# 中间件实现技术开源代码研究报告

**文**

**档**

**说**

**明**

**学 院**： 信息学院

**专 业**： 软件工程

**学 号**： 2017104138

**学 生 姓 名**： 龚志刚

2018年 10月 13日

# 1 绪论

## 1.1课题研究的背景

## 随着计算机技术的飞速发展，各种各样的应用软件需要在各种平台之间进行移植，或者一个平台需要支持多种应用软件和管理多种应用系统，软、硬件平台和应用系统之间需要可靠和高效的数据传递或转换，使系统的协同性得以保证。

## 这些，都需要一种构筑于软、硬件平台之上，同时对更上层的应用软件提供支持的软件系统，而中间件正是在这个环境下应孕而生。

## 1.2课题研究的意义

在互联网时代，企业、政府、学校、医院等各种类型的组织的知识都是以电子文件的形式存在的。这些文件的范围广泛、格式多样，是一个组织极其重要的资产。它们包括数字、文本、影像、网页、音频、视频、代码、等结构化或非结构化数据。大量无序的文件给企业管理带来了很大的负担，严重影响了企业的发展。

所以，企业网络空间和文件夹管理的设计和实现系统，这一类的系统就会越来越重要，因为信息化时代，谁掌握的资源多，谁的信息化成都高，能够快速处理文件信息，谁就会拥有绝对的优势。

在市场竞争日益激烈的今天，信息技术已成为企业赖以提升竞争力的重要手段之一，对提高公司服务质量、提高工作效率、加快信息反馈、增强竞争实力有着重大意义，所以开发一款企业级用的文件管理和分享系统是十分必要的，

## 1.3课题研究的发展

通常意义下，中间件应具有以下的一些特点 ：满足大量应用的需要 ；运行于多种硬件和OS平台 ；支持分布式计算，提供跨网络、硬件和OS平台的透明性的应用或服务的交互功能；支持标准的协议 ；支持标准的接口。

　　程序员通过调用中间件提供的大量API，实现异构环境的通讯，从而屏蔽异构系统中复杂的操作系统和网络协议。

　　中间件提供客户机与服务器之间的连接服务，这些服务具有标准的程序接口和协议。针对不同的操作系统和硬件平台，它们可以有符合接口和协议规范的多种实现。

由于标准接口对于可移植性和标准协议对于互操作性的重要性，中间件已成为许多标准化工作的主要部分。对于应用软件开发，中间件远比操作系统和网络服务更为重要，中间件提供的程序接口定义了一个相对稳定的高层应用环境，不管底层的计算机硬件和系统软件怎样更新换代，只要将中间件升级更新，并保持中间件对外的接口定义不变，应用软件几乎不需任何修改，从而保护了企业在应用软件开发和维护中的重大投资。

# 2 源码分析列表

|  |  |  |
| --- | --- | --- |
| 序号 | 内容 | 技术 |
| 1 | 前端登录页面 | Html css jquery js |
| 2 | 验证码 | 数组，字符截取 |
| 3 | 数据库连接 | JDBC |
| 4 | 文件下载 |  |
| 5 | 登录注册和人员信息 | Html css js java |

# 3 源码分析

**3.1.前端登录页面关键代码**

<script type="text/javascript">

$axure.utils.getTransparentGifPath = function() { return 'resources/images/transparent.gif'; };

$axure.utils.getOtherPath = function() { return 'resources/Other.html'; };

$axure.utils.getReloadPath = function() { return 'resources/reload.html'; };

</script>

<style type="text/css">

#errorUser{

display:none;

color:red;

font-size:15px;

position: relative;

left:850px;

top:230px;

}

#errorPwd{

display:none;

color:red;

font-size:15px;

position: relative;

left:850px;

top:295px;

}

#errorsecurty{

display:none;

color:red;

font-size:15px;

position: relative;

left:940px;

top:320px;

}

</style>

</head>

<body>

<div id="base" class="">

<!-- 登陆后台背景 (形状) -->

<div id="u0" class="ax\_形状">

<img id="u0\_img" class="img " src="images/login/login\_background.png"/>

<!-- Unnamed () -->

<div id="u1" class="text">

<p><span></span></p>

</div>

</div>

<!-- 密码 (文本框) -->

<div id="u2" class="ax\_文本框" title="请输入密码">

<input id="u2\_input" type="password" value="" title="请输入密码"/>

</div>

<span id="errorPwd">测试密码文本。</span>

<!-- 用户名 (文本框) -->

<div id="u3" class="ax\_文本框" title="请输入用户名">

<input id="u3\_input" type="text" value="" title="请输入用户名" onblur="checkUser()"/>

</div>

<span id="errorUser">测试用户名文本。</span>

<!-- 登录按钮 (形状) -->

<div id="u4" class="ax\_形状">

<img id="u4\_img" class="img " src="images/login/login\_button.png"/>

<!-- Unnamed () -->

<div id="u5" class="text">

<p><span>登录</span></p>

</div>

</div>

<!-- 提示 (形状) -->

<div id="u6" class="ax\_文本段落">

<img id="u6\_img" class="img " src="resources/images/transparent.gif"/>

<!-- Unnamed () -->

<div id="u7" class="text">

<p><span style="color:#CCCCCC;">忘记密码？</span><span style="color:#FF0000;">点击这里找回。</span></p>

</div>

</div>

<!-- 用户名文字 (形状) -->

<div id="u8" class="ax\_文本段落">

<img id="u8\_img" class="img " src="resources/images/transparent.gif"/>

<!-- Unnamed () -->

<div id="u9" class="text">

<p><span>用户名</span></p>

</div>

</div>

<!-- 密码文字 (形状) -->

<div id="u10" class="ax\_文本段落">

<img id="u10\_img" class="img " src="resources/images/transparent.gif"/>

<!-- Unnamed () -->

<div id="u11" class="text">

<p><span>密码</span></p>

</div>

</div>

<!-- 注册按钮 (形状) -->

<div id="u12" class="ax\_形状">

<img id="u12\_img" class="img " src="images/login/login\_button.png"/>

<!-- Unnamed () -->

<div id="u13" class="text">

<p><span>注册</span></p>

</div>

</div>

<!-- 验证码输入 (文本框) -->

<div id="u14" class="ax\_文本框" title="请输入验证码">

<input id="u14\_input" type="text" value="" title="请输入验证码"/>

</div>

<span id="errorsecurty">测试验证码。</span>

<!-- 验证码文字 (形状) -->

<div id="u15" class="ax\_文本段落" >

<img id="u15\_img" class="img " src="resources/images/transparent.gif"/>

<!-- Unnamed () -->

<div id="u16" class="text">

<p><span>验证码</span></p>

</div>

</div>

<!-- 验证码的显示 -->

<div id="u17" class="ax\_形状">

<img id="u17\_img" class="img " src="SecurityCodeImageAction" style="cursor:hand;" alt="看不清，换一张"/>

<!-- <div id="u18" class="text">

<p><span>1314</span></p>

</div> -->

</div>

<!-- Unnamed (形状) -->

<div id="u19" class="ax\_形状">

<img id="u19\_img" class="img " src="images/login/u19.png"/>

<!-- Unnamed () -->

<div id="u20" class="text">

<p><span></span></p>

</div>

</div>

<!-- Unnamed (形状) -->

<div id="u21" class="ax\_文本段落">

<img id="u21\_img" class="img " src="resources/images/transparent.gif"/>

<!-- Unnamed () -->

<div id="u22" class="text">

<p><span>软件工程 ©2017 &nbsp; 龚志刚</span></p>

</div>

</div>

<!-- Unnamed (形状) -->

<div id="u23" class="ax\_形状">

<img id="u23\_img" class="img " src="images/login/u23.png"/>

<!-- Unnamed () -->

<div id="u24" class="text">

<p><span></span></p>

</div>

</div>

<!-- Unnamed (形状) -->

<div id="u25" class="ax\_文本段落">

<img id="u25\_img" class="img " src="images/login/u25.png"/>

<!-- Unnamed () -->

<div id="u26" class="text">

<p><span>&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; </span></p>

</div>

</div>

<!-- Unnamed (形状) -->

<div id="u27" class="ax\_文本段落">

<img id="u27\_img" class="img " src="resources/images/transparent.gif"/>

<!-- Unnamed () -->

<div id="u28" class="text">

<p><span>一路有你</span><span></span><span>软件公司欢迎您！</span></p>

</div>

</div>

<!-- Unnamed (形状) -->

<div id="u29" class="ax\_形状">

<img id="u29\_img" class="img " src="images/login/u29.png"/>

<!-- Unnamed () -->

<div id="u30" class="text">

<p><span></span></p>

</div>

</div>

<!-- Unnamed (形状) -->

<div id="u31" class="ax\_形状">

<img id="u31\_img" class="img " src="images/login/u29.png"/>

<!-- Unnamed () -->

<div id="u32" class="text">

<p><span></span></p>

</div>

</div>

</div>

</body>

<script type="text/javascript">

function checkUser(){

var user\_name = $("#u3\_input").val();

if(user\_name == ""){

$("#errorUser").css("display","block");

$("#errorUser").text("用户名不能为空");

//定时执行器

var t=setTimeout("$('#errorUser').css('display','none')",1000)

}else{

$.ajax({

type: "GET",

url: "XEmployee\_findByName.action?employee.name="+user\_name,

/\* data:{mname:mName,

mpass:mPass}, \*/

success: function(data){

if(data=="NO"){

$("#errorUser").css("display","block");

$("#errorUser").text("该用户名不存在");

var t=setTimeout("$('#errorUser').css('display','none')",1000)

}

}

});

}

}

/\* 注意传参的形式。

user.user\_name以及&符号

\*/

//登陆

$("#u4").click(function(){

var user\_name = $("#u3\_input").val();

var user\_pwd = $("#u2\_input").val();

var securityCode = $("#u14\_input").val();

if(user\_name == ""){

alert("用户名不能为空");

}else if(user\_pwd == ""){

alert("密码不能为空");

}else{

window.location.href = "XEmployee\_xLogin.action?employee.name=" + user\_name

+ "&employee.pwd=" + user\_pwd + "&securityCode=" + securityCode;

}

});

//注册

$("#u12").click(function(){

window.location.href = "regist.jsp";

});

//添加时间戳，处理验证码缓存问题

$(function () {

//点击图片更换验证码

$("#u17\_img").click(function(){

$(this).attr("src","SecurityCodeImageAction?timestamp="+new Date().getTime());

});

});

</script>

</html>

**3.2验证码的生成**

public static String getSecurityCode(int length,SecurityCodeLevel level,boolean isCanRepeat){

//随机抽取len个字符

int len=length;

//字符集合(除去易混淆的数字0、数字1、字母l、字母o、字母O)

char[] codes={'1','2','3','4','5','6','7','8','9',

'a','b','c','d','e','f','g','h','i',

'j','k','m','n','p','q','r','s','t',

'u','v','w','x','y','z','A','B','C',

'D','E','F','G','H','I','J','K','L',

'M','N','P','Q','R','S','T','U','V',

'W','X','Y','Z'};

//根据不同的难度截取字符数组

if(level==SecurityCodeLevel.Simple){

codes=Arrays.copyOfRange(codes, 0,9);

}else if(level==SecurityCodeLevel.Medium){

codes=Arrays.copyOfRange(codes, 0,33);

}

//字符集合长度

int n=codes.length;

//抛出运行时异常

if(len>n&&isCanRepeat==false){

throw new RuntimeException(

String.format("调用SecurityCode.getSecurityCode(%1$s,%2$s,%3$s)出现异常，" +

"当isCanRepeat为%3$s时，传入参数%1$s不能大于%4$s",

len,level,isCanRepeat,n));

}

//存放抽取出来的字符

char[] result=new char[len];

//判断能否出现重复的字符

if(isCanRepeat){

for(int i=0;i<result.length;i++){

//索引 0 and n-1

int r=(int)(Math.random()\*n);

//将result中的第i个元素设置为codes[r]存放的数值

result[i]=codes[r];

}

}else{

for(int i=0;i<result.length;i++){

//索引 0 and n-1

int r=(int)(Math.random()\*n);

//将result中的第i个元素设置为codes[r]存放的数值

result[i]=codes[r]

//必须确保不会再次抽取到那个字符，因为所有抽取的字符必须不相同。

//因此，这里用数组中的最后一个字符改写codes[r]，并将n减1

codes[r]=codes[n-1];

n--;

}

}

return String.valueOf(result);

}

}

**3.3数据库连接**

jdbc.user=root

jdbc.password=123456

jdbc.driverClass=com.mysql.jdbc.Driver

jdbc.jdbcUrl=jdbc:mysql:///xjp?useUnicode=true&characterEncoding=UTF-8

jdbc.initialPoolSize=5

jdbc.maxPoolSize=10

**3.4文件下载**

public String export(){

String errorMessage = roleService.permission(session);

if(null == errorMessage){

try {

List<XEmployee> employees = employeeService.XSelect();

ExcelUtil.exportExcel(employees);

errorMessage = "员工列表导出成功！";

} catch (Exception e) {

errorMessage = "员工列表导出失败！";

}

}

session.put("ErrorMessage",errorMessage);

return ERROR;

}

### 3.5登录注册模块和人员信息

public String xLogin(){

String errorMessage = vlidateSecurityCode();

if(null == errorMessage){

employee = employeeService.xLogin(employee);

session.put("employee", loginInit());//文件上传要用

if(null != employee){

return "xLogin";

}else{

errorMessage = "该用户不存在";

}

}

session.put("ErrorMessage",errorMessage);

return ERROR;

}

public String regist(){

String errorMessage = vlidateSecurityCode();

if(null == errorMessage){

if(null == employeeService.findByName(employee.getName())){

employee = registeInit();

if(null != employeeService.save(employee)){

return "Regist";

}

}else{

errorMessage = "该用户已经存在，不允许再次注册";

}

//request.put("user", user);文件可能会用到

session.put("ErrorMessage",errorMessage);

return ERROR;

}

### 员工管理模块

public String XSelectById() {

String errorMessage = roleService.permission(session);

if(null == errorMessage){

XEmployee employee1 = employeeService.XSelectById(employee.getId());

if(employee1 != null){

request.put("employee", employee1);

return "XSelectById";

}else{

errorMessage = "没有找到该员工！";

}

}

session.put("ErrorMessage",errorMessage);

return ERROR;

}

# 4 实验结果

**4.1 前端登录页面**



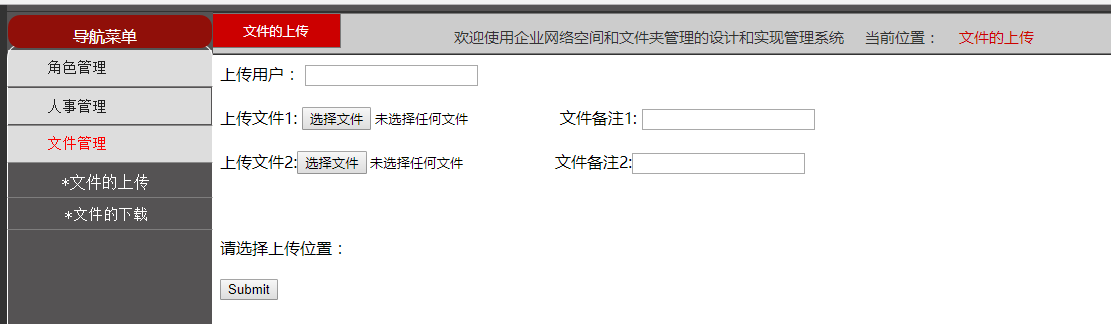
**4.2 验证码**



**4.3数据库连接**



**4.4文件下载**



**4.5登录注册和人员信息**





**5 Durid**

5.1 durid 简介

Druid是Java语言中最好的数据库连接池。Druid能够提供强大的监控和扩展功能。

5.2 Druid 的三个设计原则

　　在设计之初，开发人员确定了三个设计原则（Design Principle）。

（1）快速查询（Fast Query）：部分数据的聚合（Partial Aggregate）+内存化（In-emory）+索引（Index）。

（2）水平扩展能力（Horizontal Scalability）：分布式数据（Distributed Data）+ 并行化查询（Parallelizable Query）。

（3）实时分析（Realtime Analytics）：不可变的过去，只追加的未来（Immutable Past，Append-Only Future）。

5.3 durid 推数

josn文件：3个维度

{

"type": "index\_hadoop",

"spec": {

"ioConfig": {

"type": "hadoop",

"inputSpec": {

"type": "static",

"paths": "FILE\_PATHS"

}

},

"dataSchema": {

"dataSource": "store\_day",

"granularitySpec": {

"type": "uniform",

"segmentGranularity": {

"type": "period",

"period": "P1D",

"timeZone": "GMT+8"

},

"queryGranularity": "none",

"intervals": [

"2017-05-11"

]

},

"parser": {

"type": "hadoopyString",

"parseSpec": {

"format": "csv",

"columns": [

"quantity",

"rate",

"price"

],

"dimensionsSpec": {

"dimensions": [

"code",

"name",

"city",

]

},

"timestampSpec": {

"column": "order\_time"

}

}

},

"metricsSpec": [

{

"name": "count",

"type": "count"

},

{

"name": "quantity",

"type": "longSum",

"fieldName": "quantity"

}

]

},

"tuningConfig": {

"type": "hadoop",

"partitionsSpec": {

"type": "hashed",

"targetPartitionSize": 5000000

},

"jobProperties": {

"mapreduce.job.user.classpath.first": "true"

},

"hadoopDependencyCoordinates": [

"org.apache.hadoop:hadoop-client:2.7.3"

]

}

**6 心得体会**

随着计算机软件技术的发展，中间件技术也已经日渐成熟，并且出现了不同层次、不同类型的中间件产品。按照IDC的分类方法，中间件可分为六类。分别是数据访问中间件、远程过程调用中间件、消息中间件、交易中间件、对象中间件等。

　　数据访问中间件：是为了建立数据应用资源互操作的模式，对异构环境下的数据库实现联接或文件系统实现联接的中间件；

　　远程过程调用中间件：通过这种远程过程调用机制，程序员编写客户方的应用，需要时可以调用位于远端服务器上的过程；

　　消息中间件：用来屏蔽掉各种平台及协议之间的特性，进行相互通信，实现应用程序之间的协同；

　　交易中间件：是在分布、异构环境下提供保证交易完整性和数据完整性的一种环境平台；

　　对象中间件：在分布、异构的网络计算环境中，可以将各种分布对象有机地结合在一起，完成系统的快速集成，实现对象重用。