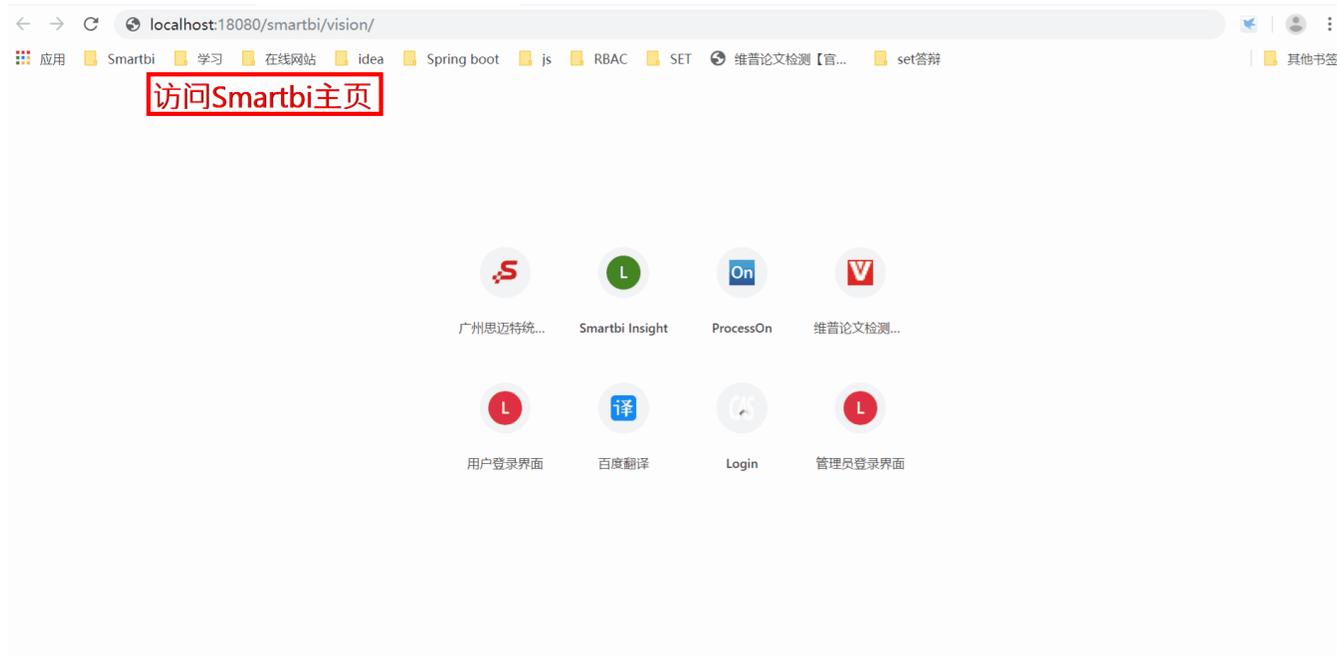


CAS单点登录

- 1. 说明
- 2. 具体操作步骤

1. 说明

在很多的應用項目中，客戶會採用 JA-SIG CAS(Central Authentication Service)作為統一登錄認證平台，要求 Smartbi 也要接入到 CAS 平台中，效果如下。



2. 具体操作步骤

下面將介紹如何在Smartbi中配置CAS，實現單點登錄。請參考如下操作步驟。

1、首先下載插件包（CAS.ext）、（CAS_tomcat10.ext tomcat10版本擴展包）；

2、用 WinRAR 之類的壓縮工具，打開 CAS.ext，將CAS.ext\META-INF\extension.xml、CAS.ext\vision\js\ext\ConfigurationPatch.js 兩個文件從 WinRAR 窗口中拖出來，對其進行修改。接着將修改後的文件，拖放回 WinRAR 窗口中，覆蓋以前的舊文件。關閉 WinRAR 窗口保存修改即可，如下圖所示。





3、打开extension.xml 文件，将相关地址改为统一平台的地址，smartbi的地址改为实际smartbi的地址，具体如下（如下涉及的地址仅需要修改IP和端口，保持与示例地址格式一致）：

(1) 修改 casServerLogoutUrl 参数，将 http://domain:18080/cas/logout 修改为CAS统一平台的注销地址。

```

4 <!-- 用于单点退出，该过滤器用于实现单点退出功能，CAS通知应用单点退出-->
5 <context-param>
6   <param-name>casServerLogoutUrl</param-name>
7   <param-value>
8     http://domain:18080/cas/logout
9   </param-value>
10 </context-param>
11 <listener>
12   <listener-class>org.jasig.cas.client.session.SingleSignOutHttpSessionListener</listener-class>
13 </listener>

```

(2) 修改 casServerLoginUrl 参数，将 http://domain:18080/cas/login 修改为CAS统一平台的登录地址。

```

42 <!-- 该过滤器负责用户的认证工作，必须启用它 -->
43 <filter>
44   <filter-name>CASFilter</filter-name>
45   <filter-class>org.jasig.cas.client.authentication.AuthenticationFilter</filter-class>
46   <init-param>
47     <param-name>casServerLoginUrl</param-name>
48     <param-value>http://domain:18080/cas/login</param-value>
49   </init-param>
50
51   <init-param>
52     <param-name>serverName</param-name>
53     <param-value>http://localhost:8080</param-value>
54   </init-param>
55 </filter>
56 <filter-mapping>
57   <filter-name>CASFilter</filter-name>
58   <url-pattern>/*</url-pattern>
59 </filter-mapping>
60

```

(3) 修改 `serverName` 参数，将其配置为 smartbi 系统的访问地址，比如：`http://localhost:18080`；注意，有两处需要配置 `serverName` 参数。

```
42 <!-- 该过滤器负责用户的认证工作，必须启用它 -->
43 <filter>
44   <filter-name>CASFilter</filter-name>
45   <filter-class>org.jasig.cas.client.authentication.AuthenticationFilter</filter-class>
46   <init-param>
47     <param-name>casServerLoginUrl</param-name>
48     <param-value>http://domain:18080/cas/login</param-value>
49   </init-param>
50
51   <init-param>
52     <param-name>serverName</param-name>
53     <param-value>http://localhost:8080</param-value>
54   </init-param>
55 </filter>
56 <filter-mapping>
57   <filter-name>CASFilter</filter-name>
58   <url-pattern>/*</url-pattern>
59 </filter-mapping>
60
```

```
61 <!-- 该过滤器负责对Ticket的校验工作，必须启用它 -->
62 <filter>
63   <filter-name>CAS Validation Filter</filter-name>
64   <filter-class>org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</filter-class>
65   <init-param>
66     <param-name>casServerUrlPrefix</param-name>
67     <param-value>http://domain:18080/cas</param-value>
68   </init-param>
69   <init-param>
70     <param-name>serverName</param-name>
71     <param-value>http://localhost:8080</param-value>
72   </init-param>
73   <init-param>
74     <param-name>useSession</param-name>
75     <param-value>true</param-value>
76   </init-param>
77   <init-param>
78     <param-name>redirectAfterValidation</param-name>
79     <param-value>true</param-value>
80   </init-param>
81 </filter>
```

(4) 修改 `casServerUrlPrefix` 参数，将 `http://domain:18080/cas` 修改为CAS统一平台的访问地址。

```
61 <!-- 该过滤器负责对Ticket的校验工作，必须启用它 -->
62 <filter>
63   <filter-name>CAS Validation Filter</filter-name>
64   <filter-class>org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</filter-class>
65   <init-param>
66     <param-name>casServerUrlPrefix</param-name>
67     <param-value>http://domain:18080/cas</param-value>
68   </init-param>
69   <init-param>
70     <param-name>serverName</param-name>
71     <param-value>http://localhost:8080</param-value>
72   </init-param>
```

4、打开 `ConfigurationPatch.js` 文件，将其中 `CASLogoutURL` 属性值改为CAS统一平台的注销地址，如下图红框内代码所示。

```
var ConfigurationPatch = {
  extensionPoints: {
    CASLogoutURL: "http://domain:18080/cas/logout"
  }
};
```

```
var ConfigurationPatch = {
  extensionPoints: {
    CASLogoutURL: "http://domain:18080/cas/logout"
  }
};
```

5、接下来部署修改后的 CAS.ext 插件包。完整的部署过程和操作步骤，请参考 [扩展包部署](#)。

6、启动Smartbi 系统后进入<http://localhost:18080/smartbi/vision/config.jsp> 配置界面，此时因配置了CAS.ext插件包，会跳转到CAS服务器登录界面，登录成功后会跳转到Smartbi的登录界面，此时已经完成了与CAS服务器的对接，但是还没有实现免密登录，需要修改“用户管理”配置段中的“加密类型”为CLASS，“登录验证类”为 smartbi.cas.auth.CASAuthUser，具体如下图所示。

加密类型：	CLASS
登录验证类：	smartbi.cas.auth.CASAuthUser

7、配置完成后，点击右下角的“保存”按钮，根据提示再重启Smartbi 后，即可看到效果。



注意：

- 本例所用 CAS 的jar包为cas-client-core-3.2.0.jar。
- 配置 CAS 登录后，就不能再用 Smartbi 系统内置的 admin 帐户了，如果希望继续使用，只能在 CAS 系统中创建一个 admin 帐户。
- 此CAS插件包只支持PC浏览器端。
- 针对标准的CAS，配置完CAS后，在Office插件/Excel插件登录系统需进行“登录配置插件”才可以成功登录，详情参考[电子表格实现CAS登录验证](#)。但根据以往支持的项目经验，项目客户通常使用的是自行改造后的CAS，届时也需要联系官方支持，对扩展包进行相应的改造开发。
- 仅V10.5版本的CAS单点登录支持移动端。
- CAS目前通过openresource.jsp打开报表，默认不走单点登录，具体使用时，可以根据实际情况配置。