1. enterprise-app/sso/

1.1. SSO之CAS单点登录实例演示

发表日期: 2012 年 5 月 16 日

本文目录：

- 一、概述
- 二、演示环境
- 三、JDK安装配置
- 四、安全证书配置
- 五、部署CAS-Server相关的Tomcat
- 六、部署CAS-Client相关的Tomcat
- 七、 测试验证SSO

一、概述

此文的目的就是为了帮助初步接触SSO和CAS 的人员提供一个入门指南，一步一步演示如何实现基于CAS的单点登录。

CAS的官网：http://www.jasig.org/cas

二、演示环境

本文演示过程在同一个机器上的（也可以在三台实体机器或者三个的虚拟机上），环境如下：

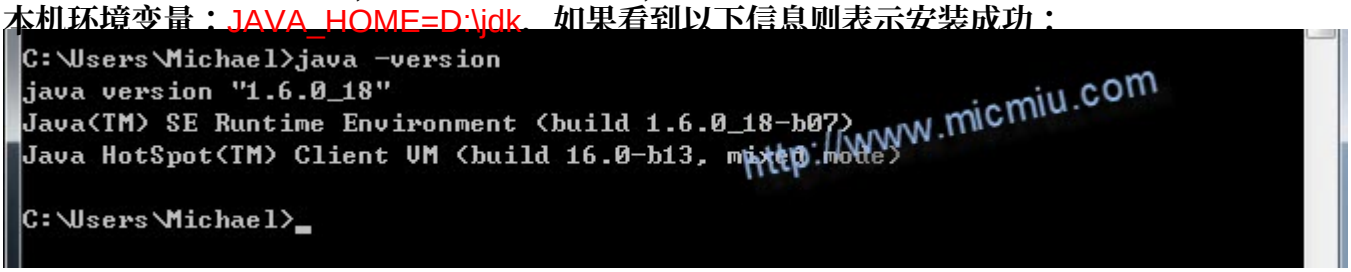
- windows7 64位，主机名称：michael-pc
- JDK 1.6.0_18
- Tomcat 6.0.29
- CAS-server-3.4.11、CAS-client-3.2.1

```
127.0.0.1    demo.micmiu.com
127.0.0.1    app1.micmiu.com
127.0.0.1    app2.micmiu.com
```

三、JDK安装配置

这个详细过程就不在描述，如果是免安装版的，确保环境变量配置正确。

本机环境变量：JAVA_HOME=D:\jdk，如果看到以下信息则表示安装成功：



四、安全证书配置

有关keytool工具的详细运用见：http://www.micmiu.com/lang/java/keytool-start-guide/

4.1. 生成证书：

```
keytool -genkey -alias ssodemo -keyalg RSA -keysize 1024 -keypass michaelpwd -validity 365 -keystore g:\sso\ssodemo.keystore -storepass michaelpwd
```

```
G:\>keytool -genkey -alias ssodemo -keyalg RSA -keysize 1024 -keypass michaelpwd
 -validity 365 -keystore g:\sso\ssodemo.keystore -storepass michaelpwd
您的名字与姓氏是什么？
  [Unknown]:  demo.micmiu.com
您的组织单位名称是什么？
  [Unknown]:  micmiu.com
您的组织名称是什么？
  [Unknown]:  micmiu
您所在的城市或区域名称是什么？
  [Unknown]:  SH
您所在的州或省份名称是什么？
  [Unknown]:  SH
该单位的两字母国家代码是什么
  [Unknown]:  CN
CN=demo.micmiu.com, OU=micmiu.com, O=micmiu, L=SH, ST=SH, C=CN 正确吗？
  [否]:  y
```

要和访问的域名一致（也可主机名）

ps：

- **截图中需要输入的姓名和上面hosts文件中配置的一致；**
- **keypass 和 storepass 两个密码要一致，否则下面tomcat 配置https 访问失败；**

4.2.导出证书：

```
keytool -export -alias ssodemo -keystore g:\sso\ssodemo.keystore -file g:\sso\ssodemo.crt -
storepass michaelpwd
```

```
G:\>keytool -export -alias ssodemo -keystore g:\sso\ssodemo.keystore -file g:\ss
o\ssodemo.crt
输入keystore密码:
保存在文件中的认证 (g:\sso\ssodemo.crt)
```

该处输入：michaelpwd

4.3.客户端导入证书：

```
keytool -import -keystore %JAVA_HOME%\jre\lib\security\cacerts -file g:\sso\ssodemo.crt -alias
ssodemo
```

```
G:\>keytool -import -keystore %JAVA_HOME%\jre\lib\security\cacerts -file g:\sso\
ssodemo.crt -alias ssodemo
输入keystore密码:
所有者:CN=demo.micmiu.com, OU=micmiu.com, O=micmiu, L=SH, ST=SH, C=CN
签发人:CN=demo.micmiu.com, OU=micmiu.com, O=micmiu, L=SH, ST=SH, C=CN
序列号:4fb0763c
有效期: Mon May 14 11:04:28 CST 2012 至Tue May 14 11:04:28 CST 2013
证书指纹:
         MD5:DC:C6:1C:C2:DE:6A:2F:FD:DC:93:BE:02:23:B2:12:43
         SHA1:93:46:AC:6B:62:4A:C8:E6:CB:16:B5:D8:09:99:50:77:BC:0D:04:28
         签名算法名称:SHA1withRSA
         版本: 3
信任这个认证？ [否]:  y
认证已添加至keystore中
```

输入：changeit 不是证书的密码：michaelpwd

ps：**该命令中输入的密码和上面输入的不是同一个密码；如果是多台机器演示，需要在每一台客户端导入该证书。**

五、部署CAS-Server相关的Tomcat

5.1. 配置HTTPS

解压apache-tomcat-6.0.29.tar.gz并重命名后的路径为 G:\sso\tomcat-cas，在文件 conf/server.xml文件找到：

```
<!--
    <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
               maxThreads="150" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS" />
    -->
```

修改成如下：

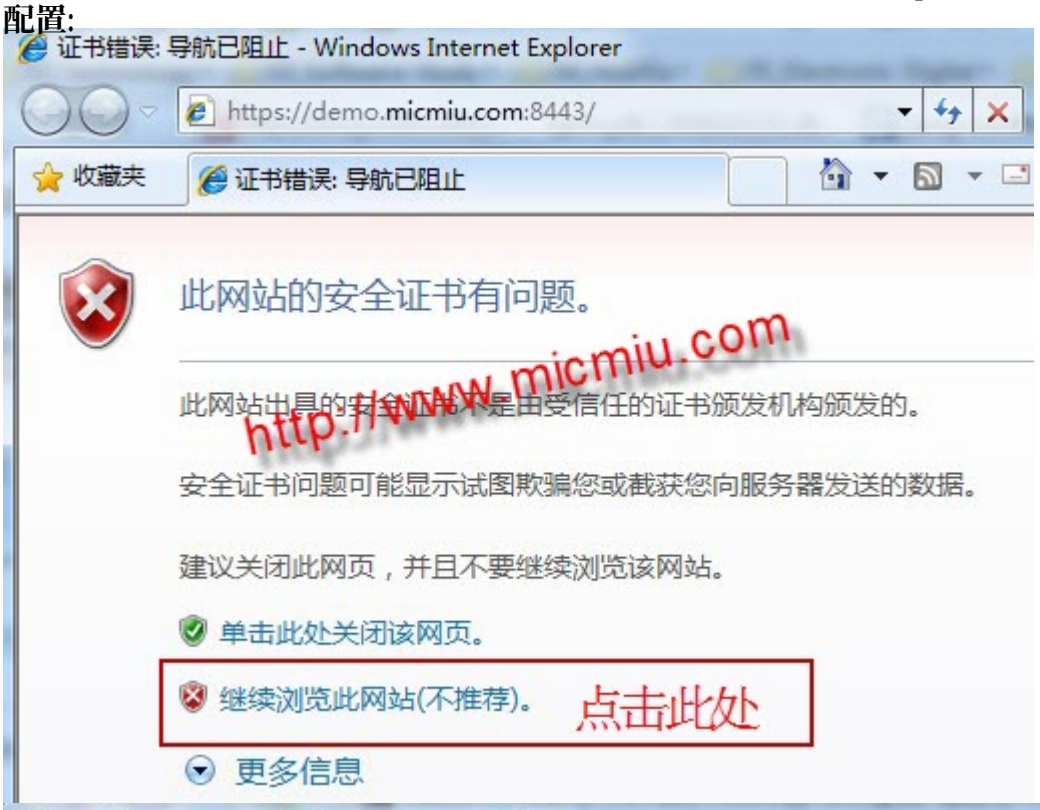
```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
               maxThreads="150" scheme="https" secure="true"
               keystoreFile="g:/sso/ssodemo.keystore" keystorePass="michaelpwd"
               clientAuth="false" sslProtocol="TLS" URIEncoding="UTF-8"
```

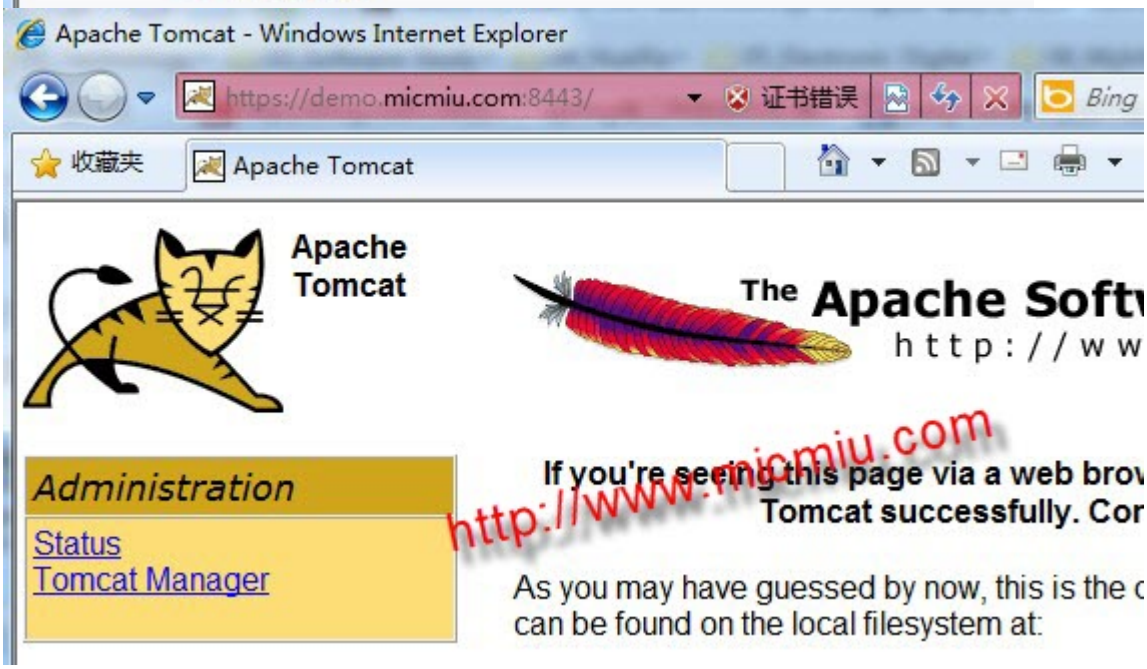
参数说明：

- keystoreFile 就是4.1中创建证书的路径
- keystorePass 就是4.1中创建证书的密码

5.2. 验证HTTPS配置

其他按照默认配置不作修改，双击%TOMCAT_HOME%\bin\startup.bat 启动tomcat-cas 验证https访问配置:





如果看到上述界面表示https 访问配置成功。

5.3 部署CAS-Server

CAS-Server 下载地址：http://www.jasig.org/cas/download

本文以cas-server-3.4.11-release.zip 为例，解压提取cas-server-3.4.11/modules/cas-server-webapp-3.4.11.war文件，把改文件copy到 G:\sso\tomcat-cas\webapps\ 目下，并重命名为：cas.war.
启动tomcat-cas，在浏览器地址栏输入：https://demo.micmiu.com:8443/cas/login，回车

CAS-server的默认验证规则：只要用户名和密码相同就认证通过（仅仅用于测试，生成环境需要根据实际情况修改），输入admin/admin 点击登录，就可以看到登录成功的页面：

看到上述页面表示CAS-Server已经部署成功。

六、部署CAS-Client相关的Tomcat

6.1Cas-Client 下载

CAS-Client 下载地址：http://downloads.jasig.org/cas-clients/
以cas-client-3.2.1-release.zip 为例，解压提取cas-client-3.2.1/modules/cas-client-core-3.2.1.jar
借以tomcat默认自带的 webapps\examples 作为演示的简单web项目

6.2 安装配置 tomcat-app1

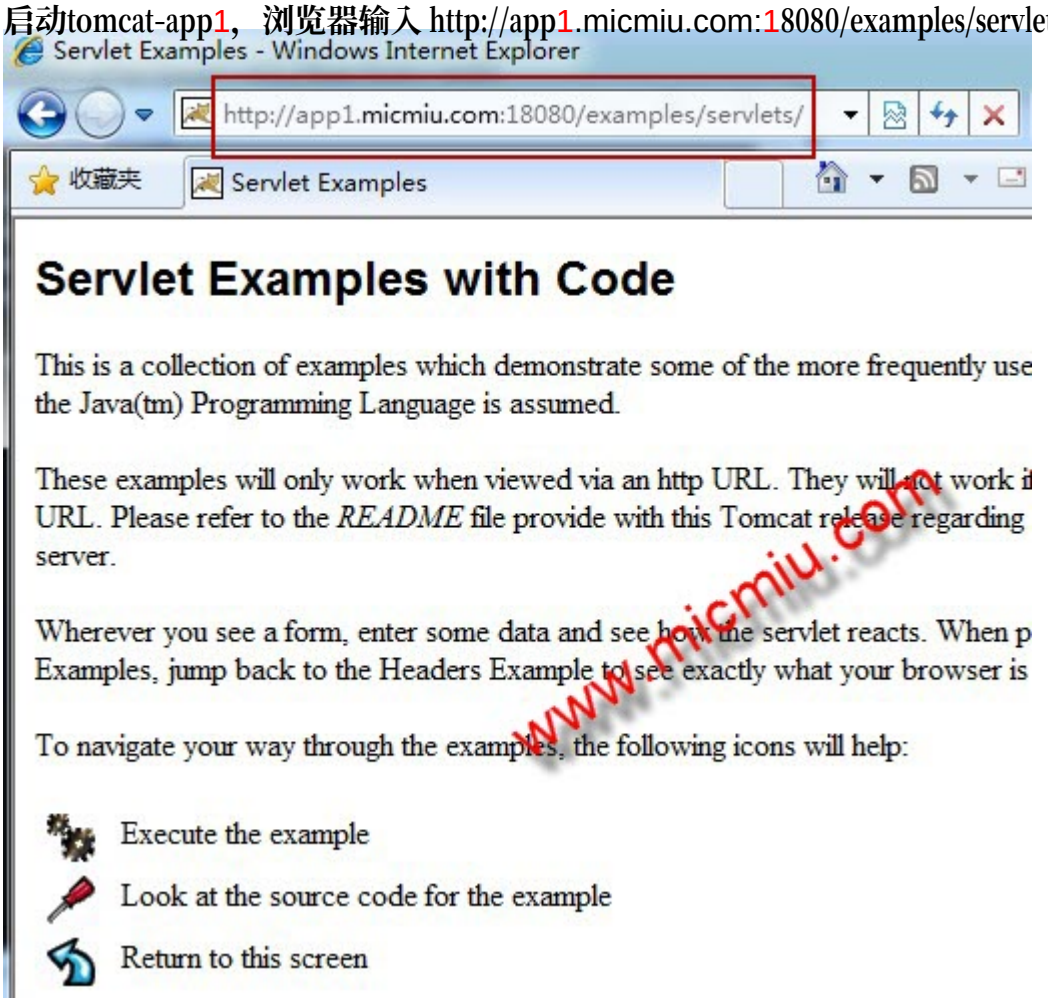
解压apache-tomcat-6.0.29.tar.gz并重命名后的路径为 G:\sso\tomcat-app1，修改tomcat的启动端口，在文件 conf/server.xml文件找到如下内容：

```
<Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

修改成如下：

```
<Connector port="18080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="18443" />
<Connector port="18009" protocol="AJP/1.3" redirectPort="18443" />
```

启动tomcat-app1，浏览器输入 http://app1.micmiu.com:18080/examples/servlets/ 回车：



看到上述界面表示tomcat-app1的基本安装配置已经成功。

接下来复制 client的lib包cas-client-core-3.2.1.jar到 tomcat-app1\webapps\examples\WEB-INF\lib\目录下， 在tomcat-app1\webapps\examples\WEB-INF\web.xml 文件中增加如下内容：

```
<!-- ========================= 单点登录开始 ========================= -->
<!-- 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>

<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>https://demo.micmiu.com:8443/cas/login</param-value>
</init-param>
<init-param>
<param-name>serverName</param-name>
<param-value>http://app1.micmiu.com: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>https://demo.micmiu.com:8443/cas</param-value>
</init-param>
<init-param>
<param-name>serverName</param-name>
<param-value>http://app1.micmiu.com: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 ========================= -->
```

有关cas-client的web.xml修改的详细说明见官网介绍：
https://wiki.jasig.org/display/CASC/Configuring+the+JA-SIG+CAS+Client+for+Java+in+the+web.xml

6.3 安装配置 tomcat-app2

解压apache-tomcat-6.0.29.tar.gz并重命名后的路径为 G:\sso\tomcat-app2，修改tomcat的启动端口，在文件 conf/server.xml文件找到如下内容：

```
<Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

修改成如下：

```
<Connector port="28080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="28443" />
<Connector port="28009" protocol="AJP/1.3" redirectPort="28443" />
```

启动tomcat-app2，浏览器输入 http://app2.micmiu.com:28080/examples/servlets/ 回车，按照上述6.2中的方法验证是否成功。

同6.2中的复制 client的lib包cas-client-core-3.2.1.jar到 tomcat-app2\webapps\examples\WEB-INF\lib\目录下， 在tomcat-app2\webapps\examples\WEB-INF\web.xml 文件中增加如下内容：

```
<!-- ========================= 单点登录开始 ========================= -->
<!-- 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>

<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>https://demo.micmiu.com:8443/cas/login</param-value>
</init-param>
<init-param>
<param-name>serverName</param-name>
<param-value>http://app2.micmiu.com:28080</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>https://demo.micmiu.com:8443/cas</param-value>
</init-param>
<init-param>
<param-name>serverName</param-name>
<param-value>http://app2.micmiu.com:28080</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 ========================= -->
```

七、 测试验证SSO

启动之前配置好的三个tomcat分别为：tomcat-cas、tomcat-app1、tomcat-app2.

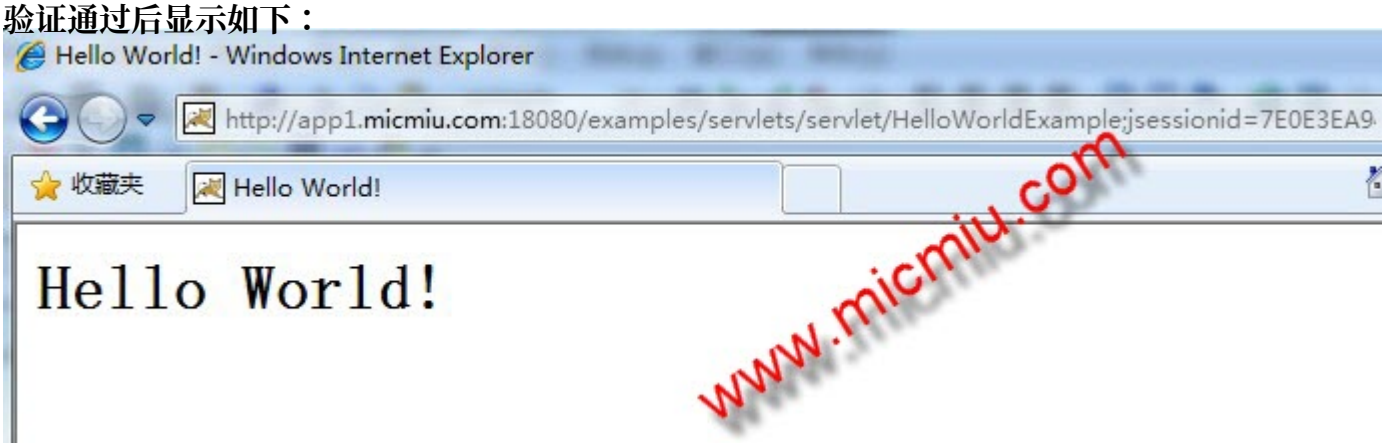
7.1 基本的测试

预期流程： 打开app1 url —-> 跳转cas server 验证 —-> 显示app1的应用 —-> 打开app2 url —-> 显示app2 应用 —-> 注销cas server —-> 打开app1/app2 url —-> 重新跳转到cas server 验证.

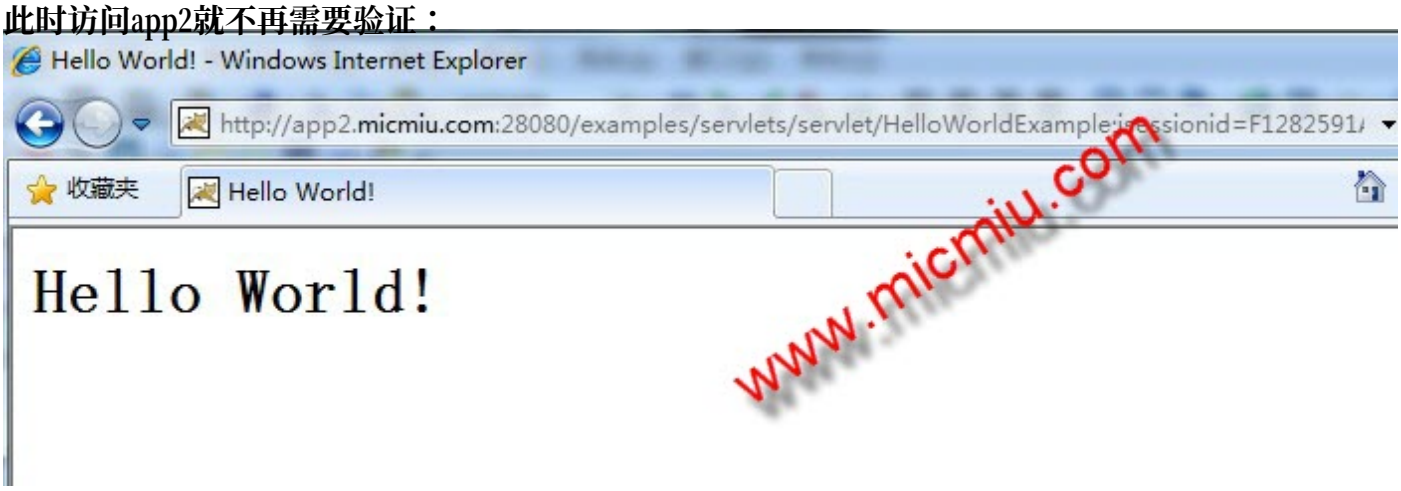
打开浏览器地址栏中输入：http://app1.micmiu.com:18080/examples/servlets/servlet/HelloWorldExample，回车：

跳转到验证页面：

验证通过后显示如下：



此时访问app2就不再需要验证：



地址栏中输入：https://demo.micmiu.com:8443/cas/logout，回车显示：

上述表示 认证注销成功，此时如果再访问：
http://app1.micmiu.com:18080/examples/servlets/servlet/HelloWorldExample 或
http://app2.micmiu.com:28080/examples/servlets/servlet/HelloWorldExample 需要重新进行认证。

7.2 获取登录用户的信息

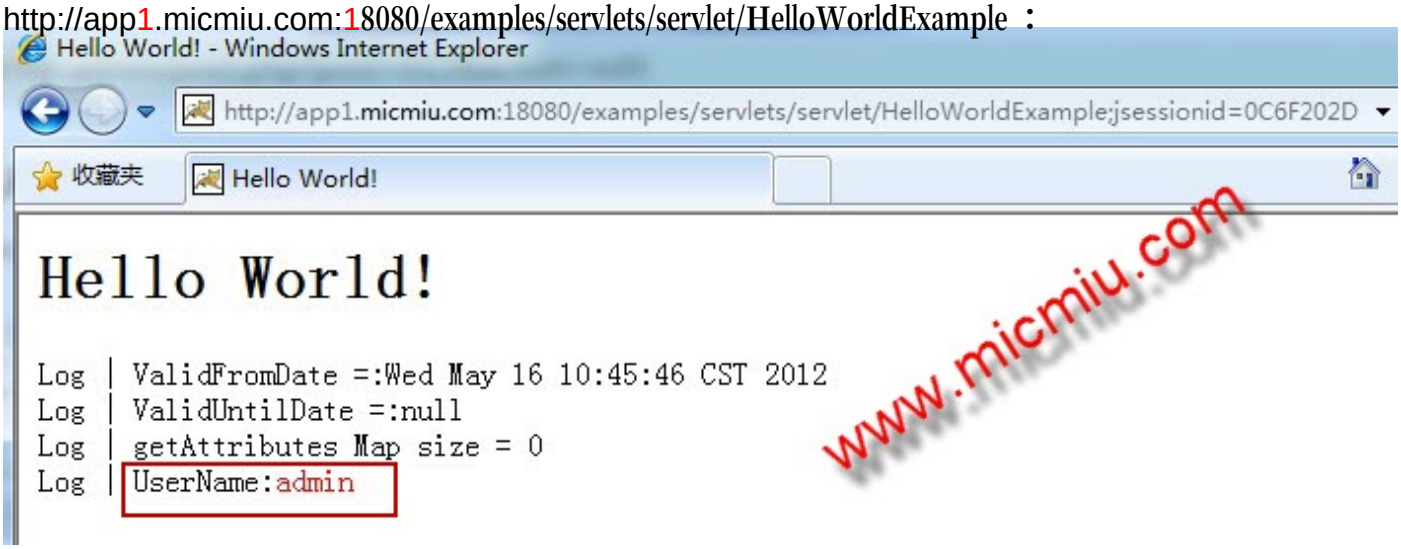
修改HelloWorldExample.java,重新编译替换webapps\examples\WEB-INF\classes\HelloWorldExample.class文件,修改后的HelloWorldExample.java代码如下：

```
import java.io.*;
import java.util.*;
import java.util.Map.Entry;

import javax.servlet.*;
import javax.servlet.http.*;

import org.jasig.cas.client.authentication.AttributePrincipal;
import org.jasig.cas.client.util.AbstractCasFilter;
import org.jasig.cas.client.validation.Assertion;

/**
 * The simplest possible servlet.
 *
 * @author James Duncan Davidson
 */

public class HelloWorldExample extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException {
ResourceBundle rb = ResourceBundle.getBundle("LocalStrings", request
.getLocale());
response.setContentType("text/html");
```

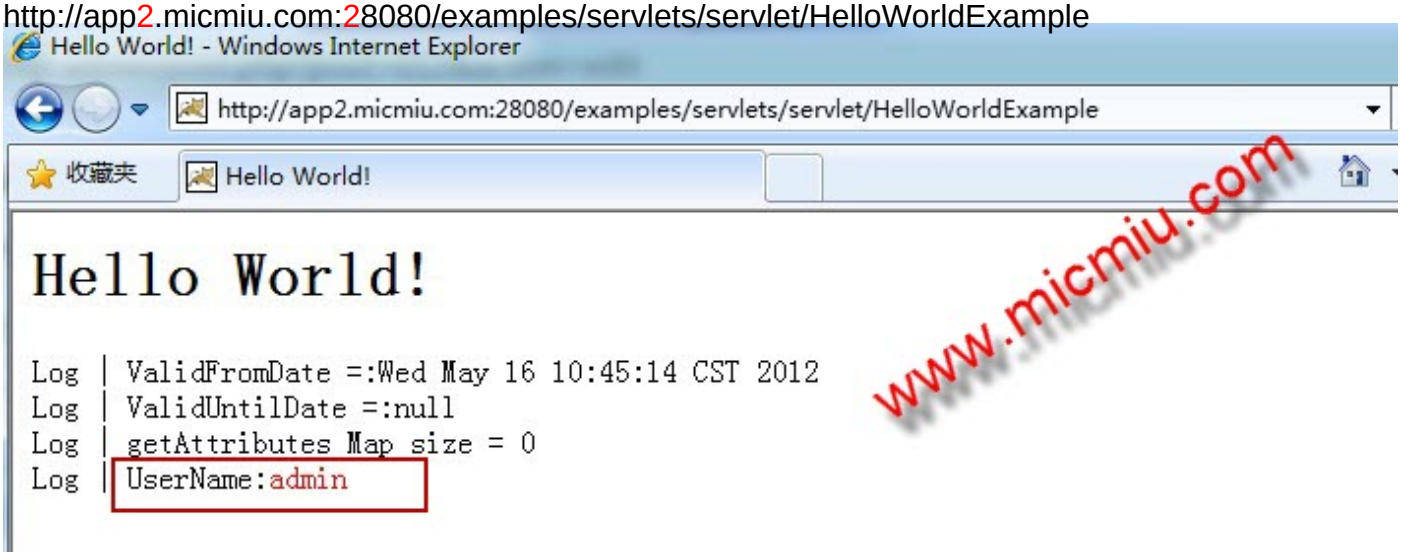
```
PrintWriter out = response.getWriter();

out.println("<html>");
out.println("<head>");

String title = rb.getString("helloworld.title");

out.println("<title>" + title + "</title>");
out.println("</head>");
out.println("<body bgcolor=\"white\">");

out.println("<a href=\"../helloworld.html\">");
out.println("<img src=\"../images/code.gif\" height=24 "
+ "width=24 align=right border=0 alt=\"view code\"></a>");
out.println("<a href=\"../index.html\">");
out.println("<img src=\"../images/return.gif\" height=24 "
+ "width=24 align=right border=0 alt=\"return\"></a>");
out.println("<h1>" + title + "</h1>");

Assertion assertion = (Assertion) request.getSession().getAttribute(
AbstractCasFilter.CONST_CAS_ASSERTION);

if (null != assertion) {
out.println(" Log | ValidFromDate =:"
+ assertion.getValidFromDate() + "<br>");
out.println(" Log | ValidUntilDate =:"
+ assertion.getValidUntilDate() + "<br>");
Map<Object, Object> attMap = assertion.getAttributes();
out.println(" Log | getAttributes Map size = " + attMap.size()
+ "<br>");
for (Entry<Object, Object> entry : attMap.entrySet()) {
out.println("     | " + entry.getKey() + "=:"
+ entry.getValue() + "<br>");
}

}
AttributePrincipal principal = assertion.getPrincipal();

// AttributePrincipal principal = (AttributePrincipal) request
// .getUserPrincipal();

String username = null;
out.print(" Log | UserName:");
if (null != principal) {
username = principal.getName();
out.println("<span style='color:red;'>" + username + "</span><br>");
}

out.println("</body>");
out.println("</html>");
}
}
```

再进行上述测试显示结果如下：

http://app1.micmiu.com:18080/examples/servlets/servlet/HelloWorldExample ：



http://app2.micmiu.com:28080/examples/servlets/servlet/HelloWorldExample



从上述页面可以看到通过认证的用户名。

到此已经全部完成了CAS单点登录实例演示。