)

展开

#### 阅读排行

搭建jenkins实现自动化部署 (13583)  
手把手教你安装Hbase, 一 (7626)  
“多变量分析”——数据挖掘! (4453)  
StarUML时序图总结 (4372)  
JRebel最新破解版及使用 (3113)  
maven tomcat plugin实践 (2976)  
Spring batch 入门学习教程 (2249)  
基于Hadoop的数据仓库I (2007)  
揭秘17位华人巨富的第一 (1519)  
spring4.1+springmvc4.1 (1476)

#### 评论排行

spring mvc+hibernate 实 (1)  
centos安装配置hadoop (1)  
机器学习之——多项式回 (1)  
spring mvc整合shiro登录 (1)  
CentOS64位6.4下Hadoop (1)  
Spring学习4-面向切面 (0)  
基于AOP操作日志 (0)  
quartz在集群环境下的最 (0)  
500G JAVA视频网盘分享 (0)  
JAVA webservice之CXF (0)

#### 推荐文章

\* 探索通用可编程数据平面  
\* 这是一份很有诚意的 Protocol Buffer 语法详解  
\* CSDN日报20170420 ——《开发和产品之间的恩怨从何处来?》  
\* Android图片加载框架最全解析——从源码的角度理解Glide的执行流程  
\* 如果两个程序员差不多, 选写作能力更好的那个  
\* 从构造函数看线程安全

#### 最新评论

机器学习之——多项式回归和正则化  
sinat\_38203678: 非常感谢  
centos安装配置hadoop超详细过程  
Yaphet2009: 这个写的太好了!!!  
spring mvc整合shiro登录 权限验证  
风行傲天: 如果写一个注册页面, 怎么访问, 我写个注册页面, 没



## 下载链接:

已经搞好的Java客户端及服务器端(MySQL)配置下载: <http://pan.baidu.com/s/11MRPr>, 两个war包, 1个jar包:

cas.war, 是配置好的cas服务器端: <http://192.168.1.10:8080/casclient>。

casclient.war, 是cas客户端示例, 所谓cas客户端, 就是指使用cas进行单点登录的系统:

<http://192.168.1.10:8080/cas/login>。

cas-client-core-3.2.1-SNAPSHOT.jar: 是我在版本3.2.1源码基础上进行修改过的, 目的是登录成功后, 自动跳转到前一页。

要修改cas.war包中deployerConfigContext.xml里面的数据库连接串及密码查询SQL, jasig日志文件的位置:

tomcat所在磁盘:/data/applogs/jasig/目录下, 可以通过修改WEB-INF/class/log4j.xml进行配置。

需要的MySQL数据库privilege建表及数据初始化脚本: create\_privilege.sql

使用这个数据库, 默认登录帐户密码: root/root

最新代码Git地址:

cas-server-3.5.2: <https://github.com/pumadong/cas-server-3.5.2>

cas-client-3.2.1: <https://github.com/pumadong/cas-client-3.2.1>

## Jasig cas服务器端配置:

Jasig cas的功能是认证(单点登录, SSO, Single Sign On), 关于系统的授权, 由于各个系统可能都不太一样, 各个系统自行负责为宜。

Jasig cas的配置非常简单, 从官网下载的源码包中, 有个module目录, 里面有制作好的war包, 可以直接使用, 当然也可以自行使用maven打包, 命令如下:

```
mvn clean package -Dmaven.test.skip=true -U
```

把war包改名为cas.war, 放到tomcat的webapps目录下, 就可以使用了, 默认使用的是登陆验证类是测试用的, 只要用户名、密码一直就算登陆成功, 为了在生产环境使用, 我们改为使用MySQL数据库验证, 步骤如下:

### 1、简单起见, 我们不使用https认证, 做如下更改:

WEB-INF\spring-configuration目录下ticketGrantingTicketCookieGenerator.xml和warnCookieGenerator.xml, 把p:cookieSecure="true"改为p:cookieSecure="false"

WEB-INF目录下deployerConfigContext.xml中, 在如下节点中, 增加p:requireSecure="false",

```
<bean
```

```
class="org.jasig.cas.authentication.handler.support.HttpBasedServiceCredentialsAuthenticationHandler" p:httpClient="httpClient" p:requireSecure="false" />
```

### 2、使用MySQL验证

WEB-INF\deployerConfigContext.xml文件中,

有登录注册页面访问不了，这个如何配置

CentOS64位6.4下Hadoop2.7.1、qq\_36482667: HAO

spring mvc+hibernate 实现事务！qq\_30355613: 看了你这篇文章对于我这个新手来说还是有点迷糊。可以加个好友吗？我的QQ604755500



对<property name="authenticationHandlers">节点，做如下更改：

```
<property name="authenticationHandlers">
  <list>
    <bean
      class="org.jasig.cas.authentication.handler.support.HttpBasedServiceCredentialsAuthenticationHandler"
      p:httpClient-ref="httpClient" p:requireSecure="false" />
    <bean class="org.jasig.cas.adaptors.jdbc.QueryDatabaseAuthenticationHandler">
      <property name="dataSource" ref="dataSource" />
      <property name="sql" value="select password from p_user where username=?" />
      <property name="passwordEncoder" ref="passwordEncoderBean"/>
    </bean>
  </list>
</property>
```

根节点，也就是beans节点中，增加如下节点：

```
<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <property name="driverClassName">
    <value>com.mysql.jdbc.Driver</value>
  </property>
  <property name="url">
    <value>jdbc:mysql://localhost:3306/test?useUnicode=true&characterEncoding=utf8</value>
  </property>
  <property name="username"><value>root</value></property>
  <property name="password"><value>root</value></property>
</bean>
<bean id="passwordEncoderBean" class="org.jasig.cas.authentication.handler.DefaultPasswordEncoder">
  <constructor-arg value="MD5" />
</bean>
```

注意：

同时，需要把cas-server-support-jdbc-3.5.2.jar包放在cas.wab包的WEB-INF目录下，把mysql-connector-java-5.1.24-bin.jar放入Tomcat的WEB-INF/lib目录里。

### 3、退出时跳转到service页

修改src/main/webapp/WEB-INF/cas-servlet.xml里的logoutController

增加p:followServiceRedirects="true"使支持logout输入service参数为跳转路径。

### 4、增加免登陆（Remember Me）功能

CAS增加免登陆（Remember Me）功能：

<https://wiki.jasig.org/display/CASUM/Remember+Me>

[http://blog.163.com/wunan\\_23/blog/static/19556232020111127714418/](http://blog.163.com/wunan_23/blog/static/19556232020111127714418/)

## Jasig cas客户端配置

### 1、pom.xml配置

```
<!-- CAS Client -->
<dependency>
  <groupId>org.jasig.cas</groupId>
  <artifactId>cas-client-core</artifactId>
  <version>3.1.10</version>
</dependency>
```

如果不使用maven的话，直接下载jar包即可。

### 2、web.xml配置

```
<!-- ===== 单点登录开始 ===== -->
<!-- 说明：这种客户端的配置方式是不需要Spring支持的 -->
<!-- 参考资料：http://blog.csdn.NET/yaoweijq/article/details/6003187 -->
```

```

<listener>
    <listener-class>org.jasig.cas.client.session.SingleSignOutHttpSessionListener</listener-class>
</listener>

<filter>
<filter-name>CAS Single Sign Out Filter</filter-name>
    <filter-class>org.jasig.cas.client.session.SingleSignOutFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>CAS Single Sign Out Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

<filter>
    <filter-name>CAS Authentication Filter</filter-name>
    <filter-class>org.jasig.cas.client.authentication.AuthenticationFilter</filter-class>
    <init-param>
        <param-name>casServerLoginUrl</param-name>
        <param-value>http://localhost:8080/cas/login</param-value>
    </init-param>
    <init-param>
        <param-name>serverName</param-name>
        <param-value>http://localhost:8080</param-value>
    </init-param>
</filter>

<filter-mapping>
    <filter-name>CAS Authentication Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

<filter>
    <filter-name>CAS Validation Filter</filter-name>
    <filter-class>org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</filter-class>
    <init-param>
        <param-name>casServerUrlPrefix</param-name>
        <param-value>http://localhost:8080/cas</param-value>
    </init-param>
    <init-param>
        <param-name>serverName</param-name>
        <param-value>http://localhost:8080</param-value>
    </init-param>
</filter>

<filter-mapping>
    <filter-name>CAS Validation Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

<!-- 该过滤器使得开发者可以通过org.jasig.cas.client.util.AssertionHolder来获取用户的登录名。比如
AssertionHolder.getAssertion().getPrincipal().getName()。 -->
    <filter>
<filter-name>CAS Assertion Thread Local Filter</filter-name>
<filter-class>org.jasig.cas.client.util.AssertionThreadLocalFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>CAS Assertion Thread Local Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

<!-- ===== 单点登录结束 ===== -->

```

### 3、单点退出

单点退出的原理，可以总结为一句话：

A客户端退出，调用cas服务器的logout，cas server会向所有相关的接入CAS的系统发送消息，这些接入CAS的系统监听到消息后，触发session失效。

注意：我们的接入CAS的系统必须是cas服务器可以访问到的，如果我们使用：<http://localhost:8080/picture>的形式访问站点，则是无法实现单点登出的，因为cas服务器访问不到我们的localhost，必须改成：<http://ip地址:8080/picture>的形式，让CAS可以访问到我们的系统。

参考：

<http://www.blogjava.net/xmatthew/archive/2008/07/09/213808.html>

<http://blog.csdn.net/yuwenrui/article/details/6607669>

## 更改单点登录界面

WEB-INF\view\jsp\default\ui

登陆界面：casLoginView.jsp

登陆成功：casGenericSuccess.jsp

登出界面：casLogoutView.jsp

cas是支持国际化的，关于页面中显示的提示信息的存储位置class/\*.properties，默认是en，为了让比较简单

的中文化，我把en做了个备份，然后把zh\_CN拷贝成en。  
在properties文件中，对于非英文的字符，采用16进制存储，关于unicode字符和16进制字符的转换，提供如下两个函数：

```
[java] view plain copy print ? C P
01. //把中文字符转换为十六进制Unicode编码字符串
02. public static String stringToUnicode(String s) {
03.     String str = "";
04.     for (int i = 0; i < s.length(); i++) {
05.         int ch = (int) s.charAt(i);
06.         if (ch > 255)
07.             str += "\\u" + Integer.toHexString(ch);
08.         else
09.             str += "\\" + Integer.toHexString(ch);
10.     }
11.     return str;
12. }
13.
14. //把十六进制Unicode编码字符串转换为中文字符串
15. public static String unicodeToString(String str) {
16.     Pattern pattern = Pattern.compile("(\\\\\\\\u(\\\\p{XDigit}{4}))");
17.     Matcher matcher = pattern.matcher(str);
18.     char ch;
19.     while (matcher.find()) {
20.         ch = (char) Integer.parseInt(matcher.group(2), 16);
21.         str = str.replace(matcher.group(1), ch + "");
22.     }
23.     return str;
24. }
```

也可以更改cas-servlet.xml文件中关于本地化的配置，把：

<bean id="localeResolver" class="org.springframework.web.servlet.i18n.CookieLocaleResolver"/> 修改为：

<bean id="localeResolver" class="org.springframework.web.servlet.i18n.FixedLocaleResolver"/>

这个本地化解析器返回一个固定的本地化信息。默认值为当前JVM的locale。

另外，我在修改一个CAS界面时，运行过程中每次刷新页面都报一个警告：WARN:oejh.HttpGenerator:Ignoring extra content，并且点击页面的submit按钮，大于在第3或者第5次的时候，页面就reload了，后来用排除法（逐渐缩小代码范围）找出问题，我的新界面的login\_soft.css中，有关于图片的css: background: url(../img/bg-white-lock.png) repeat; 但是对应的目录下没有相应图片，导致了如上的两个问题。把图片都拷贝到相应目录后，问题解决。为什么会这样，原因尚不明，估计与CAS的WebFlow相关代码有关，不深入继续研究了😁

## 关于log4j.xml

logger的默认additivity值是true，代表日志继续抛出到root进行处理，此时会被root里面所有appender处理；

logger设置additivity的值为false，代表日志被此节点的appender吃掉，不继续抛出；

root里面的<level value="ERROR" />只对空的logger节点才生效，比如：

```
<logger name="com.github.inspektr.audit.support.Slf4jLoggingAuditTrailManager" additivity="true" />
```

## 关于源码

服务器端源码(版本：**cas-server-3.5.2**)，实际上我们没有必要对服务器端源码做更改，需要的配置直接修改xml就可以了：

inspektrThrottledSubmissionContext.xml中：应该是spring-beans-3.1.xsd，不能是spring-beans.xsd，否则ref="local"会报错：

cvc-complex-type.3.2.2: Attribute 'local' is not allowed to appear in element 'ref'.

cas-server-webapp项目里面，测试下面的很多xml都是格式不正确（其实可以直接在Eclipse里面把Validate关掉），报如下错误：

The markup in the document following the root element must be well-formed.

avax.servlet.jsp.PageContext cannot be resolved to a type, 错误，参考：

<http://stackoverflow.com/questions/8669387/javax-servlet-jsp-pagecontext-cannot-be-resolved-to-a-type>

以上错误不影响源码编译。

注意，因为我们计划用MySQL进行验证，需要引入JDBC，所以在cas-server-webapp项目的pom.xml中，增加依赖，如下：

```
<dependency>
  <groupId>org.jasig.cas</groupId>
  <artifactId>cas-server-support-jdbc</artifactId>
  <version>${project.version}</version>
</dependency>
```

客户端源码（版本：**cas-client-3.2.1**），修改其中的cas-client-core：

```
[java] view plain copy print ?
01. //修改类: org.jasig.cas.client.util.AbstractCasFilter的constructServiceUrl, 修改后内容如下:
02. //return CommonUtils.constructServiceUrl(request, response, this.service, this.serverName, th
03. String currentPage = request.getScheme() + "://" + request.getServerName() + ":" + request.ge
04.     + request.getRequestURI();
05.     + (CommonUtils.isNotBlank(request.getQueryString()) ? ("?" + request.getQueryString()) : "");
06. String ticket = request.getParameter("ticket");
07. //http://192.168.1.200:8080/cas/serviceValidate?ticket=ST-74-bJlsAKJivCfkABMHlgUiu-
    cas01.example.org&service=http%3A%2F%2Flocalhost%3A8080%2Fauthority%2Fcontroller%2Fmain.do%3F
    bJlsAKJivCfkABMHlgUiu-cas01.example.org
08. //去掉参数中所有ticket相关的部分
09. currentPage = currentPage.replace("%3fticket%3D"+ticket, "");
10. currentPage = currentPage.replace("&ticket="+ticket, "");
11. currentPage = currentPage.replace("?ticket="+ticket, "");
12. currentPage = currentPage.replace("&service=", "?service=");
13. return currentPage;
14. //或者修改类: org.jasig.cas.client.util.CommonUtils的constructServiceUrl, 修改后被替换后的内容如下:
15. if (CommonUtils.isBlank(serverName)) {
16.     buffer.append(request.getRequestURI());
17. }else{
18.     if (!serverName.startsWith("https://") && !serverName.startsWith("http://")) {
19.         buffer.append(request.isSecure() ? "https://" : "http://");
20.     }else{
21.         buffer.append(serverName);
22.     }
23.     buffer.append(request.getRequestURI());
24. }
```

目的是，让客户端登陆后，自动跳转到认证前的页面，这样，我们就不需要配置cas.service.url(认证成功的返回页面)参数了，也更合理。



# 工作原理

如果session有效，则认证通过；

如果session无效，一般是别的系统单点登录，本系统第一次访问，则ticket认证，并session生成；

如果单点退出，调用cas server的logout，则cas server对所有相关单点发送session失效消息，单点监听此消息并触发session过期；

可以看到，绝大多数认证都是通过session进行的，所以对于后台系统来说，不会有任何性能问题。

参考文章：

<http://www.docin.com/p-370942269.html>

[http://blog.sina.com.cn/s/blog\\_56d8ea900100144r.html](http://blog.sina.com.cn/s/blog_56d8ea900100144r.html)

<http://www.doc88.com/p-183734738161.html>

<http://wenku.baidu.com/view/5c64b1f19e31433239689318.html>

<http://www.blogjava.net/tufanshu/archive/2011/01/21/343290.html>

<http://blog.csdn.net/rishengcsdn/article/details/10379321> (Oracle)



顶  
0

踩  
0

上一篇 Linux常用命令大全

下一篇 Win7 64位 mysql 5.7下载安装问题

## 我的同类文章

### java基础学习（24）

- 500G JAVA视频网盘分享 2017-03-28 阅读 72
- CAS4.0 SERVER登录后用... 2017-02-08 阅读 98
- 【SSO单点系列】（2）：C. 2017-02-08 阅读 38
- JRebel最新破解版及使用方. 2017-01-15 阅读 3034
- Lombok 安装、入门 - 消除... 2016-02-24 阅读 154
- Java注解学习二：package.. 2016-01-30 阅读 923
- Java注解学习：注解术语以... 2016-01-30 阅读 443
- 2015CSDN下载热门书籍 2016-01-21 阅读 124
- Java中WebService实例 2016-01-20 阅读 91
- Java WebService 简单实例 2016-01-20 阅读 290

更多文章



## 科技园北租房 | 1630 元/月

三天不满意全额退款

## 猜你在找

- 基于Servlet+JDBC+Bootstrap+MySQL+AJAX权限管理 SSO单点登录CAS 配置验证码
- 使用JDBC操作MySQL数据库
- MySQL数据库入门精讲视频教程
- mysql 遇上jdbc
- JDBC入门教程（备java基础，oracle，mysql，javaee）
- 单点登录CAS系列8-客户端配置单点登录
- cas 单点登录SSO实验之二 cas-client
- Spring Security定制开发 单点登录CAS定制开发 集成
- 单点登录三cas服务端配置



4 **¥275.00/只**  
YYD对讲机厂家通讯设备10W大功率Y1803通

去看看



5 **¥26.00/个**  
江苏电信32芯插片式光分路器箱48芯壁挂

去看看



6 **¥180.00/件**  
订做精图网络箱 ONU箱 综合箱 网络配线箱

去看看

查看评论

暂无评论

您还没有登录,请[登录](#)或[注册](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题   Hadoop   AWS   移动游戏   Java   Android   iOS   Swift   智能硬件   Docker   OpenStack  
VPN   Spark   ERP   IE10   Edipse   CRM   JavaScript   数据库   Ubuntu   NFC   WAP   jQuery  
BI   HTML5   Spring   Apache   .NET   API   HTML   SDK   IIS   Fedora   XML   LBS   Unity  
Splashtop   UML   components   Windows Mobile   Rails   QEMU   KDE   Cassandra   CloudStack  
FTC   coremail   OPhone   CouchBase   云计算   iOS6   Rackspace   Web App   SpringSide  
Maemo   Compuware   大数据   aptech   Perl   Tomado   Ruby   Hibernate   ThinkPHP   HBase  
Pure   Solr   Angular   Cloud Foundry   Redis   Scala   Django   Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

 网站客服  杂志客服  微博客服  [webmaster@csdn.net](mailto:webmaster@csdn.net)  400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved 