如果你只是想：部署一个自己的 **blog** 系统 **---** 请直接看第七章

提醒：先浏览下目录会很有帮助。

第二章：概述。
第三章 **--** 第六章：系统具体实现的描述。
第七章：讲解如何将本博客系统部署在 **cloudfoundry** 上。

此博客系统姑且叫：**xblog** 吧**...**

email: work.bfy@qq.com

摘要

将介绍一个用 Java 写的博客系统，从具体实现到最后部署。

系统具有：新用户注册、发表文章、发表评论、修改文章、删除文章、删除评论、搜索文章的功能。

系统界面使用 WordPress 的一个主题。

实现系统时未使用现成的 JavaEE 框架。

实现系统时使用了如下技术：

• jQueryUI 基于 jQuery 的用户界面库（jQuery 是一套 JavaScript 库）。用于，显示“修改密码”“修改别名”的对话框和显示退出信息的对话框。以及一个方便好用的选择表单，用于批量删除文章。

• TinyMCE 一个用 JavaScript 写的在浏览器中运行的所见即所得的 html 编辑器。供用户方便地编辑文章。

• Ajax 此技术使用在：修改密码、修改别名、退出等简单操作中。

系统中的主要 CSS 文件以及用到的多张图片都来自 WordPress 的一个主题。

系统实现后，将其部署在 cloudfoundry.com 上。

关键词：博客、Java、WordPress、TinyMCE、Paas、cloudfoundry.com

目录

第一章 前言

现在有很多公司提供免费的 Paas 服务。

例如：

GAE（Google App Engine）
SAE（Sina App Engine）
CloudFoundry.com
RHCloud.com

我们可以用来运行我们的个人网站。即，把我们的个人站点部署在 Paas 服务提供商的服务器上。形象的称为：部署在“云”端（由于我们见不到部署我们应用的服务器，而且服务器在远方运行，就像它躲在了天边的云里）。

对于大部分个人用户而言，一个博客系统基本就能满足其大部分需求。比较有名的 CMS（Content Management System 内容管理系统）有 WordPress，Drupal（Whitehouse.gov 使用[①]）等。这两个都是用 PHP 写的。

在上述的 4 个 Paas 平台中：RHCloud 和 SAE 支持 PHP，GAE 和 CloudFoundry 现在还不支持。但他们都支持 Java，并且，几乎所有的 Paas 平台都支持 Java。

所以，如果用 Java 实现一个博客系统的功能。在显示层上使用 WordPress 的一个主题。则可以得到一个界面漂亮的博客系统。而且，在部署他时，可以非常自由地选择 Paas 平台。

① http://zh.wikipedia.org/wiki/Drupal

第二章 系统概述

2.1 界面设计：

将直接使用 WordPress 中一个主题的 CSS 文件（做稍微修改）和图片。

• 在 cloudfoundry.com 上部署后的主页截图如下：

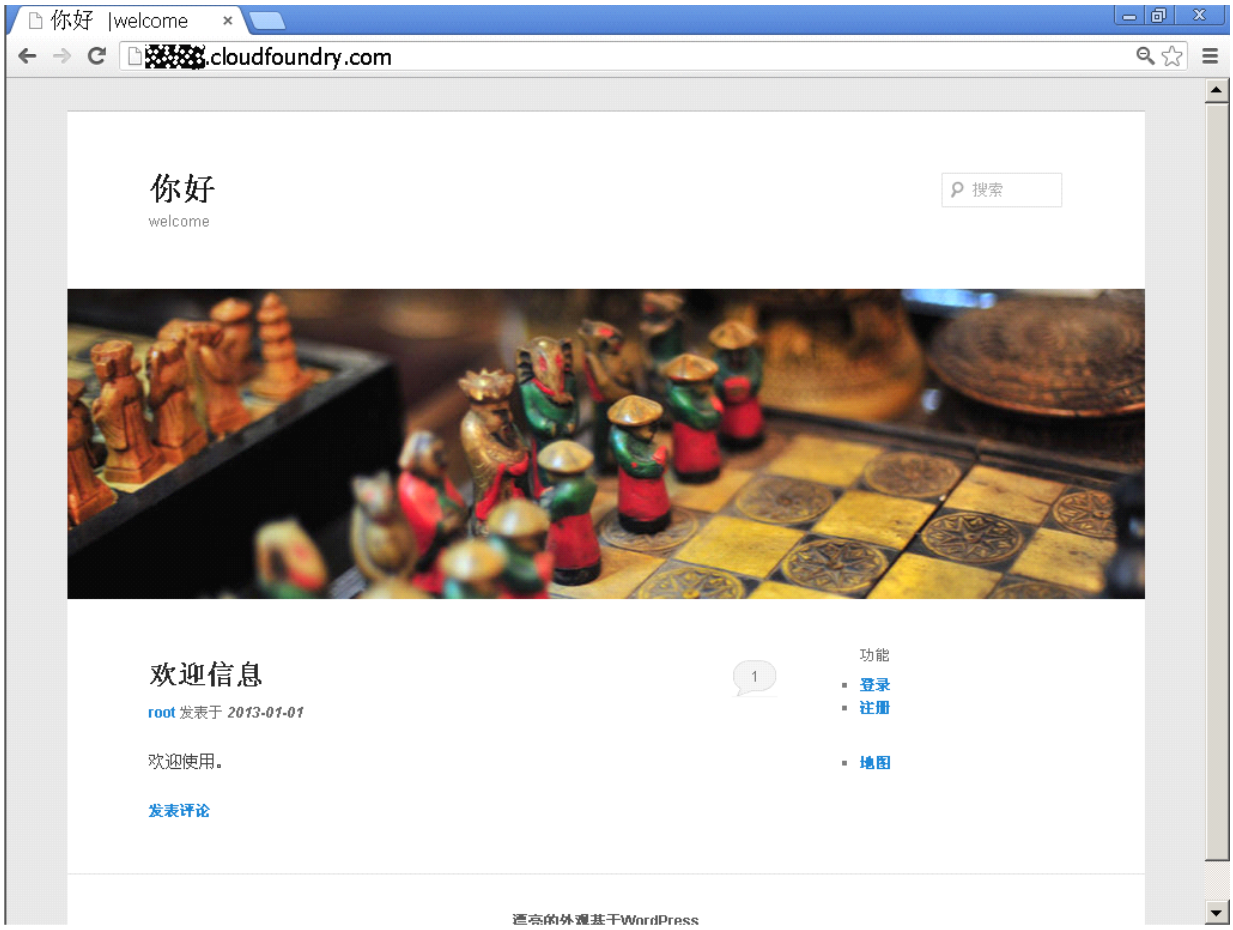

图 2.1

• 使用 TinyMCE 编辑文章的截图如下：

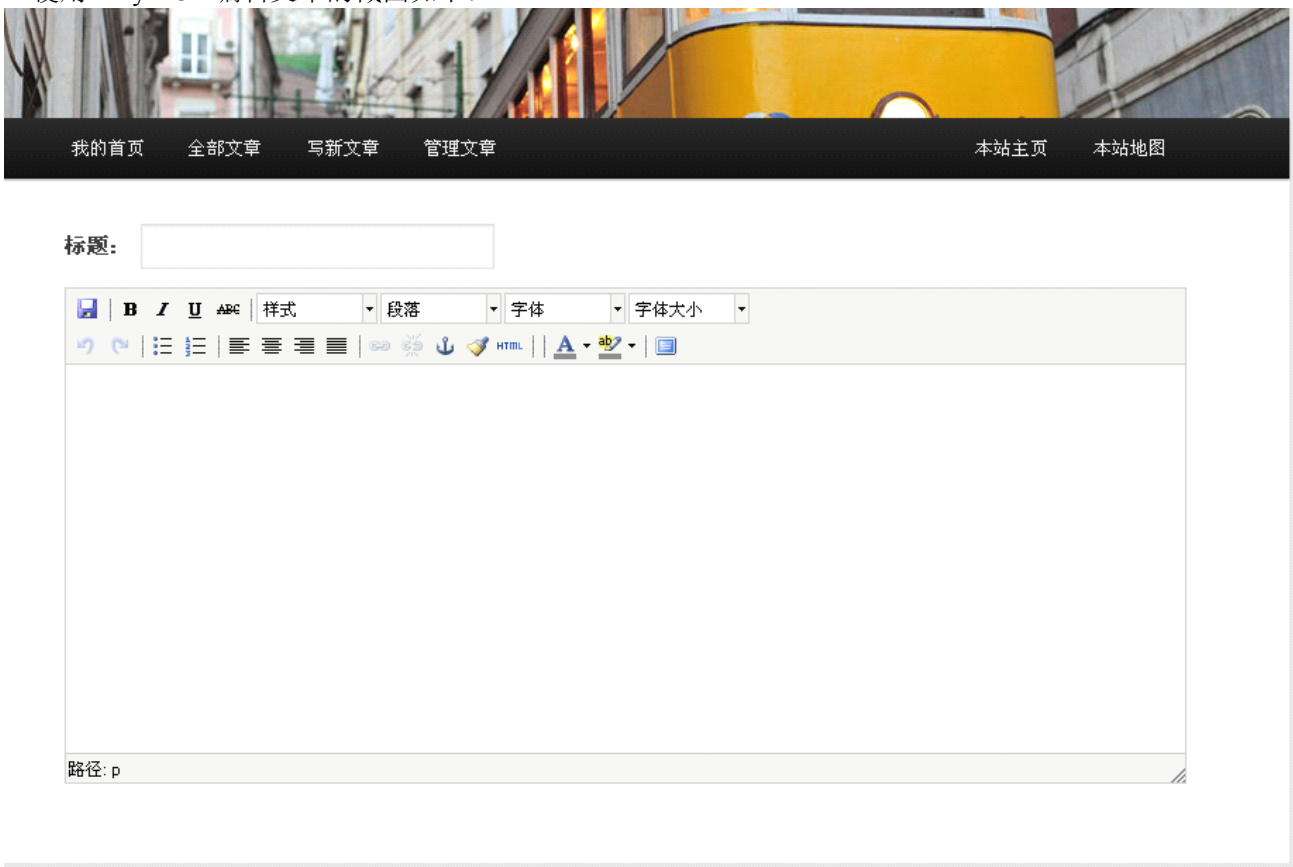

图 2.2

• 按住 Ctrl 键可选择不相邻的多篇文章，一次删除。

图 2.3

• 使用 jQueryUI 的对话框如下：

图 2.4

• 显示评论及发表评论的界面如下:

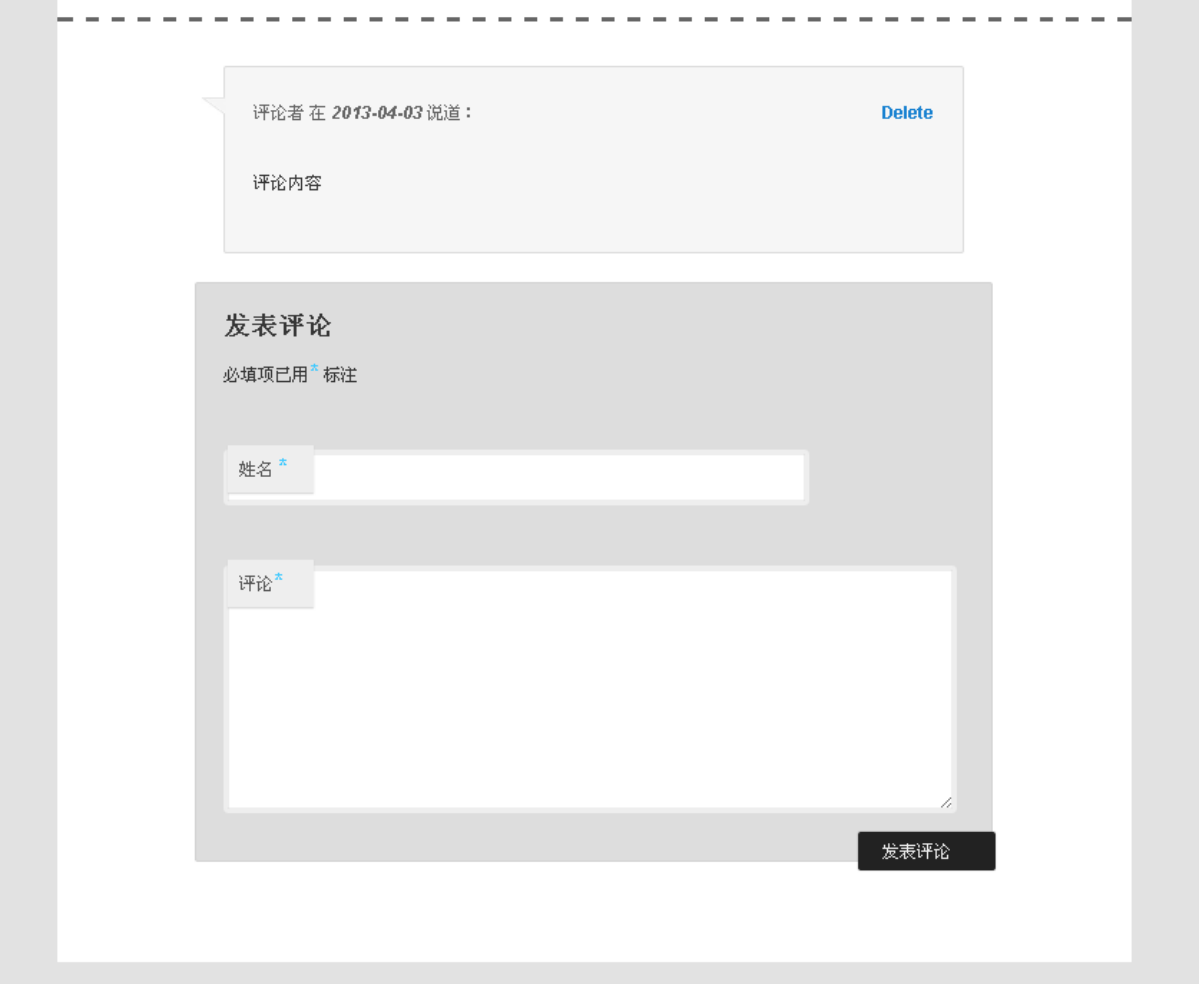

图 2.5

2.2 数据库设计（共有 3 个表）：

2.2.1 users 表

■表 *users* 用来保存用户的相关信息，示意如下：

userid	username	password	alias
1	root	*	root

userid　（用户 id）为此表的主键，数据类型为整型且自动增长。
username（用户名）的数据类型为变长字符串（最长 20）且非空。
password（密码）的数据类型为变长字符串（最长 30）且非空。
alias　（别名）的数据类型为变长字符串（最长 13）且可以为空，缺省值为“你好”。

创建 users 表的 SQL 语句，如下：

```
create table users
(
    userid          int             auto_increment,
    username        varchar(20)     not null,
    password        varchar(30)     not null,
    alias           varchar(13)     default '你好',
    primary key(userid)
)
```

创建表后，添加一个用户：root（管理员）。
用户名为 root，密码为随便输的一个，别名为 root。
初始化 users 表的 SQL 语句，如下：

```
INSERT INTO users (username, password, alias)
VALUES ('root', 'xxxxxxxx','root')
```

2.2.2 articles 表

■表 *articles* 用来保存文章的相关信息，示意如下：

articleid	userid	title	content	xdate
1	1	xxx	xxxxx	2013-01-01

articleid　（文章 id）为此表的主键，数据类型为整型且自动增长。
userid　（文章作者的用户 id）为此表的外键（与 users 表中的 userid）且为级联删除（当删除某用户时，此用户的所有文章也将被删除），数据类型为整型且非空。
title　（文章标题）的数据类型为变长字符串（最长 100）且非空（但可以是长度为 0 的空字符串）。
content　（文章内容）的数据类型为文本型且非空（但可以是长度为 0 的空字符串）。
xdate　（文章日期）的数据类型为日期型且非空。
创建 articles 表的 SQL 语句，如下：

```
create table articles
(
    articleid       int             auto_increment,
    userid          int             not null,
    title           varchar(100)    not null,
    content         text            not null,
    xdate           date            not null,
    primary key(articleid),
```

foreign key(userid) references users(userid) on delete cascade
)

创建表后添加两篇文章。其 userid 为 1 即属于 root 用户。
由于，是在创建表后马上添加的这两篇文章，所以这两篇文章的 articleid 将分别为：1,2。

初始化两篇文章的 SQL 语句如下：

INSERT INTO articles (userid, title, content, xdate)
VALUES (1, '主页介绍', '主页介绍内容', '2013-01-01')

INSERT INTO articles (userid, title, content, xdate)
VALUES (1, '欢迎试用！ ', '欢迎试用内容。', '2013-01-01')

2.2.3 comments 表

■*表 comments* 用来保存文章的相关信息，示意如下：

commentid	articleid	name	info	xdate
1	1	xxx	xxxxx	2013-01-01

commentid（评论 id）为此表的主键，数据类型为整型且自动增长。
articleid （评论所在文章的文章 id）为此表的外键（与 articles 表中的 articleid对应）且为级联删除（当删除某文章时，此文章下的所有评论也将被删除），数据类型为整型且非空。
name （评论者名字）存储发表评论者的姓名，数据类型为变长字符串（最长 29）且非空。
info （评论内容）存储评论的内容，数据类型为文本且非空。
xdate （发表评论的日期）的数据类型为日期型且非空。

创建 comments 表的 SQL 语句，如下：
create table comments
(
commentid int auto_increment,
articleid int not null,
name varchar(29) not null,
info text not null,
xdate date not null,
primary key(commentid),
foreign key(articleid) references articles(articleid) on delete cascade
)

2.3 URL 路径分配：

设，部署的博客系统的 URL 为：http://xblog.x.x 简记为Ω。

Ω/aaa -------用户“aaa”的主页
Ω/bbb -------用户“bbb”的主页
Ω/abcde -------用户“abcde”的主页

Ω/?a=1 -------访问“文章 id”(articleid) 为 1 的文章
Ω/?a=2 -------访问“文章 id”(articleid) 为 2 的文章
Ω/?a=111 -------访问“文章 id”(articleid) 为 111 的文章

Ω /?o=1　　-------请求“操作号”为 1 的操作
Ω /?o=2　　-------请求“操作号”为 2 的操作
Ω /?o=111　-------请求“操作号”为 111 的操作

Ω /aaa.bbb　-------请求根目录下文件名为 aaa.bbb 的文件

2.4 eclipse 中的项目文件分布：

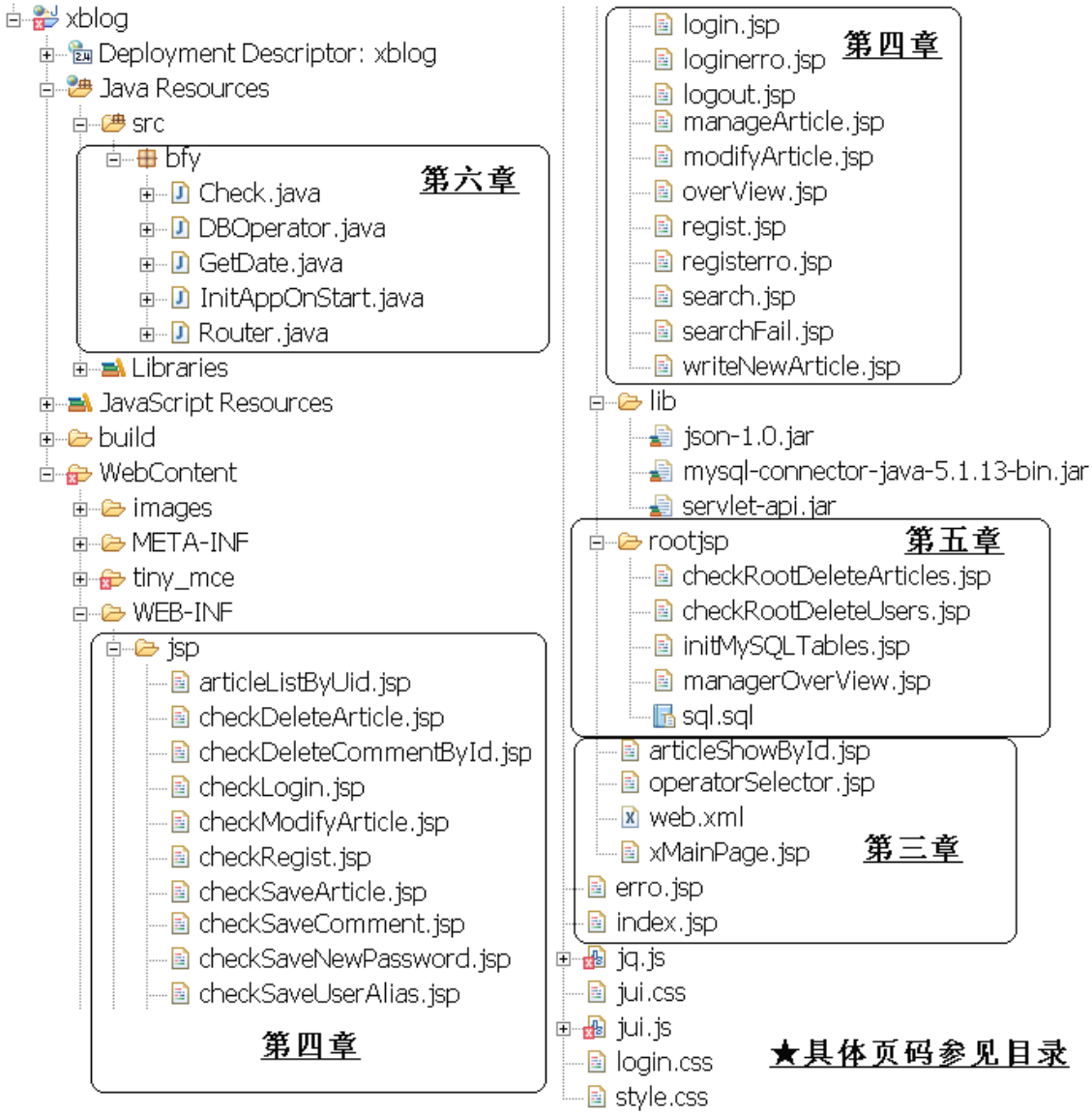

图 2.6

2.5 Router.java(6.5 节) 重定向所有请求的示意图

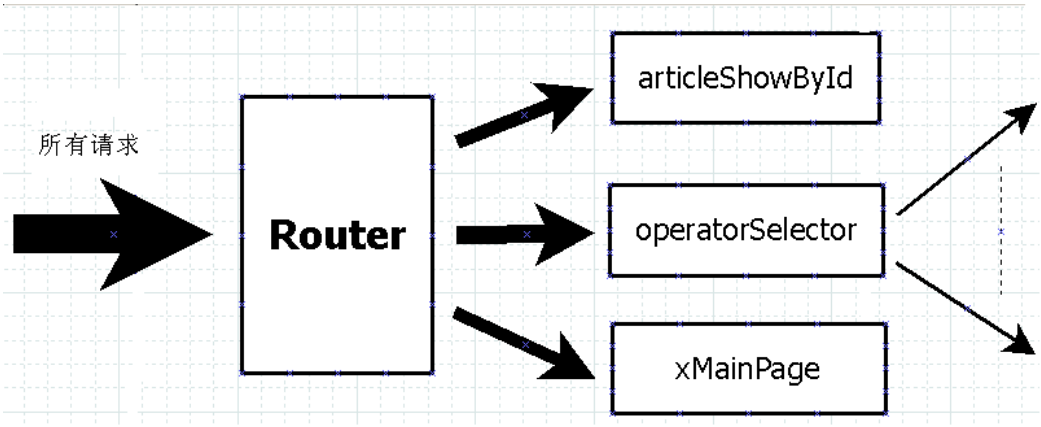

图 2.7

收到的所有请求都会经过 Router 类（继承 filter 类）“路由”。“路由”规则按照“URL 路径分配”（2.3 节）部分所描述的规则进行。但是，各个规则的优先级会有所差别。比如，当同时有 a 和 o 参数时，例如：当访问 http://xblog.x.x/?a=1&o=1 时会转去显示 articleid 为 1 的文章，而不会转去请求 operatorid 为 1 的操作。这是由于处理代码的先后顺序所造成的（见 Router 类源码中“标注 1”下方的代码部分）。

提示：从目录页可以快速查找到要看的源代码。

2.6 web.xml

在 cloudfoundry 上部署应用时（2013 年 1 月）要根据系统的环境变量 VCAP_SERVICES 来获取系统提供的 MySQL 服务的信息。

下面的“installPassword”是在初始化数据库时用来核对用户。

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
                    xmlns="http://java.sun.com/xml/ns/j2ee"
                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
                    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

    <display-name>xblog</display-name>

    <!--设置 cloudfoundry.com 上 MySQL 信息的环境变量名！ -->
    <context-param>
        <param-name>envNameMySQL</param-name>
        <param-value>VCAP_SERVICES</param-value>
    </context-param>

    <!--设置初始化数据库表的密码-->
    <!--或者说是初始化系统的密码-->
    <context-param>
        <param-name>installPassword</param-name>
        <param-value>**********</param-value>
```

```
    </context-param>

    <!-- 设置 Router 类为系统的 filter 用来过滤所有的请求 -->
    <filter>
        <filter-name>router</filter-name>
        <filter-class>bfy.Router</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>router</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>

    <!-- 设置 InitAppOnStart 类为 listener 用来初始化系统的信息-->
    <listener>
        <listener-class>bfy.InitAppOnStart</listener-class>
    </listener>

    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

第三章 WEB-INF 目录下的 jsp

3.1 erro.jsp

此页面用来显示可预知的错误信息。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
                                        isErrorPage="true"  pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>错误页面</title>
</head>
<body>
<div id="main">
    <strong>错误</strong>:
<%
//获取传来的错误信息
String s=(String)request.getAttribute("erroMessage");
if(s==null){
    //如果未设置错误信息，则显示“未知错误!”
    out.write("未知错误! ");
}else{
    //输出错误信息
    out.write(s);
}
%>
</div>
</body>
</html>
```

3.2 index.jsp

此博客应用的主页。这里会显示 articleid 为 1 的文章。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="java.sql.*"%>
<%
//获取 session 范围的 DBOperator 实例，用来进行数据库操作
DBOperator db=(DBOperator)session.getAttribute("db");

//查文章的评论数
String commentCount="0";
ResultSet rs = db.query("select count(*) from comments where articleid=?",1);

if(!rs.next()){
    //数据库正常 这里就不会执行
```

```
    //因为上面查询总会返回0 1 2 3...之类的!
    request.setAttribute("erroMessage", "获取articleid为1的文章评论时出现异常! index.jsp");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
commentCount=rs.getString("count(*)");

//查文章内容
rs = db.query("select* from articles where articleid=?",1);
if(!rs.next()){
    request.setAttribute("erroMessage", "查询不到articleid为1的文章!请初始化数据库。index");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;}
%>
<!DOCTYPE html>
<html lang="zh-CN">

<head>
<title>你好 |welcome</title>
</head>

<body>
    //生成一个随机数，用来选择一个图片显示在页面上
    <img src="./images/<%=(((int)(java.lang.Math.random()*8))+1)%>.jpg"/>

    //输出搜索框
    //例如，搜索“XXX”则其请求为：./?s=XXX&o=18
    //即，s参数传递要搜索的关键字 o参数传递18表示要调用的操作（搜索）
    <form method="get" action="./">
        <input type="text" name="s" id="s" placeholder="搜索" />
        <input type="submit" name="o" value="18" />
    </form>
    </header><!-- #branding -->

        //输出文章标题，并把标题设为指向文章的链接
        <a href="./?a=<%=rs.getString("articleid")%>">
                <%out.println(rs.getString("title"));%>
        </a>

        //输出文章的日期
        <time class="entry-date">
            <%out.println(rs.getString("xdate"));%>
        </time>

        //输出文章上的评论个数 并设为指向评论位置的链接
        <a href="./?a=<%=rs.getString("articleid")%>#comments-title" title="评论">
            <%=commentCount%>
        </a>

        //输出文章的内容
        <%out.println(rs.getString("content"));%>
```

```
        //输出指向发表评论位置的链接
        <a href="./?a=<%=rs.getString("articleid")%>#respond">
            <span   class="leave-reply">发表评论</span>
        </a>

</body>
</html>
```

3.3 articleShowById.jsp

“文章 显示 根据 Id”根据传入的 a 参数（articleid）显示对应的文章内容、评论等信息。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%
//获取请求的 a 参数（articleid）
String aid=(String)request.getParameter("a");
//检查一下传入的参数是否是一个数字
if(!Check.isNum(aid)){
    request.setAttribute("erroMessage","请求文章时使用的文章 id 不正确。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
//根据 articleid 取文章
ResultSet rs = db.query("select * from articles where articleid=?",aid);
if(!rs.next()){
    //文章不存在!
    request.setAttribute("erroMessage","请求的文章不存在！");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//取文章的评论
ResultSet rs_comments= db.query("select * from comments where articleid=?",aid);
//查询用户的信息
ResultSet rs_users = db.query("select * from users where userid=?",rs.getString("userid"));
String userAlias = "";
if(!rs_users.next()){
    //请求的文章存在，但其所有者已不存在
    //发生这种情况的概率极低，即：请求发生在用户刚被删除但其所拥有的文章还没被删除
    //如果级联删除是个原子操作的话，则这种情况不会发生
    request.setAttribute("erroMessage","数据库异常！级联删除异常！aSBI");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//取用户的别名
userAlias = rs_users.getString("alias");
```

```
%>

<!DOCTYPE html>
<html lang="zh-CN">
<head>
//输出文章所有者的别名
<title><%=userAlias%> | welcome</title>
</head>
<body>

    //输出用户的别名，并设为指向用户主页的链接
    <a href="./<%=rs_users.getString("username")%>"><%=userAlias%></a>

    //生成一个随机数，用来选择一个图片显示在页面上
    <img src="./images/<%=(((int)(java.lang.Math.random()*8))+1)%>.jpg" />

    //输出搜索框
    //例如，搜索“XXX”则其请求为：./?s=XXX&o=18
    //即，s 参数传递要搜索的关键字 o 参数传递 18 表示要调用的操作（搜索）
    <form method="get" action="./">
        <input type="text" name="s" id="s" placeholder="搜索" />
        <input type="submit" name="o" value="18" />
    </form>

    //输出作者姓名并设为指向其主页的链接
    <a href="./<%=rs_users.getString("username")%>">作者主页</a>
    //输出用户 id 并设为指向此用户全部文章列表的链接
    <a href="./?o=10&uid=<%=rs.getString("userid")%>">全部文章</a></li>

    //输出文章标题
    <%=rs.getString("title")%>
    //输出文章的日期
    <time><%=rs.getString("xdate")%></time>
    //输出文章的内容
    <%=rs.getString("content")%>


<%  //按时间先后顺序输出此文章的评论
    while(rs_comments.next()){
%>      //输出评论者的姓名
        <%=rs_comments.getString("name")%>
        //输出评论的日期
        <%=rs_comments.getString("xdate")%>
        //输出评论 id 并设为删除此评论的链接
        <a href="./?o=13&p=<%=rs_comments.getString("commentid")%>">Delete</a>
        //输出评论内容
        <%=rs_comments.getString("info")%>
<%
    }
%>
    //输出用于发表评论的表单
```

```
    <form action="./?o=12&p=<%=aid%>" method="post">
        ...
    </form>
</body>
</html>
```

3.4 operatorSelector.jsp

“操作符 选择器”根据请求中的 o 参数（operator id）将请求转发到对应的处理页面。
下面将列出此页面全部代码。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.Check"%>
<%
//获取请求中的 o 参数（operator id）
String oid=(String)request.getParameter("o");
if(!Check.isNum(oid)){
    //如果 oid 不是一个数字，则转错误页面
    request.setAttribute("erroMessage","请求中的 oid 不是一个数字！");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//-----根据 oid 的值 映射到相应的 jsp
switch (Integer.valueOf(oid))
{
    case 1: request.getRequestDispatcher("/WEB-INF/jsp/login.jsp")
                                            .forward(request,response);break;
    case 2: request.getRequestDispatcher("/WEB-INF/jsp/regist.jsp")
                                            .forward(request,response);break;
    case 3: request.getRequestDispatcher("/WEB-INF/jsp/checkLogin.jsp")
                                            .forward(request, response);break;
    case 4: request.getRequestDispatcher("/WEB-INF/jsp/checkRegist.jsp")
                                            .forward(request, response);break;
    case 5: request.getRequestDispatcher("/WEB-INF/jsp/writeNewArticle.jsp")
                                            .forward(request, response);break;
    case 6: request.getRequestDispatcher("/WEB-INF/jsp/checkSaveArticle.jsp")
                                            .forward(request, response);break;
    case 7: request.getRequestDispatcher("/WEB-INF/jsp/manageArticle.jsp")
                                            .forward(request, response);break;
    case 8: request.getRequestDispatcher("/WEB-INF/jsp/checkDeleteArticle.jsp")
                                            .forward(request, response);break;
    case 9: request.getRequestDispatcher("/WEB-INF/jsp/modifyArticle.jsp")
                                            .forward(request, response);break;
    case 10: request.getRequestDispatcher("/WEB-INF/jsp/articleListByUid.jsp")
                                            .forward(request, response);break;
    case 11: request.getRequestDispatcher("/WEB-INF/jsp/checkModifyArticle.jsp")
                                            .forward(request, response);break;
    case 12: request.getRequestDispatcher("/WEB-INF/jsp/checkSaveComment.jsp")
                                            .forward(request, response);break;
    case 13: request.getRequestDispatcher("/WEB-INF/jsp/checkDeleteCommentById.jsp")
```

```
                                        .forward(request, response);break;
    case 14: request.getRequestDispatcher("/WEB-INF/jsp/checkSaveUserAlias.jsp")
                                        .forward(request, response);break;
    case 15: request.getRequestDispatcher("/WEB-INF/jsp/checkSaveNewPassword.jsp")
                                        .forward(request, response);break;
    case 16: request.getRequestDispatcher("/WEB-INF/jsp/logout.jsp")
                                        .forward(request, response);break;
    case 17: request.getRequestDispatcher("/WEB-INF/jsp/overView.jsp")
                                        .forward(request, response);break;
    case 18: request.getRequestDispatcher("/WEB-INF/jsp/search.jsp")
                                        .forward(request, response);break;

    //测试时使用
    //case 99: request.getRequestDispatcher("/WEB-INF/jsp/test.jsp")
                                        .forward(request, response);break;

    case 100: request.getRequestDispatcher("/WEB-INF/rootjsp/initMySQLTables.jsp")
                                        .forward(request, response);break;
    case 101: request.getRequestDispatcher("/WEB-INF/rootjsp/managerOverView.jsp")
                                        .forward(request, response);break;
    case 102:request.getRequestDispatcher("/WEB-INF/rootjsp/checkRootDeleteUsers.jsp")
                                        .forward(request, response);break;
    case 103: request.getRequestDispatcher("/WEB-INF/rootjsp/checkRootDeleteArticles.jsp")
                                        .forward(request, response);break;
    default:
        request.setAttribute("erroMessage", "请求的操作不存在！");
        request.getRequestDispatcher("/erro.jsp").forward(request, response);
        break;
}
return;
%>
```

3.5 xMainPage.jsp

“x 主页”此页面用作已注册用户的个人主页。将显示用户的三篇文章，最近的文章排在上面。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%
//获取请求的用户名参数
String username=(String)request.getAttribute("username");
//检查用户名格式是否正确
if(!Check.isName(username)){
    request.setAttribute("erroMessage","请求用户主页时使用的用户名格式错误。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//将用户名转换为小写格式
String name=username.toLowerCase();
```

```
//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
//取用户信息
ResultSet rs = db.query("select* from users where username=?",name);
if(!rs.next()){
    //如果没有此用户 转错误!
    request.setAttribute("erroMessage","请求的用户主页不存在。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//如果有此用户  则取出其 userid 和别名
String userid=rs.getString("userid");
String userAlias=rs.getString("alias");

//取出此 user 的所有文章
rs = db.query("select* from articles where userid=?",userid);
%>

<!DOCTYPE html>
<html lang="zh-CN">
<head>
    //输出用户别名
    <title><%=userAlias%>| welcome</title>
</head>
<body>
    //将用户别名设为指向其主页的链接
    <a href="./<%=name%>"><%=userAlias%></a>

    //用随机数输出一张图片
    <img src="./images/<%=(((int)(java.lang.Math.random()*8))+1)%>.jpg"/>
    //输出搜索框
    <form method="get" id="searchform" action="./">
        <input type="text" class="field" name="s" id="s" placeholder="搜索" />
        <input type="submit" class="submit" name="o" id="searchsubmit" value="18" />
    </form>
    //输出显示此用户全部文章的链接
    <a href="./?o=10&uid=<%=userid%>">全部文章</a>
    //输出写新文章的链接
    <a href="./?o=5">写新文章</a>
    //输出管理文章的链接
    <a href="./?o=7">管理文章</a>
    //输出一个<li>标签 点击此标签时可以打开一个 jQuery 对话框，通过异步方式完成退出
    <li id="openLogoutInfo"style=" cursor: pointer;float: right;"><a>退出</a></li>
    //输出一个标签 点击此标签时会打开一个 jQuery 对话框，通过异步方式完成修改密码
    <li id="openSetNewPassword"style="cursor: pointer;float: right;"><a>修改密码</a></li>
    //输出一个标签 点击此标签时会打开一个 jQuery 对话框，通过异步方式完成别名设置
    <li id="openSetAlias"style="cursor: pointer;float: right;"><a>设置别名</a></li>

<%
    //输出用户的文章最多三篇，最近的文章排在上面
```

```
    if(!rs.last()){
        //如果此用户还没写文章则输出信息说明..
        out.write("<h1 class=\"entry-title\">还没写文章！</h1>");
    }
    else{
            int i=0;
            do{
%>
                //输出文章标题并设为指向文章的链接
                <a href="./?a=<%=rs.getString("articleid")%>" >
                    <%out.println(rs.getString("title"));%>
                </a>
                //输出文章的时间
                <time>
                    <%out.println(rs.getString("xdate"));%>
                </time>
                //输出此文章上的评论数并设为指向文章评论处的链接
                <a href="./?a=<%=rs.getString("articleid")%>#comments-title">
<%
                    ResultSet comments_rs = db.query("select count(*) from comments where
articleid=?",rs.getString("articleid"));
                    if(!comments_rs.next()){
                        //因为查询结果总会是0,1,2,3，...所以这里不会被执行
                        System.out.println("big erro in xMP");
                    }
%>                  //输出文章上的评论数
                    <%=comments_rs.getString("count(*)")%>
                </a>
                //输出文章的内容
                <%out.println(rs.getString("content"));%>
                //输出指向发表文章评论处的链接
                <a href="./?a=<%=rs.getString("articleid")%>#respond">
                   <span class="leave-reply">发表评论</span>
                </a>
<%
            ++i;
            }while(rs.previous()&&(i<3));
    }
%>
        //输出指向登陆界面的链接
        <li><a href="./?o=1">登录</a></li>
        //输出指向本站主页的链接
        <li><a href="./">本站主页</a></li>
</body>
</html>
```

第四章 WEB-INF/jsp 目录下的 jsp

4.1 articleListByUid.jsp

“文章 列表 根据 Uid”根据请求中的 Uid（userid）参数列出此用户的所有文章，按时间先后顺序排列文章。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%
//获取请求的 uid 值
String uid=(String)request.getParameter("uid");
//如果请求中的 uid 参数不是数字，则转错误页面
if(!Check.isNum(uid)){
    request.setAttribute("erroMessage", "请求的 uid 参数错误！");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
//查询此用户是否存在
ResultSet rs = db.query("select * from users where userid=?",uid);
if(!rs.next()){
    //请求的用户不存在!
    request.setAttribute("erroMessage", "请求的用户文章列表不存在！");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
String name=rs.getString("username"); //获取用户名
String alias=rs.getString("alias"); //获取用户的别名
//查询用户的所有文章
rs = db.query("select * from articles where userid=?",uid);
ResultSet comments_rs;  //声明查询评论的结果集
%>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
//输出用户别名
<title><%=alias%> | welcome</title>
</head>

<body>
    //输出用户别名，并将其设为指向其主页的链接
    <span><a href="./<%=name%>"><%=alias%></a></span>
    //生成一个随机数，用来选择一个图片显示在页面上
    <img src="./images/<%=(((int)(java.lang.Math.random()*8))+1)%>.jpg"/>
    //输出一个搜索框
    <form method="get" id="searchform" action="./">
```

```
        <input type="text" class="field" name="s" id="s" placeholder="搜索" />
        <input type="submit" class="submit" name="o" id="searchsubmit" value="18" />
    </form>

<%
    if(!rs.next()){
        //输出信息说明此用户还没有写文章..
        out.write("<h1 class=\"entry-title\">还没写文章！</h1>");
    }
    else{
            //输出用户的所有文章
            do{
%>
                    //输出文章标题并设为指向文章的链接
                    <a href="./?a=<%=rs.getString("articleid")%>">
                        <%out.println(rs.getString("title"));%>
                    </a>
                    //输出文章日期
                    <%out.println(rs.getString("xdate"));%>
                    //输出此文章上的评论数并设为指向文章评论处的链接
                    <a href="./?a=<%=rs.getString("articleid")%>#comments-title">
<%                  //查询文章上的评论数
                    comments_rs = db.query("select count(*) from comments where
                                    articleid=?",rs.getString("articleid"));
                    if(!comments_rs.next()){
                        //结果应该不会为空  会是 0,1,2,3,4...
                        request.setAttribute("erroMessage", "查询数据库出现逻辑上不可
                                                                能的情况！aLBU");
                        request.getRequestDispatcher("/erro.jsp").forward(request,response);
                        return;
                    }
%>
                    //输出文章上评论的个数
                    <%=comments_rs.getString("count(*)")%>
                    //输出文章的内容
                    <%out.println(rs.getString("content"));%>
                    //输出指向发表文章评论处的链接
                    <a href="./?a=<%=rs.getString("articleid")%>#respond">
                        <span class="leave-reply">发表评论</span>
                    </a>
<%
            }while(rs.next());
    }
%>
            //输出指向某些操作的链接
            <li><a href="./?o=5">写新文章</a></li>
            <li><a href="./?o=7">管理文章</a></li>
            <li><a href="./">本站主页</a></li>
</body>
</html>
```

4.2 checkDeleteArticle.jsp

“核对 删除 文章”核对用户权限后删除文章。下面是全部代码。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%
//获取 session 的 userid 和 username 属性
String userid=(String)session.getAttribute("userid");
String username=(String)session.getAttribute("username");
//如果为 null 或空则转到登陆界面让用户登录
if(userid==null||userid.length()==0){
    request.setAttribute("erroMessage", "请先登录。");
    request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
    return;
}
//获取请求的 p 参数：要删除的文章的 articleid
String s=request.getParameter("p");
if(s==null||s.length()==0){
    request.setAttribute("erroMessage", "请求删除的文章 id 不能为空！");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//以空白符为分隔符将传来的文章 id 字符串，转换成字符数组
String[] articleIds=s.split("\\s");
//核对要删除的文章 id 是否都合法
for (int i=0;i<articleIds.length;i++){
    if(!Check.isNum(articleIds[i])){
        request.setAttribute("erroMessage", "删除文章用的文章 id 错误！");
        request.getRequestDispatcher("/erro.jsp").forward(request, response);
        return;
    }
}
//获取 session 的 db 属性 以进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
//针对要删除的每篇文章核对权限 并进行相应的操作
for(int i=0;i<articleIds.length;i++){
    ResultSet   rs = db.query("select* from articles where articleid=?",articleIds[i]);
        if(!rs.next()){
        //要删除的文章不存在。
        request.setAttribute("erroMessage", "要删除的文章不存在。");
        request.getRequestDispatcher("/erro.jsp").forward(request, response);
        return;
    }
    else{
        //核对登录用户是否为文章所有者
     if((Integer.getInteger(userid))==Integer.getInteger(rs.getString("userid"))){
            //delete!!!!核对权限!!! 执行删除
            db.delete("delete from articles where articleid=?",articleIds[i]);
        }else{
```

```
            request.setAttribute("erroMessage", "登录用户不是文章所有者。");
            request.getRequestDispatcher("/erro.jsp").forward(request, response);
            return;
        }
    }
}

//完成删除后 转用户主页!
request.setAttribute("username",username);
request.getRequestDispatcher("/WEB-INF/xMainPage.jsp").forward(request, response);
return;
%>
```

4.3 checkDeleteCommentById.jsp

“核对 删除 评论 根据 id”核对权限后，删除相应的评论。下面是全部代码。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%
//检查是否已登录
String userid=(String)session.getAttribute("userid");
if(userid==null||userid.length()==0){
    request.setAttribute("erroMessage","请先登录，以取得权限！");
    request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
    return;
}

//获取请求删除的 commentid
String cid=request.getParameter("p");
if(!Check.isNum(cid)){
    //"评论 id"不正确。
    request.setAttribute("erroMessage","要删除的评论 id 不正确。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}

//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
//查询 comment 对应的 articleid
ResultSet rs = db.query("select * from comments where commentid=?",cid);
if(!rs.next()){
    //此"评论 id"不存在。
    request.setAttribute("erroMessage","此 评论 id 不存在。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//获取评论所在文章的 articleid
String aid=rs.getString("articleid");
//根据 aid 查询对应文章的信息
```

```
rs = db.query("select * from articles where articleid=?",aid);
if(!rs.next()){
    request.setAttribute("erroMessage","删除评论时，在文章表中找不到对应的文章。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//获取对应的 article 所有者的 userid
String userid2=rs.getString("userid");
//登录用户的 userid==请求删除的评论的所有者 id
if(userid.equalsIgnoreCase(userid2)){
    //执行删除评论操作
    db.delete("delete from comments where commentid=?",cid);
    //删除完后再转到文章显示页面
    request.getRequestDispatcher("/WEB-INF/articleShowById.jsp?a="+aid)
                                                        .forward(request, response);
    return;
}
//如果请求者是已登录用户
request.setAttribute("erroMessage","登录用户不是评论所在文章的所有者，请以所有者身份登录！");
request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
return;
%>
```

4.4 checkLogin.jsp

“核对 登录”核对传来的用户名、密码后设置登录状态。

下面是全部代码。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="javax.servlet.http.*"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%
//获取传来的用户名、密码
String username=request.getParameter("username");
String password=request.getParameter("password");

//如果用户名密码不符合规定则跳转到 loginerro
if((!Check.isPasswrod(password))||(!Check.isName(username))){
    request.setAttribute("erroMessage","输入不符合规定！");
    request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
    return;
}

//设置 cookie
String rem = request.getParameter("rememberme");
if(rem!= null && rem.equalsIgnoreCase("1")){
    Cookie cookie = new Cookie("xblogUsername",username);
    cookie.setMaxAge(365*24*3600);
    response.addCookie(cookie);
}
```

```
//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
//根据传来的用户名查询数据库
ResultSet rs = db.query("select * from users where username = ?", username);
if (rs.next())
{
    //用户名和密码匹配
    if (rs.getString("password").equals(password))
    {
        //设置 session 的 userid username 属性  表示用户已登录!
        session.setAttribute("userid",rs.getString("userid"));
        session.setAttribute("username", rs.getString("username"));
        request.setAttribute("username",username);

        //转用户的主页
        request.getRequestDispatcher("/WEB-INF/xMainPage.jsp").forward(request, response);
        return;
    }
    else
    {
        request.setAttribute("erroMessage","密码错误! ");
        request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
        return;
    }
}
else
{
    //用户不存在!
    request.setAttribute("erroMessage","用户名不存在。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
}
%>
```

4.5 checkModifyArticle.jsp

“核对 修改 文章”核对用户权限后把修改后的文章写入数据库。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.GetDate"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%@page import="javax.servlet.http.*"%>
<%
//获取 session 的 userid 和 username 属性
String username=(String)session.getAttribute("username");
String userid  =(String)session.getAttribute("userid");
if(userid==null||userid.length()==0){
    request.setAttribute("erroMessage","尚未登录，请先登录! ");
    request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
    return;
```

```
}
//获取请求参数中的文章 id（要修改的文章的 id）
String aid=(String)request.getParameter("p");
if(!Check.isNum(aid)){
    request.setAttribute("erroMessage","要修改的文章 id 错误！");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}

//获取文章标题 并重新解码
String t = request.getParameter("title");
if(t==null){t="";}
byte[] b=t.getBytes("ISO-8859-1");
String title=new String(b,"UTF-8");
//获取文章内容并重新解码
String c=request.getParameter("content");
if(c==null){c="";}
b=c.getBytes("ISO-8859-1");
String content=new String(b,"UTF-8");

//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");

//查询 article 所有者的 userid
ResultSet rs = db.query("select* from articles where articleid=?",aid);
if(!rs.next()){
    request.setAttribute("erroMessage", "要修改的文章不存在。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//核对登录用户是否为文章所有者
if((Integer.getInteger(userid))==Integer.getInteger(rs.getString("userid"))){
    //将修改写入数据库
    db.modify("update articles set title=?,content=?,xdate=? where articleid=?",
                                                    title,content,GetDate.get(),aid);
    request.setAttribute("username",username);
    request.getRequestDispatcher("/WEB-INF/xMainPage.jsp").forward(request, response);
    return;
}
else{
    request.setAttribute("erroMessage", "现在登录的账号不是文章的所有者！");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
%>
```

4.6 checkRegist.jsp

“核对 注册”核对用户的注册信息，完成用户注册。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.GetDate"%>
```

```
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%@page import="javax.servlet.http.*"%>

<%
//获取传来的的注册信息（用户名、密码）
String username=request.getParameter("username");
String password=request.getParameter("password");

//如果注册信息不符合规定 则跳转到 loginerro
if((!Check.isPasswrod(password))||(!Check.isName(username))){
    request.setAttribute("erroMessage","输入信息不符合规定！");
    request.getRequestDispatcher("/WEB-INF/jsp/registerro.jsp").forward(request, response);
    return;
}

//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
ResultSet rs = db.query("select username from users where username = ?", username);
if (rs.next())
{
        request.setAttribute("erroMessage","用户名已存在");
        request.getRequestDispatcher("/WEB-INF/jsp/registerro.jsp")
                                                        .forward(request, response);
        return;
}
else
{
    //将信息插入数据库!
    if(!db.insert("INSERT INTO users (username, password)VALUES (?, ?)", username,password))
    {
        request.setAttribute("erroMessage", "将新用户插入数据库时出现错误! in cR");
        request.getRequestDispatcher("/erro.jsp").forward(request, response);
        return;

    }

    //!!!! 把新注册用户的第一篇文章设为欢迎信息和操作指南!!
    //欢迎信息和操作指南---是 root 的 articleid=2 的一篇文章!!
    //把文章复制给新用户
    String title="";
    String content="";
    String userid ="";
    rs = db.query("select* from articles where articleid=?",2);
    if(rs.next()){
        title=rs.getString("title");
        content=rs.getString("content");
    }
    rs = db.query("select* from users where username=?",username);
    if(rs.next()){userid=rs.getString("userid");}
```

```
    db.insert("INSERT INTO articles (userid, title, content, xdate) VALUES
                                            (?, ?, ?, ?)",userid,title,content,GetDate.get());
    //注册成功 转用户主页!
    //设置 session 属性表示此用户已登录
    session.setAttribute("username",username);
    session.setAttribute("userid",userid);
    //设置 request 的 username 属性用来访问用户的主页
    request.setAttribute("username",username);
    request.getRequestDispatcher("/WEB-INF/xMainPage.jsp").forward(request,response);
}
%>
```

4.7 checkSaveArticle.jsp

“核对 保存 文章”核对用户权限后，将新文章写入数据库

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.GetDate"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%@page import="javax.servlet.http.*"%>
<%
//获取 session 的 userid 和 username 属性，用来检查用户是否登录
String username=(String)session.getAttribute("username");
String userid=(String)session.getAttribute("userid");
if(userid==null||userid.length()==0){
    request.setAttribute("erroMessage","尚未登录！");
    request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
    return;
}
//将收到的字符串重新解、编码
String t = request.getParameter("title");
if(t==null){t="";}
byte[] b = t.getBytes("ISO-8859-1");
String title=new String(b,"UTF-8");

String c=request.getParameter("content");
if(c==null){c="";}
b=c.getBytes("ISO-8859-1");
String content=new String(b,"UTF-8");
//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
//将新文章写入数据库
if(db.insert("INSERT INTO articles (userid, title, content, xdate) VALUES
                                            (?,?,?,?)",userid,title,content,GetDate.get()))
{
    //写入成功后转用户首页!
    request.setAttribute("username",username);
    request.getRequestDispatcher("/WEB-INF/xMainPage.jsp").forward(request, response);
    return;
}
```

```
else{
    //发生插入 erro 不应该发生此种错误!!
    request.setAttribute("erroMessage","将新文章插入数据库时发生错误。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
%>
```

4.8 checkSaveComment.jsp

“核对 保存 评论”核对用户输入的评论信息后，将评论写入数据库。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.GetDate"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%@page import="javax.servlet.http.*"%>

<%
//获取 评论者的姓名 评论信息 评论所在的文章
String name=request.getParameter("name");
String info=request.getParameter("info");
String aid =request.getParameter("p");

if(name==null||info==null||name.length()<2||name.length()>37||info.length()>571){
    request.setAttribute("erroMessage", "输入的信息不符合规定。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//检查传来的文章参数是否是数字
if(!Check.isNum(aid)){
    request.setAttribute("erroMessage", "请求存储评论时，评论所在的文章参数不正确。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db = (DBOperator)session.getAttribute("db");
ResultSet  rs = db.query("select * from articles where articleid=?",aid);

if(!rs.next()){
    request.setAttribute("erroMessage","请求存储评论时，评论所在的文章在数据库中找不到！");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//重新编解码
byte[] b=name.getBytes("ISO-8859-1");
String xname=new String(b,"UTF-8");
byte[] c=info.getBytes("ISO-8859-1");
String xinfo=new String(c,"UTF-8");
//将评论插入数据库
if(!db.insert("INSERT    INTO    comments    (    articleid,    name,    info,    xdate)    VALUES
```

```
(?,?,?,?)",aid,xname,xinfo,GetDate.get())){
    request.setAttribute("erroMessage", "将评论插入数据库时 失败。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//插入评论成功后 转到评论所在文章的页面
request.getRequestDispatcher("/WEB-INF/articleShowById.jsp?a="+aid)
                                                        .forward(request, response);
%>
```

4.9 checkSaveNewPassword.jsp

“核对 保存 新 密码”核对用户权限后保存用户的新密码。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%@page import="javax.servlet.http.*"%>

<%
//获取 session 的 userid 属性 用来检查用户是否登录
String userid=(String)session.getAttribute("userid");
if(userid==null||userid.length()==0){
    //由于修改密码用的是 Ajax 所以直接 out.write...
    out.write("请登录后修改！");
    return;
}

//获得传来的新密码
String newPassword=request.getParameter("newPassword");
//检查新密码的格式
if(!Check.isPasswrod(newPassword)){
    out.write("格式错误！");
    return;
}

//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db = (DBOperator)session.getAttribute("db");

//将用户的新密码写入数据库
db.modify("update users set password=? where userid=?",newPassword,userid);
out.write("修改成功！");
%>
```

4.10 checkSaveUserAlias.jsp

“核对 保存 用户 别名” 核对权限后保存用户的别名。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
```

```
<%@page import="javax.servlet.http.*"%>
<%
//获取 session 的 userid 属性 用来检查用户是否登录
String userid=(String)session.getAttribute("userid");
if(userid==null||userid.length()==0){
    out.write("请登录后修改！");
    return;
}
//获取用户输入的新别名
String s=request.getParameter("aliasname");
//检查别名的格式是否正确
if(s==null||s.length()<2||s.length()>11){
    out.write("格式错误！");
    return;
}

//解决编码问题
byte[] b = s.getBytes("ISO-8859-1");
String newAlias=new String(b,"UTF-8");

//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db = (DBOperator)session.getAttribute("db");
//修改用户的别名
db.modify("update users set alias=? where userid=?",newAlias,userid);
//直接通过 Ajax 输出信息
out.write("修改成功！");
%>
```

4.11 login.jsp

“登录”提供登录表单供用户使用。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%
//提取 Cookie 的用户名 用来设置登录表单
String rememberName = "";
Cookie[] cookies = request.getCookies();
for (Cookie c:cookies){
    if(c.getName().equals("xblogUsername")){
        rememberName=c.getValue();
    }
}
%>
<!DOCTYPE html>
<html>
<head>
    <title>你好 › 登录</title>
</head>
<body>
    //输出指向首页的链接
    <a href="./" title="基于 WordPress">你好</a>
```

```
    //输出供用户输入的表单
    <form name="loginform" id="loginform" action="./?o=3" method="post">
        ......
    </form>

    //输出指向首页的链接
    <a href="./" title="不知道自己在哪？">← 回到 主页</a>

</body>
</html>
```

4.12 loginerro.jsp

“登录 错误”不仅提供表单供用户使用，而且能显示错误信息。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@ page language="java" contentType="text/html; charset=UTf-8" pageEncoding="UTF-8"%>
<%
//提取 Cookie 的用户名 用来设置登录表单
String rememberName = "";
Cookie[] cookies = request.getCookies();
for (Cookie c:cookies){
    if(c.getName().equals("xblogUsername")){
        rememberName=c.getValue();
    }
}
%>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <title>你好 ›登录</title>
</head>
<body>
    //输出指向主页的链接
    <a href="./" title="基于 WordPress">你好</a>
    //输出错误信息
    <strong>信息</strong>：<%=request.getAttribute("erroMessage")%>
    //输出供用户输入的表单
    <form name="loginform" id="loginform" action="./?o=3" method="post">
        •••
    </form>
    //输出指向主页的链接
    <a href="./" title="不知道自己在哪？">← 回到 主页</a>
</body>
</html>
```

4.13 logout.jsp

“退出” 将 session 的 username 和 userid 属性设为空。表示已退出。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%
```

```
session.setAttribute("username",null);
session.setAttribute("userid",null);
//通过 Ajax 输出信息
out.write("已成功退出。");
%>
```

4.14 manageArticle.jsp

“管理 文章”提供用户文章的列表，供用户选择进行修改或删除操作。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%
//获取 session 的 userid、username 属性 用来检查用户是否登录
String userid=(String)session.getAttribute("userid");
String username=(String)session.getAttribute("username");

if(userid==null||userid.length()==0){
    //若尚未登录则要求用户先登录
    request.setAttribute("erroMessage", "请登录！");
    request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
    return;
}
//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
//查询此用户的所有文章
ResultSet rs = db.query("select* from articles where userid=?",userid);
%>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <title>管理文章</title>
</head>
<body>
    //生成一个随机数，用来选择一个图片显示在页面上
    <img src="./images/<%=(((int)(java.lang.Math.random()*8))+1)%>.jpg"/>
    //输出搜索表单
    <form method="get" id="searchform" action="./">
        <input type="text" class="field" name="s" id="s" placeholder="搜索" />
        <input type="submit" class="submit" name="o" id="searchsubmit" value="18" />
    </form>

<%
    if(!rs.next()){
        //输出信息说明此用户没有..
        out.write("<strong>还没写文章！</strong>");
    }else{
%>          //输出所有文章的列表
<%          do{
```

```
%>
                <li lang="<%=rs.getString("articleid")%>">
                                                    <%=rs.getString("title")%></li>
<%          }while(rs.next());
%>
<%
    }
%>
<%
    if(!rs.first()){
        //输出信息说明此用户没有..
        out.write("<strong>还没写文章！</strong>");
    }else {
%>          //输出所有文章的列表
            //跟上面输出文章列表的代码是一样的
            //不同的是：这两次输出一个用于“修改文章”功能 另一个用于“删除文章”
<%          do{
%>
                <li lang="<%=rs.getString("articleid")%>"><%=rs.getString("title")%></li>
<%
             }while(rs.next());
%>
<%
    }
%>
        //输出指向用户主页的链接
        <li><a href="./<%=username%>">我的主页</a></li>
        //输出指向用户全部文章的链接
        <li><a href="./?o=10&uid=<%=userid%>">全部文章</a></li>
        //输出指向写新文章页面的链接
        <li><a href="./?o=5">写新文章</a></li>
        //输出指向管理文章页面的链接
        <li><a href="./?o=7">管理文章</a></li>
        //输出指向本站主页的链接
        <li><a href="./">本站主页</a></li>
</body>
</html>
```

4.15 modifyArticle.jsp

“修改 文章”此页面用于将用户要修改的文章显示出来供用户修改。然后，将修改后的文章提交。下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%
//获得 session 范围的 userid、username 属性，用来检查用户是否登录
String userid=(String)session.getAttribute("userid");
String username=(String)session.getAttribute("username");
if(userid==null||userid.length()==0){
```

```
    request.setAttribute("erroMessage", "未登录，无法修改文章！请先登录。");
    request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
    return;
}

//获取请求的 p 参数（要修改的文章的 articleid）
String s=request.getParameter("p");
if(s==null||s.length()==0){
    request.setAttribute("erroMessage", "修改文章的请求中文章 id 不正确。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//将传来的请求参数以空白符为分隔符转换成字符数组
String[] articleIds=s.split("\\s");
//只处理字符数组的第一个元素
//因为 一次只能修改一篇文章
if(!Check.isNum(articleIds[0])){
    request.setAttribute("erroMessage", "修改文章的请求中文章 id 不正确。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}

//获取 session 的 db 属性 以进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
//根据 articleid 查询数据库
ResultSet rs = db.query("select* from articles where articleid=?",articleIds[0]);
if(!rs.next()){
    request.setAttribute("erroMessage", "要修改的文章不存在。");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
%>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <title>修改文章</title>
</head>
<body>
    //生成一个随机数，用来选择一个图片显示在页面上
    <img src="./images/<%=(((int)(java.lang.Math.random()*8))+1)%>.jpg"/>
    //输出用于搜索的表单
    <form method="get" id="searchform" action="./">
        ...
    </form>
    //输出文章的标题
    <input value="<%=rs.getString("title")%>"/>
    //输出文章内容
    <%=rs.getString("content")%>
    //输出显示用户全部文章的链接
    <a href="./?o=10&uid=<%=userid%>">全部文章</a>
    //输出指向用户主页的链接
```

```
    <a href="./<%=username%>">我的主页</a>
    //输出几个功能链接
    <a href="./?o=5">写新文章</a>
    <a href="./?o=7">管理文章</a>
    <a href="./">站首页</a>
</body>
</html>
```

4.16 overView.jsp

“概览”显示本站的所有用户、文章。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%@page import="javax.servlet.http.*"%>
<%
//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
//查询 user 表的必要内容，供显示使用
ResultSet rs_users= db.query("select userid,username,alias from users");
//查询文章表的内容，供显示使用
ResultSet rs_articles= db.query("select articleid,userid,title,xdate from articles");
%>
<!DOCTYPE html>
<html>
<head>
    <title>地图</title>
</head>
<body>
    //输出 users 表的部分内容
    <h2>用户列表</h2>
    <table><tbody>
        //输出表的第一行 标题行
        <tr>
            <td width="51">序号</td>
            <td width="121">用户 ID</td>
            <td width="211">用户名</td>
            <td width="211">用户别名</td>
        </tr>
<%       //循环输出信息
        for(int i=1;rs_users.next();i++){
%>
            //将输出的行交替设为 a1、a0 类
            //目的是用 CSS 赋予他们不同的样式
            <tr class="a<%=i%2%>">
                //输出序号（用来计数：1、2、3、4、5、•••）
                <td><%=i%></td>
                //输出用户 ID
                <td><%=rs_users.getString("userid")%></td>
```

```
                //输出用户名并设为指向用户主页的链接
                <td><a
href="./<%=rs_users.getString("username")%>"><%=rs_users.getString("username")%></a></td>
                //输出用户的别名
                <td><%=rs_users.getString("alias")%></td>
            </tr>
        <%
        }
        %>
    </tbody></table>

    //输出 articles 表的部分内容
    <h2>文章列表</h2>
    <table><tbody>
       //输出表的第一行 标题行
        <tr>
            <td width="51">序号</td>
            <td width="113">文章 ID</td>
            <td width="113">作者 ID</td>
            <td width="313">标题</td>
            <td width="171">日期</td>
        </tr>
<%
         //循环输出信息
        for(int i=1;rs_articles.next();i++){
%>
            //将输出的行交替设为 a1、a0 类
            //目的是用 CSS 赋予他们不同的样式
            <tr class="a<%=i%2%>">
                //输出序号（用来计数：1、2、3、4、5、•••）
                <td><%=i%></td>
                //输出文章 ID
                <td><%=rs_articles.getString("articleid")%></td>
                //输出文章作者的 ID
                <td><%=rs_articles.getString("userid")%></td>
                //输出文章的标题 并设为指向文章的链接
                <td>
                    <a href="./?a=<%=rs_articles.getString("articleid")%>">
                        <%=rs_articles.getString("title")%>
                    </a>
                </td>
                //输出文章的日期
                <td><%=rs_articles.getString("xdate")%></td>
            </tr>
<%
        }
%>
    </tbody></table>
</body>
</html>
```

4.17 regist.jsp

“注册”提供一个用于输入注册信息的表单，供用户注册使用。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>你好 › 注册</title>
</head>
<body>
    //输出供输入注册信息的表单
    <form action="./?o=4" method="post">
        •••
    </form>
    //输出回到主页的链接
    <a href="./" title="不知道自己在哪？">← 回到 主页</a>
</body>
</html>
```

4.18 registerro.jsp

“注册 错误”不仅提供表单供用户使用，而且能显示错误信息。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@ page language="java" contentType="text/html; charset=UTf-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>你好 › 注册</title>
</head>
<body>
    //输出错误信息
    <strong>错误</strong>：<%=request.getAttribute("erroMessage")%>
    //输出供输入注册信息的表单
    <form name="loginform" id="loginform" action="./?o=4" method="post">
        •••
    </form>
    //输出回到主页的链接
    <a href="./" title="不知道自己在哪？">← 回到 主页</a>
</body>
</html>
```

4.19 search.jsp

“搜索” 根据用户提供的关键词，从数据库中搜索，并把结果显示出来。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="java.sql.ResultSet"%>
<%
//获取请求查询的关键词串
String t = request.getParameter("s");
```

```
//如果查询的关键词为空则转到 searchFail.jsp
if(t==null||t.length()==0){
    request.getRequestDispatcher("/WEB-INF/jsp/searchFail.jsp").forward(request, response);
    return;
}
//重新解码
String s=new String(t.getBytes("ISO-8859-1"),"UTF-8");
//将关键词串分割为关键词列表
String[] keywords=s.split("\\s");
//如果第一个关键词过短或过长则转到 searchFail.jsp
if((keywords.length==0)||(keywords[0].length()<1)||(keywords[0].length()>47)){
    request.getRequestDispatcher("/WEB-INF/jsp/searchFail.jsp").forward(request, response);
    return;
}

//用第一个关键词查数据库
String keyword=keywords[0];

//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (bfy.DBOperator)session.getAttribute("db");
//查询文章列表
ResultSet rs_articles = db.query("select * from articles");
//定义了一个标志（flag）用来表示是否有结果可显示
int rsFlag=0;
%>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <title>搜索结果</title>
</head>
<body>
    //生成一个随机数，用来选择一个图片显示在页面上
    <img src="./images/<%=(((int)(java.lang.Math.random()*8))+1)%>.jpg"/>
    //输出搜索表单
    <form method="get" id="searchform" action="./">
        <input type="text" class="field" name="s" id="s" placeholder="搜索" />
        <input type="submit" class="submit" name="o" id="searchsubmit" value="18" />
    </form>
<%
    //如果用户已登录则在搜索页面输出回用户主页的链接
    String u = (String)session.getAttribute("username");
    if(u!=null){
%>
        <li><a href="./<%=u%>">我的主页</a></li>
<%
    }
%>
        <li><a href="./">本站主页</a></li>
        <li><a href="./?o=17">本站地图</a></li>
<%
//遍历所有文章
```

```
while(rs_articles.next()){
    //如果文章的标题或内容包含查询的关键字，则将文章作为结果输出
    if(rs_articles.getString("title").contains(keyword)||rs_articles.getString("content").con
tains(keyword)){
        //将标志位设为 1，即页面将显示结果而不是显示“查询不到•••”
        rsFlag=1;
%>
        //输出文章标题并设为指向文章的链接
        <a href="./?a=<%=rs_articles.getString("articleid")%>">
            <%out.println(rs_articles.getString("title"));%>
        </a>
        //输出文章的日期
        <time class="entry-date">
            <%out.println(rs_articles.getString("xdate"));%>
        </time>

        //查询文章上的评论数 并输出 并设为指向评论处的链接
        <a href="./?a=<%=rs_articles.getString("articleid")%>#comments-title">
            <%ResultSet comments_rs = db.query("select count(*) from comments where
                                articleid=?",rs_articles.getString("articleid"));
                    if(!comments_rs.next()){
                        //结果应该不会为空  会是 0,1,2,3,4...
                        request.setAttribute("erroMessage", "查询数据库出现逻辑上不可能的情况!
aLBU");
                        request.getRequestDispatcher("/erro.jsp").forward(request, response);
                        return;
                    } %>
                    //输出评论的个数
                    <%=comments_rs.getString("count(*)")%>
        </a>
        <p>
            //输出文章的内容
            <%out.println(rs_articles.getString("content"));%>
        </p>
        //输出指向 发表文章评论处的链接
        <a href="./?a=<%=rs_articles.getString("articleid")%>#respond">
            <span class="leave-reply">发表评论</span>
        </a>
<%
    }
}
%>
<%
    //如果 rsFlag==0 则说明前面没有匹配的结果
    //所以下面会输出信息：未找到
    if(rsFlag==0){
%>
        //输出搜索表单
        <form method="get" id="searchform" action="./">
            •••
        </form>
```

```
<%
    }
%>
</body>
</html>
```

4.20 searchFail.jsp

“搜索 失败”当搜索使用的关键字过短、过长等情况时，会显示此页面。

此页面将显示“未找到”信息，并提供表单供用户再次查询。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>搜索结果</title>
</head>
<body>
    //生成一个随机数，用来选择一个图片显示在页面上
    <img src="./images/<%=(((int)(java.lang.Math.random()*8))+1)%>.jpg"/>
    //输出搜索表单
    <form method="get" id="searchform" action="./">
        …
    </form>
    //输出搜索表单
    <form method="get" id="searchform" action="./">
        …
    </form>
</body>
</html>
```

4.21 writeNewArticle.jsp

“写 新 文章”此页面提供了一个以 tinyMCE 为编辑器的页面，供用户写文章使用。关于 tinyMCE 的具体使用见附录部分。下面代码中将省略有关 tinyMCE 的代码。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%@page import="javax.servlet.http.*"%>
<%
//获取 session 的 userid 和 username 属性,用来检查用户是否登录
String userid=(String)session.getAttribute("userid");
String username=(String)session.getAttribute("username");

if(userid==null||userid.length()==0){
    //尚未登录则转登录页面
    request.setAttribute("erroMessage","尚未登录！");
    request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
    return;
```

```
}
%>
<!DOCTYPE html>
<html>
<head>
    <title>写新文章</title>
</head>
<body>
    //生成一个随机数，用来选择一个图片显示在页面上
    <img src="./images/<%=(((int)(java.lang.Math.random()*8))+1)%>.jpg"/>
    //输出搜索表单
    <form method="get" id="searchform" action="./">
        ...
    </form>

    //输出些功能链接
    <li><a href="./<%=username%>">我的首页</a></li>
    <li><a href="./?o=10&uid=<%=userid%>">全部文章</a></li>
    <li><a href="./?o=5">写新文章</a></li>
    <li><a href="./?o=7">管理文章</a></li>
    <li style="float: right;"><a href="./?o=17">本站地图</a></li>
    <li style="float: right;"><a href="./">本站主页</a></li>

    //输出供用户输入文章的表单
    <form method="post" action="./?o=6">
        ...
    </form>
</body>
</html>
```

第五章 WEB-INF/rootjsp 目录下的 jsp

5.1 checkRootDeleteArticles.jsp

“核对 root 删除 文章”检查登录用户是不是 root(管理员)，如果是则有权删除任何文章。下面是全部代码。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%

//获取 session 的 userid 和 username 属性，用来检查用户是否登录
String userid=(String)session.getAttribute("userid");
String username=(String)session.getAttribute("username");

//如果尚未登录或不是以 root 身份登录则要求其以 root 身份登录
if(userid==null||userid.length()==0||(!userid.equalsIgnoreCase("1"))){
    request.setAttribute("erroMessage", "请以 root 身份登录进行操作。");
    request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
    return;
}

//获取请求删除的所有文章 id
String s=request.getParameter("p");
if(s==null||s.length()==0){
    request.setAttribute("erroMessage", "请求删除的文章 id 不能为空！");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}

//从字符串中分割出要删除的所有文章 id
String[] articleIds=s.split("\\s");

//检查要删除的每个文章 id 的格式是否正确
for (int i=0;i<articleIds.length;i++){
    if(!Check.isNum(articleIds[i])){
        request.setAttribute("erroMessage", "删除文章时使用的 articleid 不正确！cRDA");
        request.getRequestDispatcher("/erro.jsp").forward(request, response);
        return;
    }
}

//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");

for(int i=0;i<articleIds.length;i++){
        //再核对一遍
        if(username.equalsIgnoreCase("root")){
```

```
            //不能删除文章 id 为 1、2 的文章
            if((!articleIds[i].equals("1"))&&(!articleIds[i].equals("2"))){
            db.delete("delete from articles where articleid=?",articleIds[i]);
            }
        }else{
            //上面已经检查了一下 userid 是否是 root 的
            //所以在正常情况下 不会执行到这里!!
            request.setAttribute("erroMessage", "重要安全错误!! in cRDA");
            request.getRequestDispatcher("/erro.jsp").forward(request, response);
            return;
        }
    }

//转管理页面!
request.getRequestDispatcher("/WEB-INF/rootjsp/managerOverView.jsp")
                                                        .forward(request, response);
return;
%>
```

5.2 checkRootDeleteUsers.jsp

“核对 root 删除 用户”检查登录用户是不是 root(管理员)，如果是则有权限删除任何用户。下面是全部代码。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%
//获取 session 的 userid 和 username 属性，用来检查用户是否登录
String userid=(String)session.getAttribute("userid");
String username=(String)session.getAttribute("username");

//如果尚未登录或不是以 root 身份登录则要求其以 root 身份登录
if(userid==null||userid.length()==0||(!userid.equalsIgnoreCase("1"))){
    request.setAttribute("erroMessage", "请先登录。");
    request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
    return;
}

//获取请求删除的所有用户 id
String s=request.getParameter("p");
if(s==null||s.length()==0){
    request.setAttribute("erroMessage", "请求删除的用户 id 不能为空！");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
//从字符串中分割出要删除的所有用户 id
String[] userIds=s.split("\\s");
for (int i=0;i<userIds.length;i++){
    if(!Check.isNum(userIds[i])){
        request.setAttribute("erroMessage", "删除用户时使用的 userid 不正确！");
```

```
        request.getRequestDispatcher("/erro.jsp").forward(request, response);
        return;
    }
}
//获取 session 的 DBOperator 实例 用来进行数据库操作
DBOperator db= (DBOperator)session.getAttribute("db");
for(int i=0;i<userIds.length;i++){
        //再核对一遍
        if(username.equalsIgnoreCase("root")){
            if(!userIds[i].equals("1")){
                db.delete("delete from users where userid=?",userIds[i]);
            }
        }else{
            //上面已经检查了一下 userid 是否是 root 的
            //所以在正常情况下 不会执行到这里!!
            request.setAttribute("erroMessage", "重要安全错误!! in cRDU");
            request.getRequestDispatcher("/erro.jsp").forward(request, response);
            return;
        }
    }

//转管理页面!
request.getRequestDispatcher("/WEB-INF/rootjsp/managerOverView.jsp")
                                                        .forward(request, response);
return;
%>
```

5.3 initMySQLTables.jsp

“初始化 MySQL 表”初始化数据库表。创建 root 用户和两篇文章。其中的 sql 语句不便于阅读，请参阅 5.5 节（SQL.sql）。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="org.json.JSONException"%>
<%@page import="org.json.JSONObject"%>
<%@page import="bfy.Check"%>
<%
//获取传来的参数名为 root 的值，即初始化密码
String s=request.getParameter("root");
//获取 web.xml 中配置的密码
String installPassword=application.getInitParameter("installPassword");
//检查密码是否正确
if(s!=null&&s.length()<23&&s.equalsIgnoreCase(installPassword)){
    //获取 session 的 DBOperator 实例，以进行数据库操作
    DBOperator db=(DBOperator)session.getAttribute("db");
    //创建 users 表
    db.modify("create table users (userid int auto_increment, username varchar(20) not
null,password varchar(30) not null,alias varchar(13) default '你好',primary key(userid))");
    //创建 articles 表
    db.modify("create table articles( articleid int auto_increment, userid int not null, title
varchar(100) not null, content text not null, xdate date not null, primary key(articleid), foreign
```

```
key(userid) references users(userid) on delete cascade)");
db.modify("create table comments( commentid int auto_increment,  articleid int not null, name varchar(29) not null, info text not null, xdate date not null, primary key(commentid), foreign key(articleid) references articles(articleid) on delete cascade)");
    //向 users 表中插入 root 用户
    db.modify("INSERT INTO users (username, password, alias) VALUES ('root', 'root','root')");
    //向 articles 表中插入两篇文章
    db.modify("INSERT INTO articles (userid, title, content, xdate) VALUES (1, '欢迎信息', '欢迎使用。', '2013-01-01')");
db.modify("INSERT INTO articles (userid, title, content, xdate) VALUES (1, '使用说明', '使用说明。', '2013-01-01')");

}
else{
    //密码错误则转错误页面
    request.setAttribute("erroMessage","安装密码错误！");
    request.getRequestDispatcher("/erro.jsp").forward(request, response);
    return;
}
%>
<html>
<head>
    <title>initMySQLTables</title>
</head>
<body>
    initMySQLTables....OK!
    <br/>
    have fun!
</body>
</html>
```

5.4 managerOverView.jsp

“管理者 概览”此页面将显示所有用户、文章供 root 用户删除。

下面将列出此页面的关键 JSP 代码和经大量简化后的 HTML 代码。

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="bfy.DBOperator"%>
<%@page import="bfy.Check"%>
<%@page import="java.sql.*"%>
<%
//获取 session 的 userid 和 username 属性，用来检查用户是否登录
String userid=(String)session.getAttribute("userid");
String username=(String)session.getAttribute("username");

//如果尚未登录或不是以 root 身份登录则要求其以 root 身份登录
if(userid==null||userid.length()==0||(!userid.equalsIgnoreCase("1"))){
    request.setAttribute("erroMessage", "以 root 身份登录才能操作！");
    request.getRequestDispatcher("/WEB-INF/jsp/loginerro.jsp").forward(request, response);
    return;
}
//获取 session 的 db 属性以进行数据库操作
```

```
DBOperator db= (DBOperator)session.getAttribute("db");
//查询所需的信息
ResultSet rs_users= db.query("select userid,username from users");
ResultSet rs_articles= db.query("select articleid,title from articles");
%>
<!DOCTYPE html>
<html>
<head>
    <title>root 管理</title>
</head>
<body>
<%
    //如果用户表为空则输出“空!”
    if(!rs_users.next()){
        //输出信息..
        out.write("<strong>空! </strong>");
    }else{
%>
        //输出用户列表
        <form action="" method="post">
<%
            //输出用户信息
            do{
%>
                <li lang="<%=rs_users.getString("userid")%>">
                    <%=rs_users.getString("userid")%>||<%=rs_users.getString("username")%>
                </li>
<%
            }while(rs_users.next());
%>
        </form>
<%
    }
%>

<%
    //如果文章表为空则输出...
    if(!rs_articles.next()){
        //输出信息..
        out.write("<strong>空! </strong>");
    }else{
%>
            //输出文章列表
            <form id="deleteArticlesForm" action="" method="post">
<%
            do{
%>
                //输出文章列表
                <li lang="<%=rs_articles.getString("articleid")%>">
      <%=rs_articles.getString("articleid")%>||<%=rs_articles.getString("title")%>
                </li>
```

```
<%
             }while(rs_articles.next());
%>
            </form>
<%
    }
%>
</body>
</html>
```

5.5 SQL.sql

初始化表用到的 sql 语句。为便于阅读在这里集中列一下。

```
create table users
(
    userid      int auto_increment,
    username    varchar(20) not null,
    password    varchar(30) not null,
    alias       varchar(13) default '你好',
    primary key(userid)
)

create table articles
(
    articleid   int auto_increment,
    userid      int not null,
    title       varchar(100) not null,
    content     text not null,
    xdate       date not null,
    primary key(articleid),
    foreign key(userid) references users(userid) on delete cascade
)

create table comments
(
    commentid   int auto_increment,
    articleid   int not null,
    name        varchar(29) not null,
    info        text not null,
    xdate       date not null,
    primary key(commentid),
    foreign key(articleid) references articles(articleid) on delete cascade
)

INSERT INTO users (username, password, alias)
VALUES ('root', 'root','root')

INSERT INTO articles (userid, title, content, xdate)
VALUES
(
    1,
```

```
    '欢迎信息',
    '欢迎信息内容。',
    '2013-01-01'
)

INSERT INTO articles (userid, title, content, xdate)
VALUES
(
    1,
    '使用说明',
    '使用说明内容。',
    '2013-01-01'
)
```

第六章 系统中的 java 类

6.1 Check.java

此类用于核对 username、password 等是否符合规则。

```
package bfy;
public class Check{
    public Check() {}
    //检查是不是符合规定的  数字
    public static boolean isNum(String s){
    /*
     *  非空,5>长度>0,不能以 0 开头,每个字符都是 0-9
     */
        if((s!=null)&&(s.length()!=0)&&(s.length()<5)){
            char[] c=s.toCharArray();
            if(c[0]=='0') return false;
            for(int i=0;i<c.length;++i){
                if(c[i]<'0'||c[i]>'9')return false;
            }
            return true;
        }
        else
            return false;
    }
    //检查是不是符合规定的  名字
    public static boolean isName(String x){
        /*
         * 非空,0<长度<19,数字和字母的组合
         * 忽略大小写
         */
        if(x==null||x.length()==0)return false;
        String s=x.toLowerCase();
        if(s.length()<19){
            char[] c=s.toCharArray();
            for(int i=0;i<c.length;++i){
                if((c[i]<'0')||((c[i]>'9')&&(c[i]<'a'))||(c[i]>'z'))
                    return false;
            }
            return true;
        }
        else
            return false;
    }
    //检查是不是符合规定的  密码
    public static boolean isPasswrod(String s){
        /*
         * 3<长度<27，所有字符的 ASCII 码值∈[32,126]
         */
```

```
        if(s==null||s.length()<3) return false;
        if(s.length()<27){
            char[] c=s.toCharArray();
            for(int i=0;i<c.length;++i){
                if((c[i]<32)||(c[i]>126))
                    return false;
            }
            return true;
        }
        else    return false;
    }
}
```

6.2 DBOperator.java

此类用于方便地进行数据库操作。

```
package bfy;
import java.sql.*;
public class DBOperator
{
    private Connection connection;
    private String driver;
    private String url;
    private String username;
    private String password;
    public DBOperator(){}
    public DBOperator(String driver,String url,String username,String password)
    {
        this.driver = driver;
        this.url = url;
        this.username = username;
        this.password = password;
    }
    //下面是各个成员属性的 setter 和 getter 方法
    public void setDriver(String driver) {
        this.driver = driver;
    }
    public void setUrl(String url) {
        this.url = url;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getDriver() {
        return (this.driver);
    }
    public String getUrl() {
```

```
    return (this.url);
}
public String getUsername() {
    return (this.username);
}
public String getPassword() {
    return (this.password);
}
//获取数据库连接
public Connection getConnection() throws Exception
{
    if (connection == null)
    {
        Class.forName(this.driver);
        connection = DriverManager.getConnection(url,username,password);
    }
    return connection;
}
//插入记录
public boolean insert(String sql , Object... args)  throws Exception
{
    PreparedStatement pstmt = getConnection().prepareStatement(sql);
    for (int i = 0; i < args.length ; i++ )
    {
        pstmt.setObject( i + 1 , args[i]);
    }
    if (pstmt.executeUpdate() != 1)
    {
        return false;
    }
    pstmt.close();
    return true;
}
//执行查询
public ResultSet query(String sql , Object... args) throws Exception
{
    PreparedStatement pstmt = getConnection().prepareStatement(sql);
    for (int i = 0; i < args.length ; i++ )
    {
        pstmt.setObject( i + 1 , args[i]);
    }
    return pstmt.executeQuery();
}
//执行修改
public void modify(String sql , Object... args) throws Exception
{
    PreparedStatement pstmt = getConnection().prepareStatement(sql);
    for (int i = 0; i < args.length ; i++ )
    {
        pstmt.setObject( i + 1 , args[i]);
    }
```

```
        pstmt.executeUpdate();
        pstmt.close();
    }
    //关闭数据库连接的方法
    public void closeConnecton()    throws Exception
    {
        if (connection != null && !connection.isClosed())
        {
            connection.close();
        }
    }
    //删除操作!!
    public void delete(String sql , Object... args) throws Exception
    {
            PreparedStatement pstmt = getConnection().prepareStatement(sql);
            for (int i = 0; i < args.length ; i++ )
            {
                pstmt.setObject( i + 1 , args[i]);
            }
            pstmt.executeUpdate();
            pstmt.close();
    }
}
```

6.3 GetDate.java

此类将返回符合 MySQL 格式的日期（yyyy-mm-dd)。以插入 MySQL 的表中。

```
package bfy;
public class GetDate {

    public GetDate() {}
    public static String get(){
        java.util.Date date=new java.util.Date();
        String s=date.toString();
        String year=s.substring(24,28);
        String mon="00";
        switch(s.substring(4,7)){
         //(Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec).
            case "Jan":mon="01";break;
            case "Feb":mon="02";break;
            case "Mar":mon="03";break;
            case "Apr":mon="04";break;
            case "May":mon="05";break;
            case "Jun":mon="06";break;
            case "Jul":mon="07";break;
            case "Aug":mon="08";break;
            case "Sep":mon="09";break;
            case "Oct":mon="10";break;
            case "Nov":mon="11";break;
            case "Dec":mon="12";break;
```

```
            default:mon="00";break;
        }
        String day=s.substring(8,10);
        return year+"-"+mon+"-"+day;
    }
}
```

6.4 InitAppOnStart_cloudfoundry.java

“初始化 应用 当 启动_cloudfoundry”此类将实现 ServletContextListener 类，用于在应用（此博客系统）启动时初始化数据库信息。

```
package bfy;
import javax.servlet.ServletContext;
import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;
import org.json.JSONException;
import org.json.JSONObject;

public class InitAppOnStart implements ServletContextListener{
    public InitAppOnStart() {}
    @Override
    public void contextInitialized(ServletContextEvent e) {
        System.out.println("in listner init");
        ServletContext app=e.getServletContext();
        //获取 web.xml 中配置的环境变量名
        String envName=app.getInitParameter("envNameMySQL");
        try {
            //根据环境变量名获取环境变量
            JSONObject json = new JSONObject(System.getenv(envName));
            //获取环境变量中的数据库信息
            JSONObject mysqlCfg = json.getJSONArray("mysql-5.1")
                                      .getJSONObject(0).getJSONObject("credentials");
            //数据库驱动
            String driver="com.mysql.jdbc.Driver";
            //主机名
            String host = mysqlCfg.getString("host");
            //端口号
            String port = mysqlCfg.getString("port");
            //数据库名
            String name = mysqlCfg.getString("name");
            //用户名
            String username = mysqlCfg.getString("username");
            //密码
            String password = mysqlCfg.getString("password");

            //将上面获取的信息设置成整个应用（此博客系统）的属性
            //供运行期间随时取用
            app.setAttribute("driver", driver);
            app.setAttribute("url", "jdbc:mysql://"+host+":"+port+"/"+name);
            app.setAttribute("username", username);
```

```
            app.setAttribute("password", password);
        } catch (JSONException ee) {
            //输出捕获的 JSON 异常信息
            System.err.println(ee.getMessage());
        }
    }
    @Override
    public void contextDestroyed(ServletContextEvent e) {
        System.out.println("in listner destroy");
    }
}
```

6.5 Router.java

此类将实现 Filter 方法，拦截所有的请求，根据请求中的参数情况将请求定向到相应的处理页面。

```
package bfy;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class Router implements Filter
{
    //实现初始化方法
    //当应用启动时执行此初始化方法一遍
    public void init(FilterConfig config)
    {
        System.out.println("------in init:Router");
    }
    //实现销毁方法
    //当应用被关闭时执行此销毁方法一遍
    public void destroy()
    {
        System.out.println("------in destroy:Router");
    }
    //执行过滤的核心方法
    //收到的所有请求都要经过此方法的过滤
    public void doFilter(ServletRequest request,ServletResponse response,FilterChain chain)
                                                    throws IOException,ServletException
    {
        //将 ServletRequest 类型的 request 对象转换为 HttpServletRequest 类型的对象 hrequest
        HttpServletRequest hrequest = (HttpServletRequest)request;

        //获取 session 对象
        HttpSession session = hrequest.getSession();
        ServletContext application = session.getServletContext();

        //如果 session 的 db 属性为 null 则创建一个 DBOperator 对象，并把对象设为 session 范围的 db
属性
        //即：为每一个 session 设置一个 DBOperator 对象，用来完成本次 session 范围的数据库操作
        //DBOperator 的代码 可通过目录快速找到
```

```
if(session.getAttribute("db")==null){
    DBOperator db=new DBOperator(
            (String)application.getAttribute("driver"),
            (String)application.getAttribute("url"),
            (String)application.getAttribute("username"),
            (String)application.getAttribute("password")
        );
    session.setAttribute("db",db);
    System.out.println("----creat DB inrouter---");
}

//获取请求的‘a 参数’ ‘o 参数’ 和 ‘要访问的路径’
String aid=request.getParameter("a");   //例：http://xblog.x.x/?a=1  则 aid=1
String oid=request.getParameter("o");   //例：http://xblog.x.x/?o=1  则 oid=1
String path=hrequest.getServletPath();  //例：http://xblog.x.x/  则 path="/"
                                        //例：http://xblog.x.x/a  则 path="/aa"
                            //例：http://xblog.x.x/aa.bb  则 path="/aa.bb"

//调试时用的几个输出语句
//System.out.println("f------new-----");
//System.out.println("f---path:"+path);
//System.out.println("f---aid:"+aid);
//System.out.println("f---oid:"+oid);

if(aid!=null){                          //  -------------------------   标注 1
    //如果请求的"a 参数"非空，则
    //转 调用文章显示处理
    //将请求转给 articleShowById.jsp 处理
    request.getRequestDispatcher("/WEB-INF/articleShowById.jsp")
                                                    .forward(request, response);
}
else if(oid!=null){
    //如果请求的"o 参数"非空，则
    //转 调用相应的操作处理
    //将请求转给 operatorSelector.jsp 处理
    request.getRequestDispatcher("/WEB-INF/operatorSelector.jsp")
                                                    .forward(request, response);
}
else if(path.contains(".")){
    //如果请求中包含"点"就当请求的是 文件
    chain.doFilter(request, response);  //放行
}
else if(path.length()>0){
    //如果不符合上面的条件就当是 调用主页处理
    //即，调用某注册用户的主页
    //例：http://xblog.x.x/aa 则 path="/aa" ,所以下面要截取"aa"部分，不要"/".
    //即，要访问用户 aa 的主页
    request.setAttribute("username",path.substring(1));

    //将请求转给下 xMainPage.jsp 处理
    request.getRequestDispatcher("/WEB-INF/xMainPage.jsp").forward(request, response);
```

```
        }
        else{
            //在正常情况下，不会运行到这里
            //因为根据我的测试 上面的“path.length()>0”中的“path”最短时长度也>0
            request.setAttribute("erroMessage", "访问的路径错误！");
            request.getRequestDispatcher("/erro.jsp").forward(request, response);
            return;
        }
    }
}
```

第七章 部署应用

本章将简明扼要地介绍，将写好的应用部署在 cloudfoundry.com 上的过程。
注意：vmc 的命令行参数及 cloudfoundry.com 的 API 如果发生变化，请参考其官方文档。

7.1 配置 Ruby 环境，并安装 vmc

vmc 是用来向 cloudfoundry.com 上传应用及管理运行在 cloudfoundry.com 上的应用的命令行工具。(由于 vmc 是用 Ruby 写的，所以我们得先安装 Ruby)

从 http://www.rubyinstaller.org 下载并安装 Ruby。

启动 Ruby 命令行

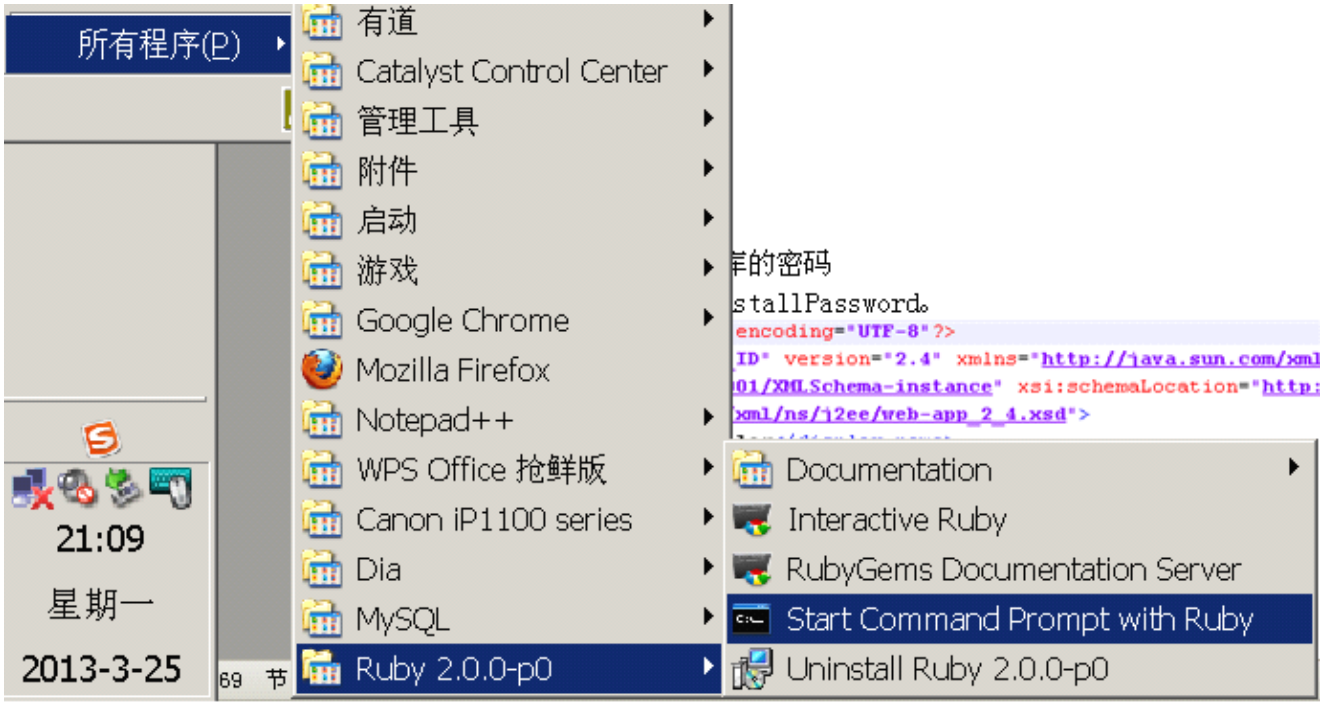

图 7.1

运行命令更新 RubyGems（如果速度较慢，就多等一会儿）(这个可以没有)：

```
gem  update  --system
```

运行命令安装 vmc（如果速度较慢，就多等一会儿）(这个必须有)：

```
gem install vmc
```

7.2 设置初始化数据库的密码

修改 web.xml 中的 installPassword。

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml
http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http:
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <display-name>xblog</display-name>

    <!--设置cloudfoundry.com上MySQL信息的环境变量名！ -->
    <context-param>
        <param-name>envNameMySQL</param-name>
        <param-value>VCAP_SERVICES</param-value>
    </context-param>


    <context-param>
        <param-name>installPassword</param-name>
        <param-value>xxxxxxx</param-value>                    改这里
    </context-param>
```

图 7.2

将 xxxxxxx 修改成自己的密码。此密码用来在初始化数据库表时，核对身份。将密码改好后，记下来，后面要用到。

将整个项目打包成 war 文件。将此文件放到一个空文件夹中，比如，在 D 盘新建个文件夹 temp，然后把 war 文件放进去。

7.3 上传应用

下面是部署的大体过程。

启动 Ruby 命令行。（见 7.1 节）
切换工作路径到 war 文件的所在目录。

```
C:\Documents and Settings\c>d:

D:\programs\Ruby200\bin>cd \temp
```

图 7.3

查看一下 war 文件是否在这里。

```
D:\temp>dir
 驱动器 D 中的卷没有标签。
 卷的序列号是 F426-F5CD

 D:\temp 的目录

2013-03-31  11:27    <DIR>          .
2013-03-31  11:27    <DIR>          ..
2013-03-31  11:27         2,198,603 xblog3.0.war
               1 个文件      2,198,603 字节
               2 个目录  7,605,653,504 可用字节
```

```
D:\temp>vmc target https://api.cloudfoundry.com
DL is deprecated, please use Fiddle
Setting target to https://api.cloudfoundry.com... OK

D:\temp>vmc push
DL is deprecated, please use Fiddle
Please log in first to proceed.

target: https://api.cloudfoundry.com

Email> [redacted].com                          ———————— 注册时使用的邮箱

Password> ********

Authenticating... OK
Name> [redacted]                               ———————— 应用名

Instances> 1

1: java_web
2: other
Framework> 1

1: java
2: java7
3: other
Runtime> 2

1: 64M
2: 128M
3: 256M
4: 512M
5: 1G
6: 2G
Memory Limit> 4

Creating [redacted]... OK
1: [redacted].cloudfoundry.com
2: none
Domain> [redacted].cloudfoundry.com            ———— 设置部署应用的域名

Updating [redacted]... FAILED      ———— 失败：由于域名应经被别人用了
The URI: [redacted].cloudfoundry.com" has already been taken or reserved

1: [redacted].cloudfoundry.com
2: none
Domain> [redacted].cloudfoundry.com            ———— 换一个试试

Updating [redacted]... OK                      ———————— 这个ok

Create services for application?> y             ———— 创建服务
```

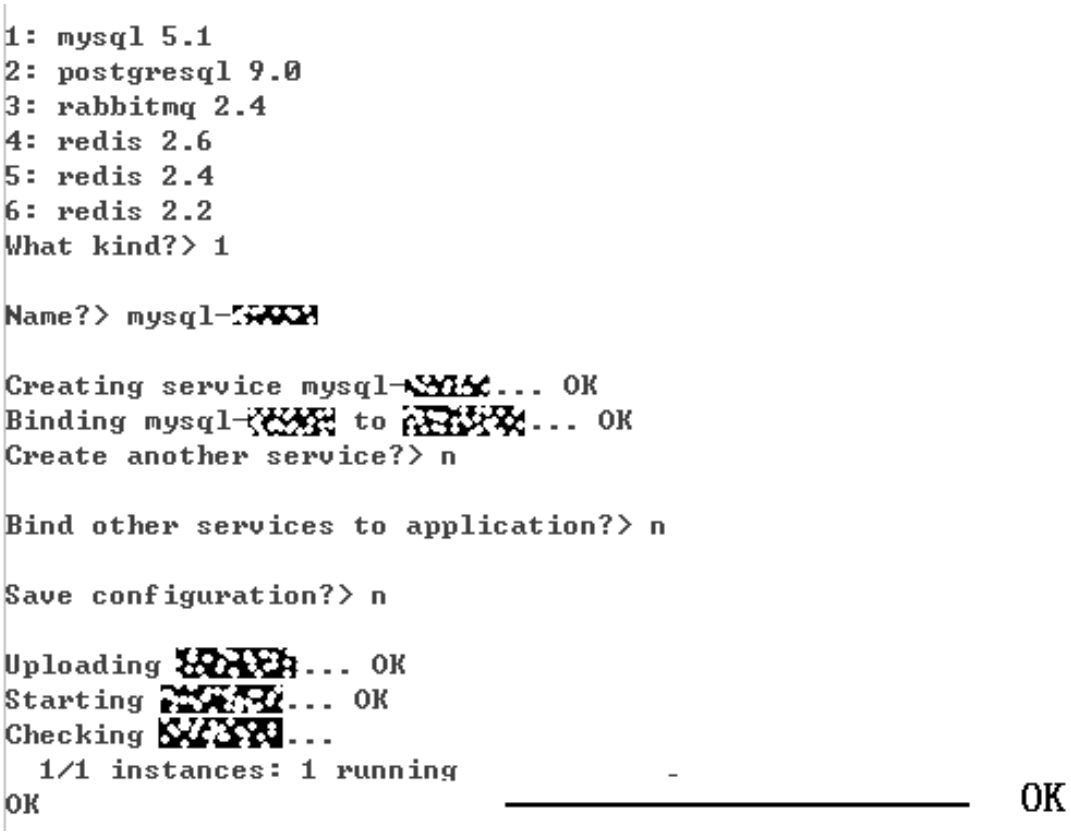

图 7.4

7.4 初始化数据库表

访问以下链接：
http://xxxx.cloudfoundry.com/?o=100&root=修改后的初始化密码
其中 xxxx 应替换成你部署应用时使用的域名。
“修改后的初始化密码”为 7.2 节中修改的密码。

如果显示：

```
initMySQLTables....OK!
have fun!
```

则说明初始化成功。

7.5 修改 root 密码

访问部署的博客系统主页：
http://xxxx.cloudfoundry.com（其中 xxxx 应替换成你部署应用时使用的域名）
点击右侧的“登录”链接。使用用户名：root 密码：root 登录。
登录后点击“修改密码”，修改 root 用户的密码。

7.6 使用 root 管理功能

以 root 身份登录后，可访问以下链接进行管理：
http://xxxx.cloudfoundry.com/?o=101（其中 xxxx 应替换成你部署应用时使用的域名）

7.7 关于管理应用的说明

如果要管理上传到 cloudfoundry.com 平台上的应用，请参见其官方文档说明。

参考文献

轻量级 Java EE 企业应用实战:Struts2+Spring3+Hibernate 整合开发/李刚编著.——3 版.——北京：电子工业出版社，2011.3

JAVA 核心技术，卷Ⅰ：基础知识（原书第 8 版）/（美）昊斯特曼（Horstmann,C.S.）著；叶乃文，邝劲筠,杜永萍译.——北京：机械工业出版社，2008.5

数据库系统原理：附数据库系统原理自学考试大纲/丁宝康主编.——北京：经济科学出版社，2007.3

计算机网络：自顶向下方法（原书第 4 版）/（美）库罗斯（Kurose,J.F.）等著；陈鸣译.——北京：机械工业出版社，2008.12

计算机科学概论：第 7 版/（美）布鲁克希尔（Brook shear,J.G.）著：王保江等译.——北京：人民邮电出版社，2003.9

C 程序设计语言（第 2 版）/（美）柯尼汉（Kernighan,B.W.），（美）里奇（Ritchie,D.M.）著；徐宝文等译.——北京：机械工业出版社，2001.3

C 程序设计语言（第 2 版·新版） 习题解答 / （美）汤朵（Tondo,C.L.），（美）吉米拜尔（Gimpel,S.E.）著；杨涛等译.——北京：机械工业出版社，2004.1

wikipedia	http://zh.wikipedia.org/
w3school	http://www.w3school.com.cn/
java	http://www.java.com/
jQuery	http://jquery.com/
tinyMCE	http://www.tinymce.com/
json	http://www.json.org/
cloudfoundry	http://www.cloudfoundry.com/

附录

• TinyMCE 的配置

将 TinyMCE 的包在网站上放置好后，在页面中引入 tiny_mce.js。引入后会自动从网站相应的位置调用需要的组件。并将页面中的 textarea 标签设置为 tiny_mce 编辑器。下面给出使用 tinyMCE 时的全部配置代码，和页面的示意代码。

```
<!DOCTYPE html>
<head>
    <title>写新文章</title>
<!-- TinyMCE -->
<script type="text/javascript" src="tiny_mce.js"></script>
<script type="text/javascript">
    tinyMCE.init({
        // General options
        language : "zh-cn",          //配置下语言
        mode : "textareas",          //将 textarea 标签显示为 tinyMCE 编辑器
        theme : "advanced",          //使用高级主题（功能多）
        plugins : "save,inlinepopups,fullscreen",    //加载些 plugin

        // Theme options             //配置编辑器上方的第一行工具栏
        theme_advanced_buttons1                                                    :
"save,|,bold,italic,underline,strikethrough,|,styleselect,formatselect,fontselect,fontsizesel
ect",

        //配置第二行工具栏
        theme_advanced_buttons2                                                    :
"undo,redo,|,bullist,numlist,|,justifyleft,justifycenter,justifyright,justifyfull,|,link,unli
nk,anchor,cleanup,code,|,insertdate,inserttime,preview,|,forecolor,backcolor,|,fullscreen",

        //配置工具栏的位置 顶部
        theme_advanced_toolbar_location : "top",
        //配置工具栏的对齐方式为：左对齐
        theme_advanced_toolbar_align : "left",
        //配置状态栏在底部
        theme_advanced_statusbar_location : "bottom",
        //将编辑器设为可调节大小
        theme_advanced_resizing : true,

        //一些样式
    style_formats : [
            {title : 'Bold text', inline : 'b'},
            {title : 'Red text', inline : 'span', styles : {color : '#ff0000'}},
            {title : 'Red header', block : 'h1', styles : {color : '#ff0000'}},
            {title : 'Example 1', inline : 'span', classes : 'example1'},
            {title : 'Example 2', inline : 'span', classes : 'example2'},
```

```
            {title : 'Table styles'},
            {title : 'Table row 1', selector : 'tr', classes : 'tablerow1'}
        ],
    });
</script>
<!-- /TinyMCE -->
</head>

<body>
    <div>
        <form method="post" action="./?o=6">

            <textarea name="content" style="height: 360px;"></textarea>

        </form>
    </div>
</body>
</html>
```

---------- 全文结束 ----------