title: SSO单点登录之CAS-Server
categories: Web开发
date: 2016-12-27 18:15:15
tags:
- SSO
- CAS

- Portal

Prerequisites

- `CAS Server 3.5.x.`，下文以`$CAS-SERVER`表示。
- `CAS Client 3.3.x.`，下文以`$CAS-CLIENT`表示。
- `Tomcat 7.x`，下文以`$CAS_TOMCAT_HOME`表示。
- `GateIn-3.8.1.Final-tomcat-7`，下文以`$GATEIN_HOME`表示。

CAS介绍

`CAS`（Central Authentication Service）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。单点登录，既在多个应用系统中，用户只需登录一次就可以访问所有相互信任的应用系统。`CAS Client`支持非常多的客户端，包括`JAVA、PHP、Ruby`等。

接下来，接介绍一下`CAS`系统。

CAS 系统组成

`CAS`系统架构由两部分组成，`CAS Server`、`CAS Clients`，两者可以通过多种协议进行通信。

CAS Server

`CAS Server`是一个构建在`Spring Framework`上的`Java servlet`，通过分配和诊断`tickets`，负责认证用户以及授权。

CAS Clients

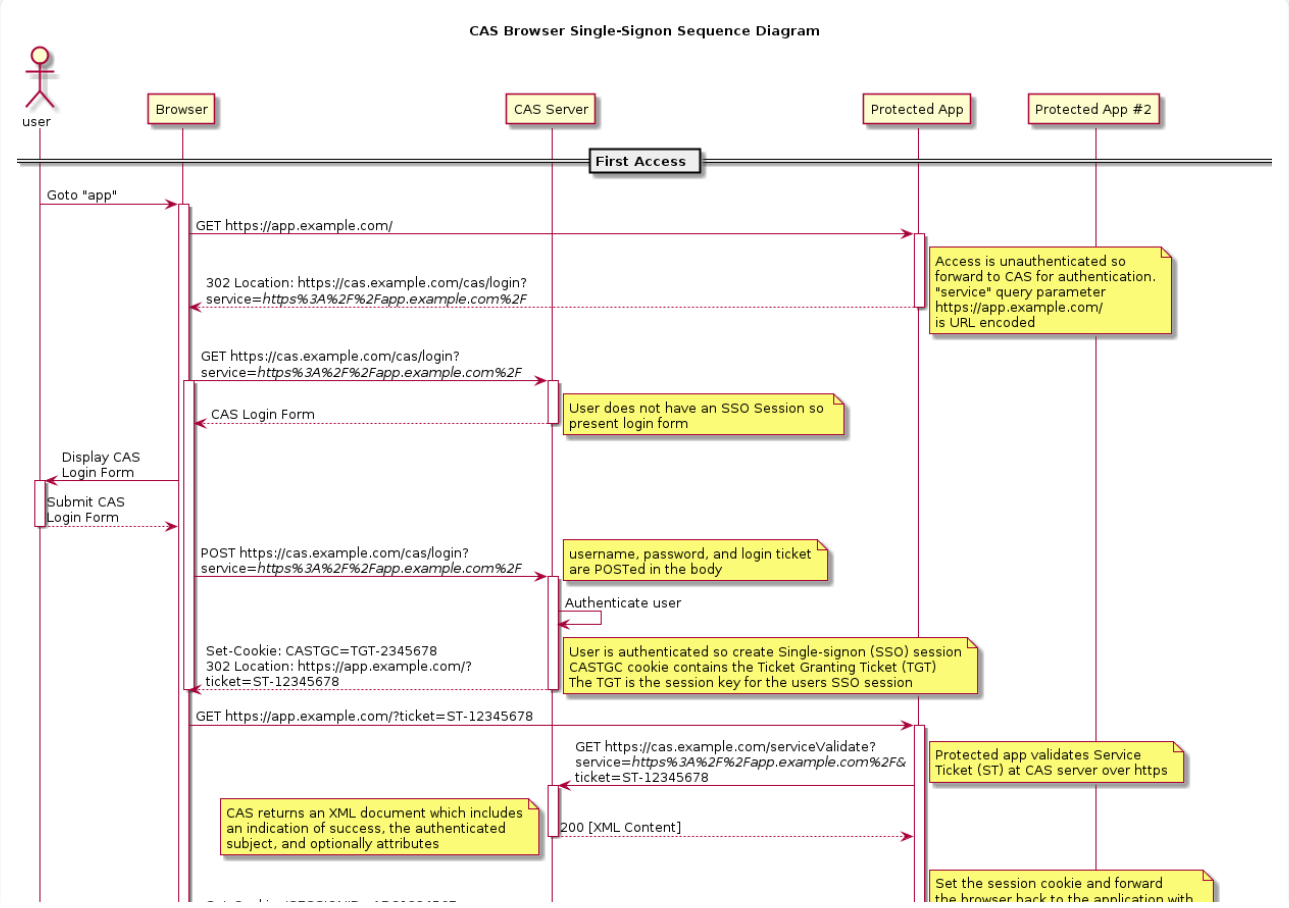
`CAS Clients`负责对用户的认证工作，`CAS Clients`负责处理对客户端受保护资源的访问请求，需要登录时，重定向到`CAS Server`。

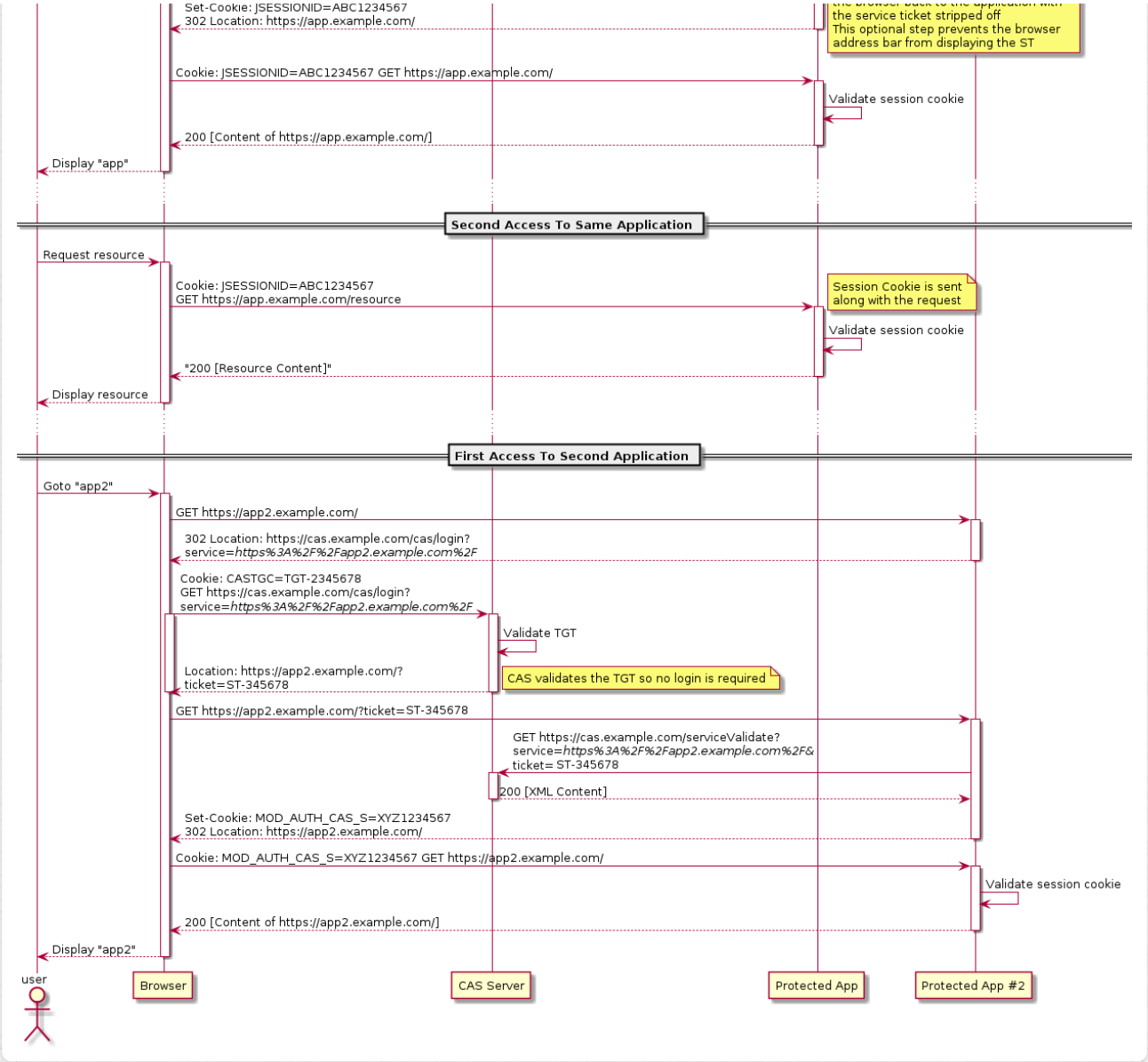
`CAS Clients`与受保护的客户端应用部署在一起，以`Filter`方式保护受保护的资源。

CAS协议流程

CAS协议流程如下图所示，展示了用户同时访问多个应用系统的流程。多个应用系统能实现单点登录的基本流程为，当用户首次访问需要登录才能访问的页面时，会自动重定向到`CAS Server`的登陆页面，成功认证完后，会在`CAS Server`的域中设置`CASTGC`的`Cookie`，最终登陆完成后，服务器会创建一个`session`会话，用于之后与应用系统的交互。当用户同时再登陆另外一个应用系统时，同样会跟之前一样，重定向到`CAS Server`的登陆页面，区别是此时已经有了`CAS Server`域中的`CASTGC`，重定向时会附带该`Cookie`，`CAS Server`验证之后返回一个`ticket`，之后浏览器重新请求原访问页面，并附带`ticket`参数，`CAS Server`诊断有效后，返回原应用系统的重定向，且设置该域的`session Cookie`，浏览器最后请求原页面，并附带`session Cookei`，应用系统诊断后返回请求内容。

需要注意的是，当成功登陆完某个系统后，如果继续再访问该系统的其他资源页面，是不需要再次与`CAS Server`进行交互的，应用系统将根据`session`直接进行诊断。





GateIn Portal集成CAS Server

部署CAS

1. 打开 `$CAS_TOMCAT_HOME/webapps/$CAS-SERVER/WEB-INF/deployerConfigContext.xml`, 替换:

```
<bean
class="org.jasig.cas.authentication.handler.support.SimpleTestUsernamePass
/>
```

为如下:

```
<bean class="org.gatein.sso.cas.plugin.AuthenticationPlugin">
<property name="gateInProtocol"><value>http</value></property>
<property name="gateInHost"><value>localhost</value></property>
<property name="gateInPort"><value>8080</value></property>
<property name="gateInContext"><value>portal</value></property>
<property name="httpMethod"><value>POST</value></property>
</bean>
```

如上所示，用来配置`GateIn Portal`的服务地址。

2. 下载`GateIn SSO package`，下载地址，解压后，将其`cas/plugin/WEB-INF/lib`下的`jar`包拷贝到`$CAS_TOMCAT_HOME/webapps/$CAS-SERVER/WEB-INF/lib`目录。

3. 默认，登出用户时`CAS Server`会展示一个`CAS`提供的登出页面，然后跳转回`Portal`页，如果想要保留原有`Portal`的登出，打开`$CAS_TOMCAT_HOME/webapps/$CAS-SERVER/WEB-INF/cas-servlet.xml`，添加`followServiceRedirects="true"`参数：

```
<bean id="logoutController" class="org.jasig.cas.web.LogoutController"
p:centralAuthenticationService-ref="centralAuthenticationService"
p:logoutView="casLogoutView"
p:warnCookieGenerator-ref="warnCookieGenerator"
p:ticketGrantingTicketCookieGenerator-
ref="ticketGrantingTicketCookieGenerator"
p:followServiceRedirects="true"/>
```

部署GateIn Portal

---

1. 为了能让`Portal`使用`CAS Server`提供的单点登陆系统，首先配置`Portal`的`SSO`参数，在`$GATEIN_HOME/gatein/conf/configuration.properties`文件中，修改和添加如下内容：

```
gatein.sso.enabled=true
gatein.sso.callback.enabled=${gatein.sso.enabled}
gatein.sso.login.module.enabled=${gatein.sso.enabled}
gatein.sso.login.module.class=org.gatein.sso.agent.login.SSOLoginModule
gatein.sso.server.url=http://localhost:8086/cas-server
gatein.sso.portal.url=http://localhost:8080
gatein.sso.filter.logout.class=org.gatein.sso.agent.filter.CASLogoutFilter
gatein.sso.filter.logout.url=${gatein.sso.server.url}/logout
gatein.sso.filter.login.sso.url=${gatein.sso.server.url}/login?
service=${gatein.sso.portal.url}/@@portal.container.name@@/initiatessologi
```

如上，为配置`CAS Server`的服务器信息等。

2. 如果需要改变账户系统的存储方式， 比如改为`MySQL`数据库，还需要在`$GATEIN_HOME/gatein/conf/configuration.properties`文件中修改成如下所示，

同时，下载`mysql-connect-java.jar` `jar`包，放入`$GATEIN_HOME/lib`目录下。

```
gatein.idm.datasource.name=jdbcidm
gatein.idm.datasource.driver=com.mysql.jdbc.Driver
gatein.idm.datasource.url=jdbc:mysql://localhost:3306/jdbcidm_${name}
gatein.idm.datasource.username=root
gatein.idm.datasource.password=123
```

在`$GATEIN_HOME/conf/server.xml`的`<GlobalNamingResources></GlobalNamingResources>`节点中声明绑定的数据源，添加如下，字段含义可参考该文，注意，数据库名必须为`jdbcidm_portal`，且需要提前手动创建，无法自动创建，但是`GateIn Portal`会自动创建用户相关表：

```
<Resource auth="Container" driverClassName="com.mysql.jdbc.Driver" lo
gAbandoned="true" maxActive="20" maxIdle="10" maxWait="10000" minEvic
tableIdleTimeMillis="60000" name="exo-idm_portal" password="123" remo
veAbandoned="true" removeAbandonedTimeout="10" type="javax.sql.DataSo
urce" url="jdbc:mysql://localhost:3306/jdbcidm_portal" username="root
"/>
```

在`GateIn Portal`与`CAS`整合后，账户系统将由`GateIn Portal`接管，也就是说，如果`GateIn Portal`服务没有开启，则`CAS Server`将无法进行认证。

3. 在`$GATEIN_HOME/conf/server.xml`的`Host`元素下添加`ServletAccessValve`，如：

```
<Host name="localhost" appBase="webapps"
  unpackWARs="true" autoDeploy="true">

  <Valve className="org.gatein.sso.agent.tomcat.ServletAccessValve" /
>

  <!-- SingleSignOn valve, share authentication between web applicati
ons
  ...
```

其目的是开启`SSO`组件，将其加入`Catalina`容器的请求处理管道中，这样，`SSO`组件将有机会处理每一个`Request`请求。

参考

- https://www.ibm.com/developerworks/cn/opensource/os-cn-cas/
- https://www.exoplatform.com/docs/public/index.jsp?topic=%2FPLF35%2FADM.Configuration.Connect_To_A_Production_Database.html
- https://repository.jboss.org/nexus/content/groups/public/org/gatein/sso/sso-packaging/1.4.1.Final/sso-packaging-1.4.1.Final.zip
- http://www.cnblogs.com/vhua/p/cas_1.html