第一步： SSL准备(默认给出的认证服务是没有开启https的,具体参照服务端README.md,再执行这一步)

1、生成数字证书

keytool -genkey -keystore "E:\localhost.keystore" -alias localhost -keyalg RSA

2、使用 localhost.keystore 导出数字证书（公钥）到 D:\localhost.cer client要用到

keytool -export -alias localhost -file E:\localhost.cer -keystore E:\localhost.keystore

3、将数字证书（公钥）导入到JDK中

① cd E:\Program Files\Java\jdk1.7.0\_51\jre\lib\security

② keytool -import -alias localhost(对应keystore的用户名) -file E:\localhost.cer -noprompt -trustcacerts -storetype jks -keystore cacerts -storepass localhost

--------如果失败，请删除cacerts重试。

2、设tomcat下的server.xml

<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"

maxThreads="150" scheme="https" secure="true"

clientAuth="false" sslProtocol="TLS"

keystoreFile="E:\localhost.keystore" keystorePass="localhost"/>

第二步： 部署认证服务器

配置Tomcat:根据SSL修改server.xml 添加JVM参数 建议:JAVA\_OPTS="-server -Xms700m -Xmx700m -XX:PermSize=20m -XX:MaxPermSize=40m -XX:+PrintGCDateStamps -XX:+PrintGCDetails -Xloggc:${catalina.base}/logs/gc.log"

根据提供的casserver.war,根目录下README.md修改相关配置后,启动即可.

第三步: 部署客户端服务器(shiro)

1. 添加依赖

<dependency>

<groupId>org.jasig.cas.client</groupId>

<artifactId>cas-client-core</artifactId>

<version>3.3.2</version>

</dependency>

<dependency>

<groupId>org.apache.shiro</groupId>

<artifactId>shiro-cas</artifactId>

<version>1.2.4</version>

</dependency>

1. 修改sso目录下properties相关配置,用spring-shiro-cas.xml替换原来的shiro配置文件.这里要注意要自己实现ShiroCasRealm用于查询该用户的权限.

casFilter对应的路劲,要与sso.properties中的service的后缀一致.

1. 修改web.xml

<!-- SSO 单点登出配置 请放在最前面-->

<listener>

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

</listener>

<filter>

<filter-name>singleSignOutFilter</filter-name>

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

<init-param>

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

<param-value>http://localhost:8080/cas-server</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>singleSignOutFilter</filter-name>

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

</filter-mapping>

<!-- SSO 单点登出配置 -->

1. 完成，启动.

第三步: 部署客户端服务器(spring+web.xml)

1. 添加依赖(略)
2. 修改sso.properties相关配置.将sso.xml添加到applicationContext.
3. 修改web.xml

<!-- cas filter配置,对应sso.xml -->

<listener>

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

</listener>

<filter>

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

<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>

<init-param>

<param-name>targetBeanName</param-name>

<param-value>singleSignOutFilter</param-value>

</init-param>

</filter>

<filter-mapping>

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

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

</filter-mapping>

<filter>

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

<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>

<init-param>

<param-name>targetBeanName</param-name>

<param-value>authenticationFilter</param-value>

</init-param>

</filter>

<filter-mapping>

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

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

</filter-mapping>

<filter>

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

<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>

<init-param>

<param-name>targetBeanName</param-name>

<param-value>ticketValidationFilter</param-value>

</init-param>

</filter>

<filter-mapping>

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

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

</filter-mapping>

<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>

<filter>

<filter-name>CAS Assertion Thread Local Filter</filter-name>

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

</filter>

<filter-mapping>

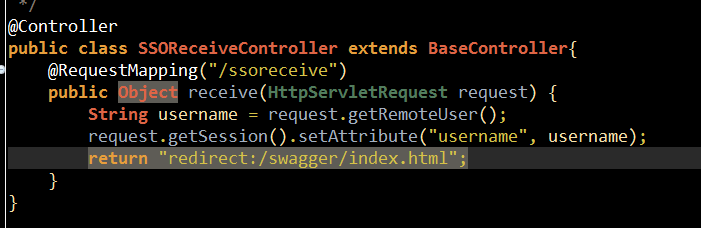
<filter-name>CAS Assertion Thread Local Filter</filter-name>

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

</filter-mapping>

<!-- sso filter结束 -->

1. 新增Controller，路劲和sso.properties当中的casService的后缀一致.负责登陆成功+ticket验证成功后接收返回username的地方,也可以视为整个client真正的入口,拿到username后记录下来,重定向到client的index页面



5、完成启动。