用户认证系统

# 1 单点登录系统介绍

应用系统A

Cluster

(user-session)

SSO

token

应用系统B

Cluster

(user-session)

应用系统C

Cluster

(user-session)

E

token

# 2 webservice

1 基于soap协议(对http的post的封装)

Soap风格

消息格式统一，有成熟的实现框架(cxf)

2 基于http协议(原始协议)

Rest风格

效率高，调用方便，没有统一的协议格式，没有成熟的实现框架

3 cxf整合spring搭建ws框架

|  |
| --- |
| 1 支持soap协议接口服务  2 支持rest风格协议接口服务 |

# 3 用户认证系统框架搭建

|  |
| --- |
| MVC：Spring + mybatis + springmvc  RPC：Spring + mybatis + CXF |

# 4 spring整合jaxfactory

Bean

Factorybean

实现类（程序代码）

# 5 任务

1 完成cxfuser框架的搭建

2 整合交易系统

3 完成用户接口服务、地址接口服务

4 完成多数据源的整合

# 6 多数据源

数据源

业务系统

数据源1

数据源路由器(切换)

Key=1

业务系统

数据源2

数据源3

线程1

Key

Key=1

1

2

线程2

Key

# 7 webservice安全问题

1 使用接口的客户端的身份验证

UsernameToken

|  |
| --- |
| 1 在服务器上加入WSS4JInInterceptor  2 配置拦截器的类型、密码格式、回掉函数  3 自定义拦截器回掉函数(对拦截器的用户名和密码进行设置)  4 根据服务器的注册的UsernameToken完善客户端代理拦截器WSS4JOutInterceptor |

|  |
| --- |
| <soapenv:Header>  **<wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">**  <wsse:UsernameToken >  <wsse:Username>cxf</wsse:Username>  <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#**PasswordText**">wss4j</wsse:Password>  </wsse:UsernameToken>  **</wsse:Security>**  </soapenv:Header> |

2 使用接口发送信息的加密问题

根据时间或者次数自动更新token

|  |
| --- |
| 1 加入时间要素，  服务器在保存token时，将token和时间字符串拼接后进行md5加密，这样可以根据时间的变化每次获得不同额token  2 时间的处理  A 时间格式问题  B 时间日期的加减 |

# 8 配置rest风格ws

|  |
| --- |
| 1 配置pom依赖  2 在spring中发布rest风格的接口服务  3 rest风格接口服务的参数和返回值的设置  4 如何调用 |

自己理解:

服务端工作：在spring容器中管理，

<jaxws:endpoint address=*"/user"*

implementorClass=*"com.atguigu.service.UserServerInf"*>

<jaxws:implementor>

<bean class=*"com.atguigu.service.UserServerImp"*></bean>

</jaxws:implementor>

<jaxws:inInterceptors>

<ref bean=*"in"*></ref>

</jaxws:inInterceptors>

</jaxws:endpoint>

这种方式在tomcat启动时就启动接口服务。

配置密码回调函数。在函数中自定义回调密码加密规则。注意和客户端保持一致。

客户端工作。

由于客户端只有在访问时才需要调用服务端的接口服务。所以不能在spring容器中管理。

需要进行的工作。

最好能将接口注入到spring容器中进行托管，需要写一个工具类继承factorybean，也需要写回调密码的加密规则，要和服务端保证一致。

注意在继承factorybean的返回接口实例方法中要 加入if判断，否则会有understood异常

**if** (t.getSimpleName().equals("UserServerInf")) {

Map<String, Object> map = **new** HashMap<String, Object>();

map.put(WSHandlerConstants.***ACTION***, WSHandlerConstants.***USERNAME\_TOKEN***);

map.put(WSHandlerConstants.***PASSWORD\_TYPE***, "PasswordText");

map.put(WSHandlerConstants.***PW\_CALLBACK\_CLASS***, MyPasswordCallBack.**class**.getName());

map.put(WSHandlerConstants.***USER***, "username");

WSS4JOutInterceptor wss4jOutInterceptor = **new** WSS4JOutInterceptor(map);

jax.getOutInterceptors().add(wss4jOutInterceptor);

}