Flex+Red5语音视频聊天程序实现

 (2011-12-28 19:52:55)

[转载▼](javascript:;)



|  |  |
| --- | --- |
| 标签：  [杂谈](http://search.sina.com.cn/?c=blog&q=%D4%D3%CC%B8&by=tag) |  |

**java代码部分**

**package first;  
import java.util.HashMap;  
import java.util.Iterator;  
import java.util.Map;  
import java.util.Set;**

**import org.red5.server.adapter.ApplicationAdapter;  
import org.red5.server.api.IConnection;  
import org.red5.server.api.IScope;  
import org.red5.server.api.Red5;  
import org.red5.server.api.service.IServiceCapableConnection;  
import org.red5.server.api.so.ISharedObject;**

**public class Application extends ApplicationAdapter {    
 // 属性  
 private IScope appScope;  
   
 private String userName;  
 //共享存储在线用户  
 private ISharedObject listSO;   
 private Map<String,IConnection> onlineList = new HashMap<String, IConnection>();// 在线用户表  
 //程序运行时志向  
 public boolean appStart(IScope app) {  
  if (!super.appStart(app)) {  
      return false;  
  }  
  appScope = app;  
  return true;  
 }  
 @Override  
 public boolean appConnect(IConnection arg0, Object[] arg1) {  
    
  String userId=arg0.getClient().getId();  
  if(!super.appConnect(arg0, arg1)){  
   return false;  
  }  
  if (arg1 != null ) {  
   userName = (String) arg1[0];  
  }  
  if(onlineList.get(userName) != null){  
   rejectClient("请不要重复登录");  
   return false;  
  }  
  onlineList.put(userName, arg0);  
  listSO = getSharedObject(appScope, "listSO", false);   
  listSO.setAttribute(userId, userName);  
  System.out.println("The user:"+userName+","+userName+" logined successfully");  
  return true;  
 }  
   
 public void getOnloadUser(Object[] params) {     
  String clientName = params[0].toString();    
  if(null == clientName || "".equals(clientName)) {  
     return ;  
   }  
  //给所有客户端数据  
   IScope scope = Red5.getConnectionLocal().getScope();  
   Iterator it = scope.getConnections().iterator();  
   for (;it.hasNext();) {  
    Set connections = (Set)it.next();  
    IConnection tempConn = (IConnection)connections.iterator().next();  
    if (tempConn instanceof IServiceCapableConnection) {  
     IServiceCapableConnection sc = (IServiceCapableConnection) tempConn;  
     sc.invoke("result\_getOnloadUser", new Object[]{clientName});  
    }  
   }  
 }  
 //聊天  
  public void sayToAll(Object[] params) {  
   IConnection conn = Red5.getConnectionLocal();  
   String user\_id = conn.getClient().getId();  
   String clientName =(String) listSO.getAttribute(user\_id);  
   System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*发言者是："+clientName);  
   String sayToName=params[0]==null?"":params[0].toString().trim();    
   String sayWhat=params[1]==null?"":params[1].toString().trim();  
   if("".equals(sayToName)||"All".equals(sayToName))// 发消息给聊天室的所有人.  
   {  
    IScope scope = Red5.getConnectionLocal().getScope();  
    Iterator it = scope.getConnections().iterator();   
    for (;it.hasNext();) {  
      Set connections = (Set)it.next();  
      IConnection tempConn = (IConnection)connections.iterator().next();  
     if (tempConn instanceof IServiceCapableConnection) {  
         IServiceCapableConnection sc = (IServiceCapableConnection) tempConn;  
      // 调用客户端showMessage方法。  
         sc.invoke("showMessage", new Object[]{clientName+" to All:"+sayWhat});  
      }  
   }  
   }else{  
     IConnection tempConn=onlineList.get(sayToName);       
     if (tempConn instanceof IServiceCapableConnection) {  
      IServiceCapableConnection sc = (IServiceCapableConnection) tempConn;       
      sc.invoke("showMessage", new Object[]{clientName+" to "+sayToName+":"+sayWhat});  
      }  
     IServiceCapableConnection sc = (IServiceCapableConnection) conn;  
     sc.invoke("showMessage", new Object[]{clientName+" to "+sayToName+":"+sayWhat});  
   }  
  }  
 // 用户断开连接的时候触发  
  public void appDisconnect(IConnection conn) {  
   String dis\_user\_id = conn.getClient().getId();  
   String user = (String) listSO.getAttribute(dis\_user\_id);  
   // 根据ID删除对应在线纪录  
   onlineList.remove(user);  
   // 删除用户列表共享对象的对应属性  
   listSO.removeAttribute(dis\_user\_id);  
   IScope scope = Red5.getConnectionLocal().getScope();  
    Iterator it = scope.getConnections().iterator();   
    for (;it.hasNext();) {  
     Set connections = (Set)it.next();  
     IConnection tempConn = (IConnection)connections.iterator().next();  
     if (tempConn instanceof IServiceCapableConnection) {  
         IServiceCapableConnection sc = (IServiceCapableConnection) tempConn;  
       // 服务器端调用客户端flash方法。  
         sc.invoke("disconnectMessage", new Object[]{user});  
      }  
    }  
  }  
 //视频邀请  
  public void videoInvite(Object[] params) {  
   IConnection conn = Red5.getConnectionLocal();  
   String user\_id = conn.getClient().getId();  
   String clientName =(String) listSO.getAttribute(user\_id);  
   System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*视频邀请者是："+clientName);  
   String sayToName=params[0]==null?"":params[0].toString().trim();  
   if("".equals(sayToName)||"All".equals(sayToName))// 发消息给聊天室的所有人.  
   {  
   System.out.println("不可以邀请0或者多个人");    
   }else{  
     IConnection tempConn=onlineList.get(sayToName);       
     if (tempConn instanceof IServiceCapableConnection) {  
      IServiceCapableConnection sc = (IServiceCapableConnection) tempConn;       
      sc.invoke("showInviteMessage", new Object[]{clientName+";"+sayToName});  
      }  
   }  
 }  
  //同意邀请后调用邀请方方法  
  public void agreeVideoInvite(Object[] params) {  
    IConnection conn = Red5.getConnectionLocal();  
    System.out.println("<<<<<params length"+params.length);  
    //邀请者  
    String inviteUserName=params[0]==null?"":params[0].toString().trim();  
    //被邀请者  
    String otherUserName=params[1]==null?"":params[1].toString().trim();  
    System.out.println("\*\*\*邀请者是:"+inviteUserName);  
    System.out.println("\*\*\*被邀请者是:"+otherUserName);  
    if("".equals(inviteUserName))// 发消息给聊天室的所有人.  
    {  
    System.out.println("出错了");    
    }else{  
      System.out.println("\*\*\*\*\*\*\*\*\*调用成功");  
      IConnection tempConn=onlineList.get(inviteUserName);       
      if (tempConn instanceof IServiceCapableConnection) {  
       IServiceCapableConnection sc = (IServiceCapableConnection) tempConn;       
       sc.invoke("showVideo", new Object[]{otherUserName});  
       System.out.println("\*\*\*\*\*\*\*\*\*调用结束");  
       }  
    }  
  }  
}**

**Flex3代码部分**

**<?xml version="1.0" encoding="utf-8"?>  
<mx:WindowedApplication xmlns:mx="**[**http://www.adobe.com/2006/mxml**](http://www.adobe.com/2006/mxml)**" layout="absolute">  
 <mx:Script>  
  <![CDATA[  
   import mx.events.CloseEvent;  
   import mx.collections.ArrayCollection;  
   import mx.controls.Alert;  
     
   private var listSO:SharedObject;  
   private var userArr:Array;    
   private var conn:NetConnection;  
   private var localUsername:String;  
   private var stm:NetStream;  
   private var stm2:NetStream;  
   //摄像头  
   private var cam:Camera;    
   //麦克风  
   private var mic:Microphone;  
   private var video\_0:Video;  
   private var video\_1:Video;  
   [Bindable]  
   public var cards:ArrayCollection;  
   //视频邀请集合  
   public var videoUsers:Array;  
   protected function login(event:MouseEvent):void  
   {  
    // TODO Auto-generated method stub      
    localUsername= txt\_name.text;  
    if(localUsername==""){  
     Alert.show("用户名不能为空");  
     return;  
    }else{  
     if(conn==null){  
      conn = new NetConnection();  
      conn.client = this;  
      conn.addEventListener(NetStatusEvent.NET\_STATUS, \_statusHandler);            
      conn.connect("rtmp://192.168.0.10/szftest", localUsername);  
     }  
    }  
   }  
   //状态监听  
   private function \_statusHandler(evt:NetStatusEvent):void {  
    if (evt.info.code == "NetConnection.Connect.Success") {  
     this.currentState = "chat";  
     Alert.show("连接成功");  
     \_setListSO();      //创建用户列表共享对象  
     this.showJoinInInfo(localUsername);   
    }  
    if (evt.info.code == "NetConnection.Connect.Failed") {  
     Alert.show("连接失败");   
    }  
    if (evt.info.code == "NetConnection.Connect.Closed") {  
     Alert.show("连接关闭");  
    }  
   }  
   public function showJoinInInfo(message:String) : void  
   {  
    //conn.call("getOnloadUser", new Responder(result\_getUsers);       
    conn.call("getOnloadUser", null,message);  
   }  
   public function result\_getOnloadUser(str:String):void {   
    txt\_chatmsg.text += str + "加入聊天室" + "\n";      
   }  
   //创建用户列表共享对象  
   private function \_setListSO():void{  
    listSO = SharedObject.getRemote("listSO", conn.uri, false);  
    listSO.connect(conn);  
    listSO.addEventListener(SyncEvent.SYNC,\_listSOSyncHandler);  
   }  
   //用户列表共享对象被更新之后事件  
   private function \_listSOSyncHandler(evt:SyncEvent):void{  
    \_showUserList();     //更新用户列表  
   }  
   //更新用户列表  
   private function \_showUserList():void{  
    cards = new ArrayCollection(  
     [ {label:"All"}]  
    );  
    userArr = new Array();  
    //用户数组更新  
    for (var tmp:String in listSO.data) {  
     userArr.push(listSO.data[tmp]);  
    }  
    //添加ArrayCollection       
    for (var i:int = 0; i < userArr.length; i++ ) {  
     cards.addItem( { label:userArr[i] } );  
    }  
    //将数组添加到列表中显示出来  
    userList.dataProvider = cards;  
    users.dataProvider=cards;  
   }  
     
   public function showMessage(message:String) : void  
   {  
    txt\_chatmsg.text += message + "\n";  
   }  
   protected function sendMessage(event:MouseEvent):void  
   {  
    // TODO Auto-generated method stub  
    var sendString:String = txt\_yousay.text;  
    //var sendTo:String=users.selectedItem.userName==null?"":users.selectedItem.userId;  
    var sendTo:String=userList.selectedItem.label;      
    txt\_yousay.text = "";  
    conn.call("sayToAll", null,sendTo,sendString);  
   }  
   //断线通知  
   public function disconnectMessage(disUser:String):void{  
    txt\_chatmsg.text += disUser + "退出聊天室\n";  
   }  
   //视频邀请  
   protected function video\_clickHandler(event:MouseEvent):void  
   {  
    // TODO Auto-generated method stub  
    var sendTo:String=userList.selectedItem.label;  
    conn.call("videoInvite", null,sendTo);      
   }  
   //显示自己的视频  
   public function \_publishVideo():void{  
    stm = new NetStream(conn);  
    cam = Camera.getCamera();  
    if(cam==null)  
    {  
     Alert.show("没有可以使用的摄像头");  
     return;  
    }else{  
     cam.setLoopback(true);  
                 cam.setMotionLevel(50,100);  
                 cam.setMode(320,240,15,true);  
                 stm.attachCamera(cam);  
    }  
    //获取麦克风  
    mic = Microphone.getMicrophone();  
    if(mic==null){  
     Alert.show("没有可以使用的麦克风");  
    }else{  
     mic.setUseEchoSuppression(true);  
     stm.attachAudio(mic);  
    }      
    stm.publish("somes"+txt\_name.text,"live");      
    video\_1 = new Video();  
    video\_1.width = my\_video.width;  
    video\_1.height = my\_video.height;  
    video\_1.attachCamera(cam);  
    my\_video.addChild(video\_1);  
   }  
   //被邀请方获取对方视频  
   private function \_getVideo():void{      
    stm2 = new NetStream(conn);  
    video\_0 = new Video(other\_video.width,other\_video.height);  
    video\_0.attachNetStream(stm2);  
    other\_video.addChild(video\_0);  
    stm2.play("somes"+videoUsers[0]);  
   }  
   //邀请方获取对方视频  
   private function \_getOtherVideo(other:String):void {      
    stm2 = new NetStream(conn);  
    video\_0 = new Video(other\_video.width,other\_video.height);  
    video\_0.attachNetStream(stm2);  
    other\_video.addChild(video\_0);  
    stm2.play("somes"+other);  
   }  
   //视频被邀请方调用  
   public function showInviteMessage(message:String):void{  
    videoUsers=message.split(";");  
    Alert.show(videoUsers[0]+"邀请你视频，是否同意?","是吗",Alert.YES|Alert.NO,this,alertClientHandel);  
   }  
   //被邀请方同意调用  
   public function alertClientHandel(event:CloseEvent):void{  
    if(event.detail==Alert.YES){  
     \_publishVideo();       
     \_getVideo();  
     conn.call("agreeVideoInvite",null,videoUsers[0],videoUsers[1]);  
    }else{  
       
    }  
   }  
   //邀请方接到被邀请方的同意后调用  
   public function showVideo(other:String):void{      
    \_publishVideo();  
    \_getOtherVideo(other);  
   }  
   private function closeStream():void{  
    Alert.show("关闭");  
   }  
     
  ]]>  
 </mx:Script>  
 <mx:states>  
  <mx:State name="chat">  
   <mx:SetProperty target="{form1}" name="width" value="0"/>  
   <mx:SetProperty target="{form1}" name="height" value="0"/>  
   <mx:SetProperty target="{form1}" name="x" value="0"/>  
   <mx:SetProperty target="{form1}" name="y" value="0"/>  
   <mx:AddChild position="lastChild">  
    <mx:Panel x="10" y="10" width="381" height="370" layout="absolute" title="聊天信息">  
     <mx:TextArea x="10" y="10" width="215" height="235" id="txt\_chatmsg"/>  
     <mx:ComboBox x="233" y="34" width="118" id="userList"></mx:ComboBox>  
     <mx:Label x="233" y="11" text="用户列表"/>  
     <mx:DataGrid x="233" y="64" height="256" id="users" width="118">  
      <mx:columns>  
       <mx:DataGridColumn headerText="用户名" dataField="label"/>  
      </mx:columns>  
     </mx:DataGrid>  
     <mx:TextInput x="10" y="253" height="67" width="150" id="txt\_yousay"/>  
     <mx:Button x="168" y="253" label="发送" width="57" click="sendMessage(event)"/>  
     <mx:Button x="168" y="253" label="发送" width="57" click="sendMessage(event)"/>  
     <mx:Button x="168" y="298" label="视频" width="57" click="video\_clickHandler(event)"/>  
    </mx:Panel>  
   </mx:AddChild>  
   <mx:AddChild position="lastChild">  
    <mx:VideoDisplay x="399" y="30" width="181" height="158" id="my\_video"/>  
   </mx:AddChild>  
   <mx:AddChild position="lastChild">  
    <mx:Label x="399" y="10" text="我的"/>  
   </mx:AddChild>  
   <mx:AddChild position="lastChild">  
    <mx:VideoDisplay x="399" y="224" width="181" height="156" id="other\_video"/>  
   </mx:AddChild>  
   <mx:AddChild position="lastChild">  
    <mx:Label x="399" y="198" text="对方的"/>  
   </mx:AddChild>  
  </mx:State>  
 </mx:states>  
   
 <mx:Form x="10" y="10" width="283" height="126" id="form1">  
  <mx:FormItem label="用户名：">  
   <mx:TextInput id="txt\_name"/>  
  </mx:FormItem>  
  <mx:FormItem>  
   <mx:Button label="登陆" click="login(event)"/>  
  </mx:FormItem>  
 </mx:Form>  
   
</mx:WindowedApplication>**