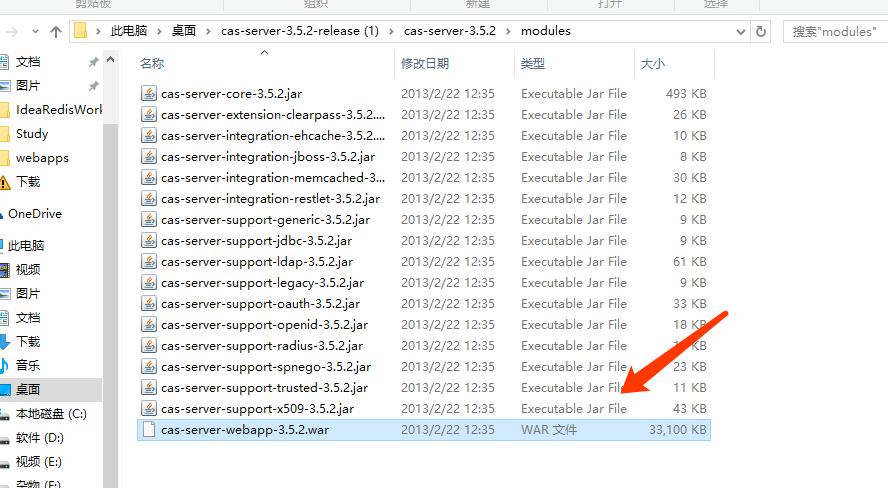
单点登录服务端

# 1、tomcat 版本一定要和jdk版本一致，否则根本不能够显示 （JDK1.8 Tomcat 8.0）

## 1、下载服务器war包



# 2、部署到tocmat webapp下面 ，启动tomcat healerjean:28080/cas/login (admin/admin登录)

## 1、配置tomcat端口

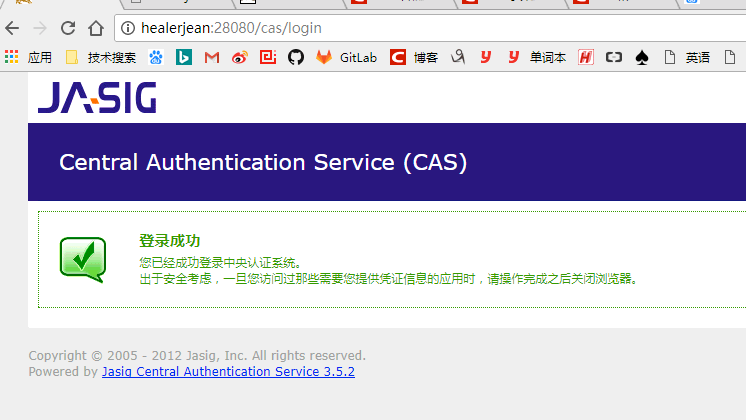
<Server port="28005" shutdown="SHUTDOWN">

<Connector port="28080" protocol="HTTP/1.1"

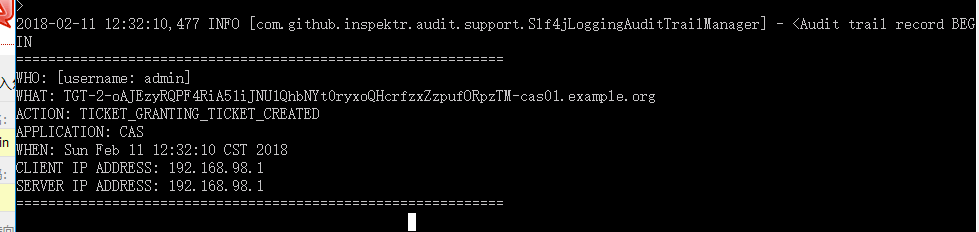
connectionTimeout="20000"

redirectPort="28443" />

<Connector port="28009" protocol="AJP/1.3" redirectPort="28443" />



## 3、tomcat 后台



、

# 3、客户端配置

## 1、配置客户端tomcat端口

<Server port="18005" shutdown="SHUTDOWN">

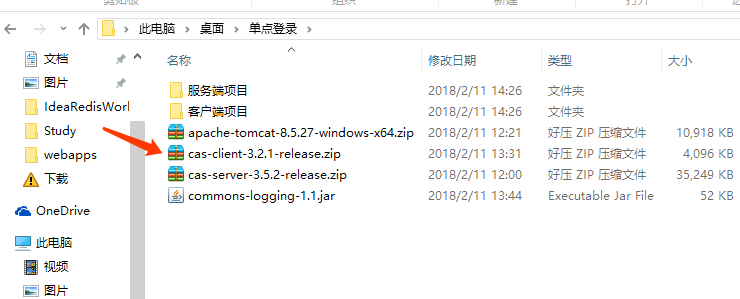
<Connector port="18080" protocol="HTTP/1.1"

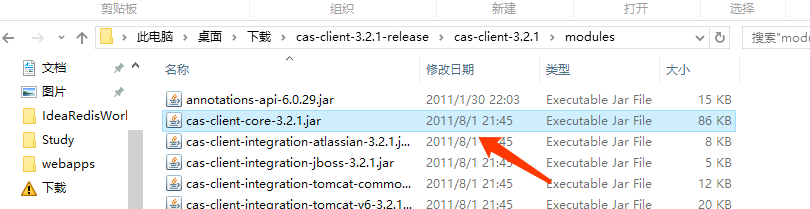
connectionTimeout="20000"

redirectPort="18443" />

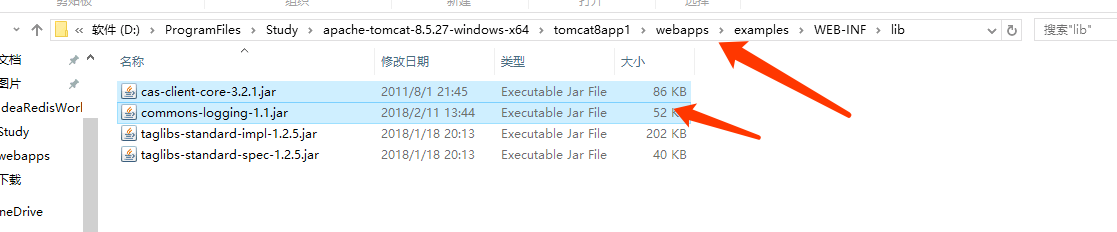
<Connector port="18009" protocol="AJP/1.3" redirectPort="18443" />

## 2、下载客户端代码， 找到里面客户端核心cas-client-core-3.2.1.jar包





## 3、，tomcat中原有的example项目保留，将上面的jar包和从网上下载的common.logging.1.1.jar包放入 examples 项目的lib中



# 4、配置客户端代码进行单点登录，打开example 中的web.xml

**<!-- 用于单点退出，该过滤器用于实现单点登出功能，可选配置-->**

**<listener>**

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

**</listener>**

**<!-- 该过滤器用于实现单点登出功能，可选配置。 -->**

**<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>**

**<!-- 该过滤器负责用户的认证工作，必须启用它 -->**

**<filter>**

**<filter-name>CAS Filter</filter-name>**

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

**<init-param>**

**<param-name>casServerLoginUrl</param-name>**

**<param-value>http://healerjean:28080/cas/login</param-value>**

**<!--这里的server是服务端的IP -->**

**</init-param>**

**<init-param>**

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

**<param-value>http://healerjeancasapp1:18080</param-value>**

**</init-param>**

**</filter>**

**<filter-mapping>**

**<filter-name>CAS Filter</filter-name>**

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

**</filter-mapping>**

**<!-- 该过滤器负责对****Ticket的校验工作，必须启用它 -->**

**<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://healerjean:28080/cas</param-value>**

**</init-param>**

**<init-param>**

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

**<param-value>http://healerjeancasapp1:18080</param-value>**

**</init-param>**

**</filter>**

**<filter-mapping>**

**<filter-name>CAS Validation Filter</filter-name>**

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

**</filter-mapping>**

**<!-- 该过滤器负责实现HttpServletRequest请求的包裹， 比如允许开发者通过HttpServletRequest的**getRemoteUser**()方法获得SSO登录用户的登录名，可选配置。 -->**

**<filter>**

**<filter-name>CAS HttpServletRequest Wrapper Filter</filter-name>**

**<filter-class>**

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

**</filter>**

**<filter-mapping>**

**<filter-name>CAS HttpServletRequest Wrapper 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>**

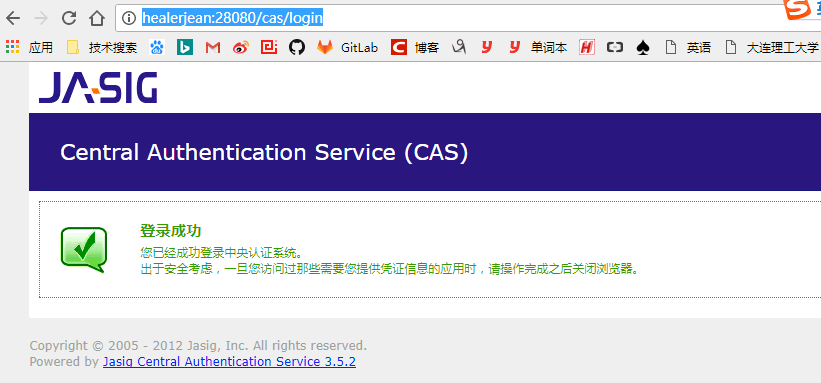
**</filter-mapping>**

**<!-- ======================== 单点登录结束 ======================== -->**

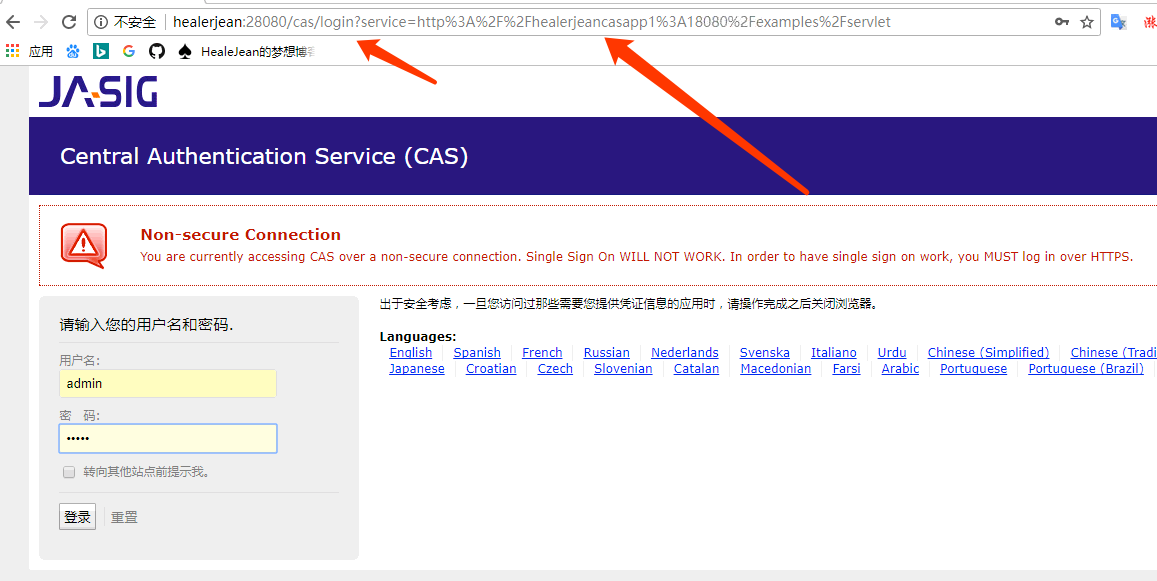
**</web-app>**

# 5、启动服务器和客户端的代码

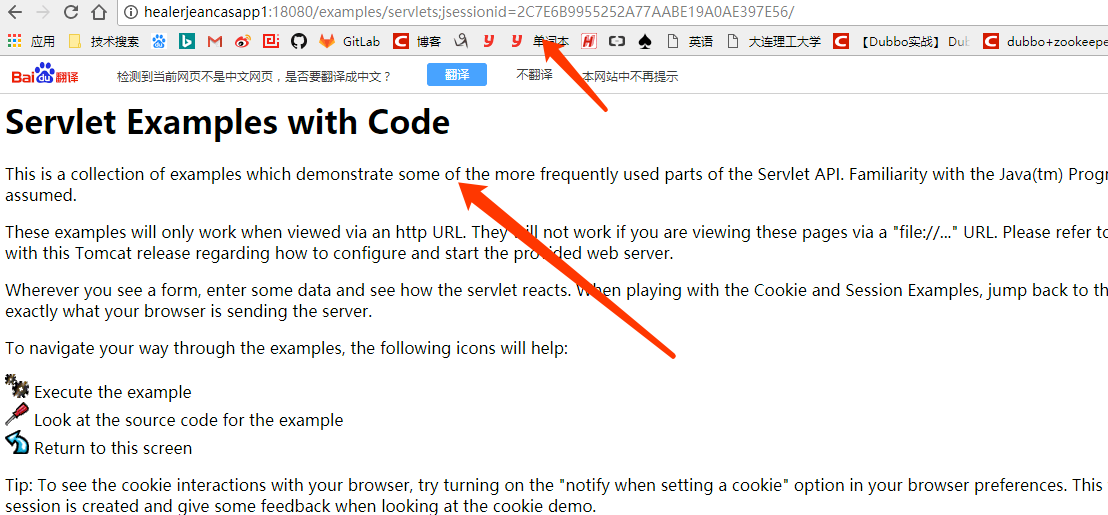
## 1、检测服务器是否可用 http://healerjean:28080/cas/login



## 2、客户端访问 http://healerjeancasapp1:18080/examples/servlets，目标地址直接跳转到了

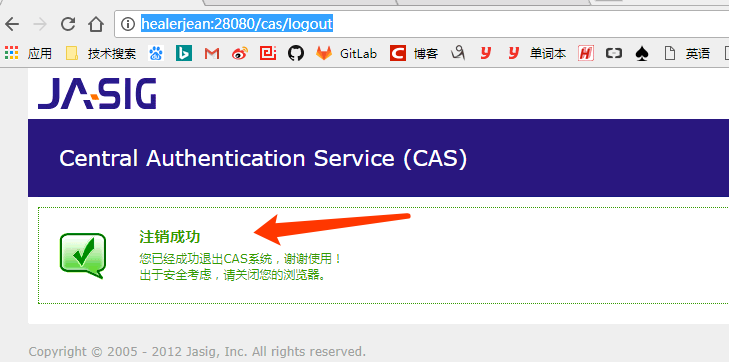


## 3、输入密码登录后就会跳转到它自然的页面



# 6、登出

## http://healerjean:28080/cas/logout



# 7、客户端获取登录信息

public class HelloWorldExample extends HttpServlet {  
  
 private static final long ***serialVersionUID*** = -6593274907821061823L;  
  
 @SuppressWarnings("unchecked")  
 public void doGet(HttpServletRequest request, HttpServletResponse response){  
  
 //断言  
 Assertion assertion = (Assertion) request.getSession().getAttribute(AbstractCasFilter.***CONST\_CAS\_ASSERTION***);  
  
 AttributePrincipal principal = assertion.getPrincipal();  
 String username = principal.getName(); //admin，账户  
  
 }  
}

　　　　AttributePrincipal principal = (AttributePrincipal) request.getUserPrincipal();  
  
　　　　final Map attributes = principal.getAttributes();

/\*\*  
 \* 类描述：  
 \* 创建人： j.sh  
 \* 创建时间： 2016/4/13  
 \* version：1.0.0  
 \*/  
public class AppSessionHelper {  
  
 public static final String ***SESSION\_USER\_ID*** = "\_session\_user\_id\_";  
 public static final String ***SESSION\_USER*** = "\_session\_user\_";  
 public static final String ***SESSION\_ACCOUNT*** = "\_session\_account\_";  
 public static final String ***SESSION\_HEADIMG*** = "\_session\_headimg\_";  
  
  
 public static String getRemoteUserAccount(){  
 return AssertionHolder.getAssertion().getPrincipal().getName();  
 }  
  
 public static Long getRemoteUserId(){  
 return Long.parseLong(AssertionHolder.getAssertion().getPrincipal().getAttributes().get("id").toString());  
 }  
  
 public static String getRemoteUserName(){ //管理员设置的姓名之类的（个人理解）  
 return AssertionHolder.getAssertion().getPrincipal().getAttributes().get("name").toString();  
 }  
  
}

## 下面二者都是登录的用户名

request.getRemoteUser()  
AssertionHolder.getAssertion().getPrincipal().getName())