

用AJAX J2EE实现网上会议室系统 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/144/2021\\_2022\\_\\_E7\\_94\\_A8AJAX\\_J2\\_c104\\_144440.htm](https://www.100test.com/kao_ti2020/144/2021_2022__E7_94_A8AJAX_J2_c104_144440.htm) 今年大家都在炒作Web2.0，其中的一门技术Ajax也是跟着火了起来，因此前面我写了一篇名为《忽悠一下AJAX》的文章，简单地分析了一下Ajax的技术的实质。虽然笔者不太喜欢跟风，但Ajax有一些地方还是比较有用的。前段时间做了EasyJF开源团队的网上会议系统，就用到了Ajax技术，下面把设计思路发出来跟大家分享一下。

一、系统实现的功能 本会议室系统主要用于EasyJF开源团队的成员网上会议使用，会议系统模拟传统的会议形式，可以同时开设多个不同主题的会议室，每个会议室需要提供访问权限控制功能，会议中能够指定会议发言模式（分为排队发言、自由发言两种），系统能自动记录每个会议室的发言信息，可以供参会人员长期查阅。会议系统的用户支持游客帐号参加会议，同时也提供跟其它用户系统的接口，比如EasyJF官网中的开源论坛系统。会议系统暂时使用文字聊天的方式，并提供语音及视频的接口。

二、技术体系 服务器端使用Java语言，MVC使用EasyJWeb框架；客户端使用AJAX技术与服务器端交互数据；会议历史信息储存格式使用文本格式，方便系统安装运行，也便于管理。

三、会议室服务器端设计 会议室服务器端是整个会议系统的核心部分，服务器端程序设计的好坏影响到整个系统的质量。首先，根据会议室要实现的功能进行抽象分析。一个会议室对象，应该包括会议主题、会议简介、参会人数限制、公告、会议室类型、访问权限设定、房间密码、当前参会的人员、当前发言的人员、排队等

待发言的人员等参数信息。我们把他封装一个Java对象当中。如下面的ChatRoom代码所示：

```
public class ChatRoom{ private String cid;//主键 private String title;//会议室主题 private String intro;//会议室简介 private String announce;//会议室公告 private String owner;//会议室创建人 private Integer maxUser;//最大在线人数 private Integer intervals;//最大刷新时间间隔 private String vrtype;//访问权限 private String vrvalue;//访问值 private Integer status;//会议室状态 private Date inputTime.}
```

需要一个管理会议室的类，与会议有关的操作（如启动会议、关闭会议）等都直接找他。该类还应该即有自动定时检测用户在线情况（防止用户意外退出）、把内存中的会议历史发言信息保存到文本文件中等功能。这里可以考虑使用一个ChatService类提供这些功能：

```
public class ChatService implements Runnable {private static final Map service=new HashMap();//会议室服务,系统中的当前会议室存放到该表集合中private static final int maxServices=10;//可以同时开的最大会议室数private static final SimpleDateFormat df=new SimpleDateFormat("yyyy-MM-dd").private final List msgs;//会议发言信息Chatprivate final List users;//在线用户,ChatUserprivate final List talkers;//排队发言人数Talkerprivate final List manager;//会议室管理员private Talker currentTalker;//当前发言人public ChatService(){ this.msgs=new ArrayList(). this.users=new ArrayList(). this.talkers=new ArrayList(). this.manager=new ArrayList(). this.maxUser=1000;//最大1000人同时 this.interval=1000*60*5;//5分钟以前的信息}}
```

会议发言信息也需要封装成一个类，表示发言人、接收人、内容、发言时间、

类型等，大致如下面的Chat类：`public class Chat {private String cid.private String sender.private String reciver.private String content.private Date vdate.private Integer types.private Integer status.}` 还有表示参加会议的人的信息，包括参会人名称、IP地址、状态等，如下面的ChatUser类所示：`public class ChatUser {private String ip.private String port.private String userName.private Date lastAccessTime.private Integer status.}` 另外还需要一个表示当前发言人的Talker类，表示当前的发言人，发言开始时间，发言预计结束时间等。在服务器端的设计中，会议室信息服务器应该能以多线程的方式运行，即启动一个会议就新开一个线程，每个会议线程维护自己的会议状态，如参会人、发言人，保存会议历史发言信息以及清空内存中的数据等操作。

#### 四、客户端设计

会议室客户端包括两个部分，一个部分是会议室的管理界面，主要包会议室的“添删改查”及“启动”或“关闭”会议服务的操作。这部分我们直接使用EasyJWeb Tools中的添删改查业务引擎AbstractCrudAction可以快速实现。界面也比较简单，直接使用EasyJWeb Tools代码生成工具引擎生成即可。会议室管理的客户端是传统的Java Web技术，因此没有什么要考虑的。客户端的第二个部分也即会议系统的主要部分，该部分主要有两个界面，第一个页面是会议室进入的选择页面。也即把已经启动的会议室列出来，用户选择一个会议室进入，这个页面也是使用传统的Java Web技术。第二个页面是进入会议室后的主界面，这个界面是整个会议系统的主要界面，所有参与会议的操作都在这里运行的。这个界面需要不断的与服务器端交互传输数据，传输的内容包括用户的发言、其它人

给用户的发言、会议室的状态等。有的传输信息需要即时响应（如用户发言），有的信息可以设置成定时响应（如会议室状态）。Java Web程序中与服务器端交互数据主要有两种方式，一种是直接刷新页面，另外一种是使用Socket直接跟Web服务器端口通讯。由于Socket编程相对复杂，我们选择第一种直接刷新页面的方式，这种方式又可以分为几种，包括传统的Form提交，传统的自动刷新网页取得数据以及使用ActiveXObject对象(如xmlhttp)直接与服务器交互数据，也即AJAX方式。由于使用AJAX方式用户感觉不到页面在刷新，表现起来好于手动或自动刷新页面的方式，因此我们决定选择AJAX方式实现客户端与服务器端进行数据交互。用户发言的时候，直接使用xmlhttp对象Post数据到服务器。为了能不断接收到别人的发言信息，需要定时不断的从服务器端读取数据，因此，需要在客户端启动一个定时器，每隔一定的时候自动使用xmlhttp对象到服务器端下载别人的发言信息，并显示到会议室信息主界面中。另外还要定时刷新参会的人数、会议室当前发言人、会议室的公告等会议状态信息，这也可以通过从客户端启动一个定时器，通过xmlhttp对象与服务器交互得到。另外还有一些操作，锁定会议室、踢人、指定发言人的发言时间、给会议室加密码等功能，也通过xmlhttp的方式与服务器传输命令实现。

### 五、核心代码说明

#### 1、服务器端核心代码

在EasyJF开源团队的会议系统中，由于是以EasyJF官网的论坛系统、后台管理等是集成一起的。服务器ChatService与ChatRoom共同合并到了一个ChatService.java类中，实现会议室管理及会议服务功能。ChatService类的部分主要代码如下：

```
package com.easyjf.chat.business;public class
```

```
ChatService implements Runnable {private static final Map
service=new HashMap().//会议室服务,系统中的当前会议室存
放到该表集合中private static final int maxServices=10.//可以同时
开的最大会议室数private static final SimpleDateFormat df=new
SimpleDateFormat("yyyy-MM-dd").private final List msgs.//会议发
言信息Chatprivate final List users.//在线用户,ChatUserprivate final
List talkers.//排队发言人数量private final List manager.//会议
室管理员private Talker currentTalker.//当前发言人private String
cid.//会议室idprivate String title.//会议室主题private String
intro.//会议室简介private String owner.//会议室创建人private int
maxUser.//最大在线人数private int interval.//最大刷新时间间
隔private String vrtype.//访问权限private String vrvalue.//访问
值private String announce.private String password.//房间进入密
码private int status.//会议室状态private String filePath.//private
Thread thread.private boolean isStop=false.public ChatService(){
this.msgs=new ArrayList(). this.users=new ArrayList().
this.talkers=new ArrayList(). this.manager=new ArrayList().
this.maxUser=1000.//最大1000人同时 this.interval=1000*60*5.//5
分钟以前的信息}/** * 停止所有会议室 */public static void
clear(){ if(!service.isEmpty()) { Iterator it=service.values().iterator().
while(it.hasNext()) { ChatService chat=(ChatService)it.next().
chat.stop(). } } service.clear().}/** * 创建一个会议室 * @param
name 会议室ID * @return */public static ChatService create(String
name){ChatService ret=null.if(service.containsKey(name)){
ChatService s=(ChatService)service.get(name). s.stop().
service.remove(name).}if(service.size()){ ret=new ChatService().
```

```

service.put(name,ret).}return ret.}/** * 停止某个会议室 * @param
name 会议室ID * @return */public static boolean close(String
name){ ChatService chatRoom=ChatService.get(name).
if(chatRoom!=null) { chatRoom.stop(). service.remove(name). }
return true.}/** * 获得一个会议室信息 * @param name 会议室ID
* @return */public static ChatService get(String name){
if(service.containsKey(name))return
(ChatService)service.get(name). else return null.}public void run() {
// TODO Auto-generated method stub
//this.thread=Thread.currentThread(). while(!isStop) {
//System.out.println("开始监控一个会议室!" this.title). this.flash().
try{ Thread.sleep(5000). } catch(Exception e) { e.printStackTrace().
} } //System.out.println("结束!").}public void stop(){ this.flashAll().
isStop=true.}//会议室中有人发言public boolean talk(Chat chat){
boolean ret=false. if(canTalk(chat.getSender())) {
this.msgs.add(chat). ret=true. } return ret.}public boolean
exit(ChatUser user){ talk(geneSystemMsg(user.getUserName() "退
出了会议室！")). return this.users.remove(user).} //刷新信息，
保存会议信息public void flash(){ flashChatMsg().
flashChatUser().} }
2、MVC处理部分的Action代码 在EasyJF的
会议系统中，由于使用EasyJWeb作为MVC框架，因此处
理Ajax比较简单，下面是会议室系统的核心Action主要代码
。 package com.easyjf.chat.action.public class ChatAction extends
AbstractCmdAction { private ChatService chatRoom. public Object
doBefore(WebForm form, Module module) { // TODO
Auto-generated method stub

```

```
if(chatRoom==null)chatRoom=ChatService.get((String)form.get("
cid")). return super.doBefore(form, module). } public Page
doInit(WebForm form, Module module) { // TODO
Auto-generated method stub return doMain(form,module). } //用
户登录进入会议室 public Page doMain(WebForm form, Module
module) { if(chatRoom!=null){ ChatUser user=getChatUser().
if(!chatRoom.join(user))form.setResult("msg","不能加入房间 ,
可能是权限不够 ! "). form.setResult("chatRoom",chatRoom).
form.setResult("user",user). } else { form.setResult("msg","会议未
启动或者会议室不存在 ! "). } return module.findPage("main"). }
//处理用户发言信息 public Page doSend(WebForm form,
Module module) { if(chatRoom==null)return new
Page("err","/err.html","thml").//返回会议室不存在的错误 Chat
chat=(Chat)form.toPo(Chat.class).
chat.setCid(chatRoom.genId()). chatRoom.talk(chat). return
doRecive(form,module). } //用户接收发言信息 public Page
doRecive(WebForm form, Module module) {
if(chatRoom==null)return new Page("err","/err.html","thml").//返
回会议室不存在的错误 String
lastReadId=CommUtil.null2String(form.get("lastReadId")).
//System.out.println(lastReadId). form.setResult("list",
chatRoom.getNewestMsg(getChatUser(),lastReadId)). return
module.findPage("msgList"). } //用户刷新会议状态信息 public
Page doLoadConfig(WebForm form, Module module) {
if(chatRoom==null)return new Page("err","/err.html","thml").//返
回会议室不存在的错误 form.setResult("userList",
```

```

chatRoom.getUsers()). form.addResult("talkerList",
chatRoom.getTalkers()). return module.findPage("config"). } //用
户退出 public Page doExit(WebForm form, Module module) {
if(chatRoom==null)return new Page("err","/err.html","thml").//返
回会议室不存在的错误 chatRoom.exit(getChatUser()).
form.addResult("msg","退出成功").
ActionContext.getContext().getSession().removeAttribute("chatUs
er"). return new
Page("msg","/chat/xmlMsg.xml",Globals.PAGE_TEMPLATE_TYP
E). }
3、 客户端AJAX部分核心代码 EasyJF会议系统中，服务器
发送给客户端的都是格式化的xml文档数据。下面是核心
的AJAX函数及发送接收会议信息的客户端代码。
function
newXMLHttpRequest() { var xmlreq = false. if
(window.XMLHttpRequest) { xmlreq = new XMLHttpRequest(). }
else if (window.ActiveXObject) { try { xmlreq = new
ActiveXObject("Msxml2.XMLHTTP"). } catch (e1) { try { xmlreq =
new ActiveXObject("Microsoft.XMLHTTP"). } catch (e2) { } } }
return xmlreq.} //处理返回信息//xmlHttp返回值,//method:方法
名 方法必须带一个参数如doRecive(xNode).function
handleAjaxResult(req,method) { return function () { if
(req.readyState == 4) { if (req.status == 200) { // 将载有响应信息
的XML传递到处理函数 var objXMLDoc=new
ActiveXObject("Microsoft.XMLDOM").
objXMLDoc.loadXML(req.responseText).
eval("if(objXMLDoc.firstChild)" method
"(objXMLDoc.firstChild.nextSibling)."). } else { //alert("HTTP

```



```
error: " req.status). } } } } //执行客户端Ajax命令 //url 数据post地址 //postData 发送的数据包 //handleMethod 处理返回的方法
function executeAjaxCommand(url,postData,handleMethod){
var req = newXMLHttpRequest(). req.onreadystatechange
=handleAjaxResult(req,handleMethod). req.open("POST", url,
true).
req.setRequestHeader("Content-Type","application/x-www-form-u
rlencoded"). req.setRequestHeader("charset","utf-8").
req.send(postData).} //用户发言
function doSend(){
if(!check())return false. var msg=EditForm.content.value. var
reciver=EditForm.reciver.value. var
url="/chat.ejf?easyJWebCommand=send&lastReadId="
lastReadId. var postData="sender=" myName "amp.content=" msg.
clearTimeout(reciveTime).
executeAjaxCommand(url,postData,"recive").
EditForm.content.value="" } //接收发言信息
function doRecive(){
var reciver=EditForm.reciver.value. var
url="/chat.ejf?easyJWebCommand=recive&lastReadId="
lastReadId. executeAjaxCommand(url,"","recive"). } //处理接收到的
发言信息
function recive(list){ var id="". for(var
oNode=list.firstChild.oNode.oNode=oNode.nextSibling) // 依次
分析每个节点 { chatContent.innerHTML =showMsg(oNode).
id=oNode.getAttribute("cid"). } if(id!="") lastReadId=id.
chatContent.scrollTop=chatContent.scrollHeight.
reciveTime=setTimeout("doRecive()",5000). }
六、系统演示
大家可以到EasyJF开源团队的官方网站看程序演示效果，地址是
```

:

<http://www.easyjf.com/chatRoom.ejf?easyJWebCommand=show&mp.ejid=2538093638804337>结束语 Ajax从技术上讲主要就是javascript、dhtml、css、xmlDOM、xmlhttp等一些我们很早就接触了的技术。而xmlDOM及xmlhttp也没有什么东西，写程序的时候把参考文档打开Copy就OK，dhtml及javascript涉及的东西就多了，不能只是看参考文档，需要把他真正消化，并能灵活动用，这就需要大家都练习了。笔者建议大家不要滥用Ajax。对于高手建议多研究一些业务及系统级算法设计等，对于新手嘛，把基本的技术(客户端的包括dhtml、css、javascript、xml等，J2EE服务器端的设计模式、UML建模、Servlet、JDBC或ORM系统、XML、EJB及一些框架、工具等)学好才是硬道理。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)