**SO ： 单点登录（Single Sign On , 简称 SSO ）**

**CAS ： CAS(Central Authentication Service)** 耶鲁大学的开源项目

a comprehensive platform for your authentication and authorization needs

an open and well-documented authentication protocol

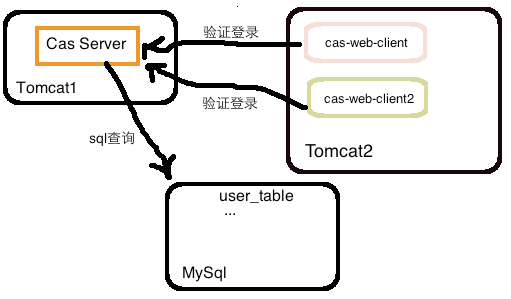
[WAR Overlay method](http://apereo.github.io/cas/development/installation/Maven-Overlay-Installation.html):

**doc:**

**http://www.open-open.com/lib/view/open1392714410256.html**

**开源**

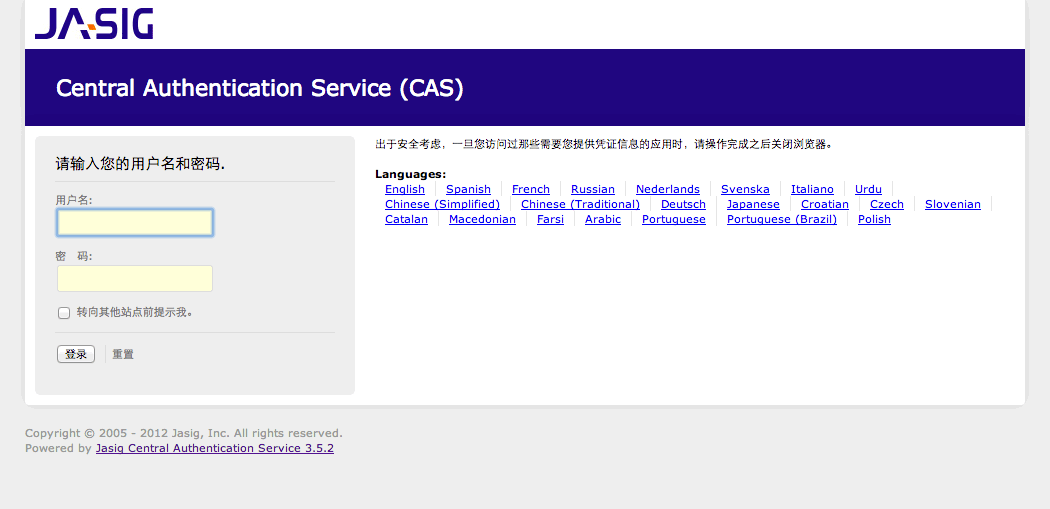
https://github.com/apereo/cas



已经打包的server：

jasig CAS实现单点登录(数据库认证)

改名为cas.war, 复制到${TOMCAT\_HOME}/webapps/下，启动tomcat，这时进入https://www.tomcat1.com:8443/cas/, 见下图，则服务端已经配置ok：



这时输入用户名及密码(保持一样就可以，默认CAS验证方式只要用户名密码一样就行)

* 配置web客户端：

新建WEB项目cas-web-client, 并加入依赖包：cas-client-core-3.2.1.jar，commons-logging-1.1.jar

<?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\_2\_5.xsd" id="WebApp\_ID" version="2.5">

    <display-name>cas-web-client2</display-name>

    <listener>

        <listener-class>

            org.jasig.cas.client.session.SingleSignOutHttpSessionListener

        </listener-class>

    </listener>

*<!-- CAS 登出监听器/过滤器 -->*

    <filter>

        <filter-name>CasSingleSignOutFilter</filter-name>

        <filter-class>org.jasig.cas.client.session.SingleSignOutFilter</filter-class>

    </filter>

    <filter-mapping>

        <filter-name>CasSingleSignOutFilter</filter-name>

        <url-pattern>/\*</url-pattern>

    </filter-mapping>

*<!-- CAS 认证过滤器 -->*

    <filter>

       <filter-name>CASFilter</filter-name>

       <filter-class>org.jasig.cas.client.authentication.AuthenticationFilter</filter-class>

        <init-param>

            <param-name>casServerLoginUrl</param-name><!-- Cas Server登录url-->

            <param-value>https://www.tomcat1.com:8443/cas/login</param-value>

        </init-param>

        <init-param><!-- 配置当前web应用所在的web服务器域名url -->

            <param-name>serverName</param-name>

            <param-value>http://www.tomcat2.com:8080</param-value>

        </init-param>

    </filter>

    <filter-mapping>

        <filter-name>CASFilter</filter-name>

        <url-pattern>/\*</url-pattern>

    </filter-mapping>

*<!-- CAS 票据验证过滤器 -->*

    <filter>

        <filter-name>CasTicketFilter</filter-name>

        <filter-class>

            org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</filter-class>

        <init-param>

            <param-name>casServerUrlPrefix</param-name>

            <param-value>https://www.tomcat1.com:8443/cas</param-value>

        </init-param>

        <init-param>

            <param-name>serverName</param-name>

            <param-value>http://www.tomcat2.com:8080</param-value>

        </init-param>

    </filter>

    <filter-mapping>

        <filter-name>CasTicketFilter</filter-name>

        <url-pattern>/\*</url-pattern>

    </filter-mapping>

*<!-- CAS 外覆过滤器，用于获取认证用户信息 -->*

    <filter>

        <filter-name>CasRequestWrapFilter</filter-name>

        <filter-class>

            org.jasig.cas.client.util.HttpServletRequestWrapperFilter</filter-class>

    </filter>

    <filter-mapping>

        <filter-name>CasRequestWrapFilter</filter-name>

        <url-pattern>/\*</url-pattern>

    </filter-mapping>

*<!-- CAS 外覆过滤器，用于获取认证用户信息 -->*

    <filter>

        <filter-name>AssertionThreadLocalFilter</filter-name>

        <filter-class>org.jasig.cas.client.util.AssertionThreadLocalFilter</filter-class>

    </filter>

    <filter-mapping>

        <filter-name>AssertionThreadLocalFilter</filter-name>

        <url-pattern>/\*</url-pattern>

    </filter-mapping>

    <welcome-file-list>

        <welcome-file>index.jsp</welcome-file>

    </welcome-file-list>

</web-app>

server自定义验证：

由于基于数据库验证，还得将cas-server-support-jdbc-3.5.2.jar和mysql-connector驱动包拷贝到WEB-INF/lib目录下

做的是修改一些配置，就是WEB-INF/deployerConfigContext.xml文件，先配置mysql数据源：

<!-- DataSource，根据你的环境来定 -->

<bean id="mysqlDataSource" class="org.apache.commons.dbcp.BasicDataSource">

<property name="driverClassName" value="com.mysql.jdbc.Driver" />

<property name="url" value="jdbc:mysql://192.168.141.129:3306/blog?useUnicode=true&amp;characterEncoding=utf-8"/>

<property name="username" value="mysqladmin" />

<property name="password" value="mysqladmin" />

</bean>

修改 authenticationManager bean的属性 authenticationHandlers：

<property name="authenticationHandlers">

    <list>

        <bean class="org.jasig.cas.authentication.handler.support.HttpBasedServiceCredentialsAuthenticationHandler" p:httpClient-ref="httpClient" />

        <!--注释掉默认的的认证实现<bean

        class="org.jasig.cas.authentication.handler.support.SimpleTestUsernamePasswordAuthenticationHandler" />-->

        <!-- 数据库查询认证处理器 -->

        <bean class="org.jasig.cas.adaptors.jdbc.QueryDatabaseAuthenticationHandler">

            <property name="dataSource" ref="mysqlDataSource"/>

           <property name="sql"

                value="select password from t\_user where username = ?" />

            <property name="passwordEncoder" ref="myPasswordEncoder"></property>

</bean>

    </list>

</property>

<!-- 自己定义的密码转译器，你可以自己定义，也可以用CAS提供的 -->

<bean id="myPasswordEncoder" class="org.jasig.cas.custom.encoder.MyPasswordEncoder" />

这样就配置完成了，将项目导出war包，并部署到tomcat1所在机器上运行起来，效果会之前一样。