

5.4 用 Servlet 管理广告条显示

5.4.1 实例说明

广告已经成为我们生活中不可或缺的部分，我们几乎无时无刻不在主动或被动地受到广告的影响。网络编程中当然也离不开广告，下边这个例子的作用就是轮流显示广告。因为广告客户已经为广告付钱，所以程序必须使所有的广告都能得到显示，而且机会均等。当然有些情况比如根据付款的多少区分优先级等，这里就不做考虑了。下边这个示例的作用是管理页面的广告显示。

5.4.2 编程思路

管理广告显示首先就是要能方便地添加广告条。广告的最基本的数据是图片和链接指向。保存数据最基本的有两种方法，一种是把图片保存在固定路径下，把图片的路径和链接保存在数据库中。这种方法涉及到两个问题，如果允许上传图片并保存图片，就必须开放保存图片位置的写权限。这样做的缺点首先是程序交给客户使用时，你必须告诉他要开放哪个位置的写权限，如果客户是个电脑盲，真的要花好大的力气才能使他明白。还有更重要的是安全问题，开放写权限对安全有着巨大的损害。另一种方法是把图片和链接都保存在数据库中，这种方法的缺点就是增加了服务器的开销。下边的实例就是采用这种方法。

5.4.3 准备工作

这里的准备工作主要是数据库的准备，这个示例还是使用 Mysql 数据库。首先创建一个 banner 表，并且这个表有如下的 4 个字段。

- id: banner 的惟一标识，它会自动增加。
- link: banner 的链接，很少有广告条没有链接。
- type: 图片类型，应该是 gif、jpg、png 中的一种。
- file: 图片。

banner.sql 的代码如下所示：

```
use test;
create table banner(
    id int primary key auto_increment,
    link varchar(200) not null,
    type varchar(50) not null,
    file longblob
);
```

直接导入 banner.sql 文件就可以创建 banner 表了，如图 5-6 所示。



```
Command Prompt - mysql
D:\>mysql<banner.sql -uroot -p
Enter password: *****

D:\>mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 3.23.52-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use test;
Database changed
mysql> desc banner;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)       |      | PRI | NULL    | auto_increment |
| link  | varchar(200)  |      |     |          |                |
| type  | varchar(50)   |      |     |          |                |
| file  | longblob      | YES  |     |          |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql>
```

图 5-6 创建一个 banner 表

5.4.4 代码分析

formData.java 的代码如下所示:

```
package mail;
//自定义数据类型
import java.util.*;
public class formData{
    public String name;        //表单项名称
    public String value;       //表单项的值, 如果是文件, 这个值是文件名
    public String contentType; //如果是文件, 这个值是文件的类型, 否则为 null
    public byte[] file;        //把文件保存为字节数组
}
```

前边已经讲过关于文件上传并解析表单数据, 但只是解析一个文件。这个例子稍微要复杂一些, 表单中可以有任意多的条目 (text、button、textarea 和 checkbox 等), 也可以有任意多的文件, 但原理还是一样的。

这个文件是解析表单后, 返回一个 formData 的 Vector, 并将文件保存在 byte 数组中。

parseRequest.java 的代码如下所示:

```
import java.util.*;
import java.io.*;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.ServletInputStream;
import javax.servlet.ServletException;

public class ParseRequest{
    private String ContentType = ""; //表单类型
    private String CharacterEncoding = ""; //编码格式

    //设定表单类型
    public void setContentType(String s){
        ContentType = s;
        int j;
```




```

        if((j = ContentType.indexOf("boundary=")) != -1){
            ContentType = ContentType.substring(j + 9);
            ContentType = "--" + ContentType;
        }
    }
    //设置文件编码格式
    public void setCharacterEncoding(String s){
        CharacterEncoding = s;
    }

    public Vector getFormData( HttpServletRequest req)
    throws ServletException, IOException{
        setCharacterEncoding(req.getCharacterEncoding());
        setContentType(req.getContentType());
        return getFormData(req.getInputStream());
    }
    //解析表单数据并保存到 Vector 中
    public Vector getFormData( ServletInputStream servletinputstream)
    throws ServletException, IOException{
        Vector formdatas=new Vector();
        formData fd=null;
        int size=2048000;//上传的文件的最大限制为 2M
        byte[] buffer = new byte[size];
        int bufferLength=0;
        byte temp[];
        byte Linebyte[] = new byte[4096]; //设定缓冲区大小为 4096
        //由于 Java 中不可以用指针，在函数中改变外部的值用数组方法
        int ai[] = new int[1];
        int ail[] = new int[1];

        String line;
        String boundary="--";
        String itemname="";
        String filename="";
        //textarea 的值可能有回车，用 StringBuffer 累加
        StringBuffer sb=new StringBuffer("");
        boolean nameflag=false;
        boolean fileflag=false;
        int skipline=0;
        int i;
        /*从 ServletInputStream 读取一行，注意这里不是严格的读取一行，如果在小于缓
        * 冲区大小的时候出现换行符，则读取一行，否则读满缓冲区
        * Linebyte: 缓冲区
        * ai: 读到的位置，也就是换行符的位置
        * servletinputstream: 浏览器请求
        * CharacterEncoding: 数据类型
        */
        while((line = readLine(Linebyte, ai, servletinputstream,
            CharacterEncoding)) != null){
            //得到 boundary

```



```

if((i=line.indexOf("boundary=="))>=0){
    boundary=line.substring(i+10);
    continue;
}
//这里使用 nameflag 和 fileflag 控制读取
if(!nameflag&&!fileflag){
    //得到表单项的名称
    itemname=getName(line);

    if(itemname!=null&&itemname!=""){
        fd=new FormData();
        fd.name=itemname;
        nameflag=true;
        //得到文件名
        filename=getFileName(line);

        if(filename!=null && filename!=""){
            fd.value=filename;
            fileflag=true;
        }else{
            fileflag=false;
        }
        skipline=0;
        continue;
    }
    //如果表单项的类型不是 file
    if(nameflag&&!fileflag){
        if(skipline<1){
            //跳过一行
            skipline++;
            continue;
        }
        else{
            //得到表单项的值
            if(line.indexOf(boundary)>=0){
                nameflag=false;
                fd.value=trim(sb.toString());
                formdatas.addElement(fd);
                sb=new StringBuffer("");
            }else{
                sb.append(line);
            }
        }
        continue;
    }
}
//如果表单项的类型是文件
if(nameflag&&fileflag){

```



```

        if(skipline<1){
            //得到文件类型
            if(line.indexOf("Content-Type:")>=0){
                fd.contentType=trim(line.substring(14));
            }
            else{
                //跳过一行
                skipline++;
            }
            continue;
        }else{
            //把文件内容保存到 byte 数组
            if(line.indexOf(boundary)>=0){
                if(bufferLength>=2){
                    //去掉最后的换行符
                    bufferLength-=2;
                    temp=new byte[bufferLength];
                    for(i=0;i<bufferLength;i++){
                        temp[i]=buffer[i];
                    }
                    fd.file=temp;
                    buffer=new byte[size];
                    bufferLength=0;
                }

                formdatas.addElement(fd);
                nameflag=false;
                fileflag=false;
            }else{
                for(i=0;i<ai[0];i++,bufferLength++){
                    buffer[bufferLength]=Linebyte[i];
                }
            }
        }
    }

    return formdatas;
}

//得到表单项的名称
public String getName(String s){
    int i=s.indexOf(" name=\"");
    if(i>0){
        s=s.substring(i+7);
        i=s.indexOf("\");
        return s.substring(0,i);
    }else{
        return "";
    }
}

```



//提取文件名

```
private String getFileName(String s){
    try{
        int i = s.indexOf("filename=");
        if(i >= 0){
            s = s.substring(i + 10);
            if((i = s.indexOf("\"")) > 0)
                s = s.substring(0, i);
            i = s.lastIndexOf("\\");
            if(i < 0 || i >= s.length() - 1){
                i = s.lastIndexOf("/");
                if(i < 0 || i >= s.length() - 1)
                    return s;
            }
            return s.substring(i + 1);
        }else
            return "";
    }catch(Exception e){
        return "";
    }
}
```

//使用 readLine 函数，以把表单提交上来的数据按行读取

```
private String readLine(byte Linebyte[], int ai[],ServletInputStream
servletinputstream,String CharacterEncoding){
    try{ //读取一行
        ai[0] = servletinputstream.readLine(Linebyte, 0, Linebyte.length);
        if(ai[0] == -1)
            return null;
    }catch(IOException ex){
        return null;
    }
    try{
        if(CharacterEncoding == null){
            //用默认的编码方式把给定的 byte 数组转换为字符串
            return new String(Linebyte, 0, ai[0]);
        }else{
            //用给定的编码方式把给定的 byte 数组转换为字符串
            return new String(Linebyte, 0, ai[0], CharacterEncoding);
        }
    }catch(Exception ex){
        return null;
    }
}
```

//这个 trim 函数用来去掉回车换行符，而不是 String 中的 trim 函数用表去掉空格

```
public String trim(String s){
    if(s.charAt(s.length()-1)=='\n'){
        if(s.length()<=2)
            return "";
        else
```



```

        s=s.substring(0,s.length()-2);
        return trim(s);
    }
    if(s.indexOf("\r\n")==0){
        s=s.substring(2);
        return trim(s);
    }
    return s;
}
}

```

addBanner.java 的代码如下所示:

//这个文件的作用是上传图片并保存到数据库中

```

import java.io.*;
import java.sql.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

```

```

public class addBanner extends HttpServlet {
    public void doGet(HttpServletRequest request,HttpServletResponse response)
        throws IOException, ServletException{
        response.setContentType("text/html;charset=gb2312");
        PrintWriter out = response.getWriter();
        try{
            ParseRequest pf=new ParseRequest();
            //解析表单数据
            Vector v= pf.getFormData(request);

            formData fd;
            String link="";
            String contentType="";
            byte[] image=null;

            Enumeration e=v.elements();
            while(e.hasMoreElements()){
                fd=(formData)e.nextElement();
                if(fd.name.equals("link")){
                    //得到广告的链接
                    link=fd.value;
                    continue;
                }else if(fd.name.equals("image")){
                    //得到图片
                    contentType=fd.contentType;
                    image=fd.file;
                }
            }
            if(!link.equals("")&&!contentType.equals("")&&image!=null){
                //注册驱动程序
                Class.forName("com.mysql.jdbc.Driver");
            }
        }
    }
}

```




```

//连接数据库的字符串
string sConnStr = "jdbc:mysql://localhost/test?user=root&password"
                + "=password&useUnicode=true&characterEncoding=8859_1";
//建立一个数据库连接
Connection conn = DriverManager.getConnection(sConnStr);
//添加广告信息的 SQL 语句
string sql="insert into banner (link,type,file) values(?,?,?)"
//建立准备状态
PreparedStatement pstmt = conn.prepareStatement(sql);
//加入数据
pstmt.setString(1,link);
pstmt.setString(2,contentType);
pstmt.setBytes(3,image);
//执行操作
pstmt.executeQuery();
//关闭数据库连接
pstmt.close();
conn.close();
out.println("连接和图片已经记入数据库");
}else{
    out.println("数据不完整, 请重新输入。");
}
}catch(NullPointerException npe)
}catch(Exception e){
    System.err.println(e);
    e.printStackTrace();
}

out.println("<html>");
out.println("<body>");
out.println("<head>");
out.println("</head>");
out.println("<body bgcolor=\"white\">");
out.println("<h3>上传图片</h3>");
    out.print("<form action=\"");
out.print("addBanner\" ");
out.println("method=POST enctype='multipart/form-data'>");
out.println("连接");
out.println("<input type=text size=20 name=link>");
out.println("<br>");
out.println("图片");
out.println("<input type=file size=20 name=image>");
out.println("<br>");
out.println("<input type=submit value='提交'>");
out.println("</form>");
out.println("</body>");
out.println("</html>");
}

//本文件无区别地对待 post 和 get 方法递交来的数据
public void doPost(HttpServletRequest request,HttpServletResponse

```




```
response)throws IOException, ServletException{
    doGet(request, response);
}
```

showBanner.java 的代码如下所示:

//这个文件的作用是随机显示 banner

```
import java.io.*;
import java.sql.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class showBanner extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException{
        try{
            Class.forName("com.mysql.jdbc.Driver");
            String sConnStr = "jdbc:mysql://localhost/test?user=root&password
+ "=password&useUnicode=true&characterEncoding=8859_1";
            Connection conn = DriverManager.getConnection(sConnStr);
            String sql="select id,link from banner where link!='' and
            type!='' and file!='';
            Statement stmt = conn.createStatement();
            ResultSet rs=stmt.executeQuery(sql);
            rs.last();                //把指针置底以便得到广告的数量
            Random r=new Random();
            //生成小于广告条数量的随机数
            int selectedbanner=Math.abs((r.nextInt())%(rs.getRow()));
            int i=0;
            int id=0;
            String link="";

            rs.absolute(1);           //把指针返回到顶部
            while(true){
                if(selectedbanner==i++){ //当 i 等于随机数时得到该 banner 的 id 和链接
                    id=rs.getInt("id");
                    link=rs.getString("link");
                    break;
                }
                rs.next();
            }
            //关闭数据库连接
            rs.close();
            stmt.close();
            conn.close();
        }
    }
}
```



```

        response.setContentType("text/html;charset=gb2312");
        PrintWriter out = response.getWriter();
        //输出显示随机广告条的 HTML 代码, 图片用 Servlet 显示
        out.print("<a href='"+link+"'><img border=0
src='showImage?id="+id+"'></a>");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

//本文件无区别地对待 post 和 get 方法递交来的数据
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException{
    doGet(request, response);
}
}

```

showImage.java 的代码如下所示:

//用来显示图片的 Servlet

```

import java.io.*;
import java.sql.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class showImage extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws IOException, ServletException{
        try{
            //得到要显示的图片的 ID
            String id=request.getParameter("id");
            //注册驱动程序
            Class.forName("com.mysql.jdbc.Driver");
            //初始化数据库连接字符串
            String sConnStr = "jdbc:mysql://localhost/test?user=root&password"
            +"=password&useUnicode=true&characterEncoding=8859_1";
            //连接数据库
            Connection conn = DriverManager.getConnection(sConnStr);
            //获取图片数据的 SQL 语句
            String sql="select type,file from banner where id="+id;
            //建立一个 Statement 对象
            Statement stmt = conn.createStatement();
            //执行查询
            ResultSet rs=stmt.executeQuery(sql);
            InputStream in=null;

```




```

//得到图片的类型和图片数据
if (rs.next())
    in = rs.getBinaryStream("file");
response.setContentType(rs.getString("type"));
//建立输出对象
ServletOutputStream sout = response.getOutputStream();
byte b[] = new byte[1024];
//输出图片
while(in.read(b)!=-1) {
    sout.write(b);
}
sout.flush();
sout.close();
//关闭数据库连接
rs.close();
stmt.close();
conn.close();
} catch(Exception e){
    e.printStackTrace();
}
}

public void doPost(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException{
    doGet(request, response);
}
}

```

5.4.5 运行结果

运行上述代码，可以添加广告条，如图 5-7 所示。

广告条添加成功后的页面如图 5-8 所示。

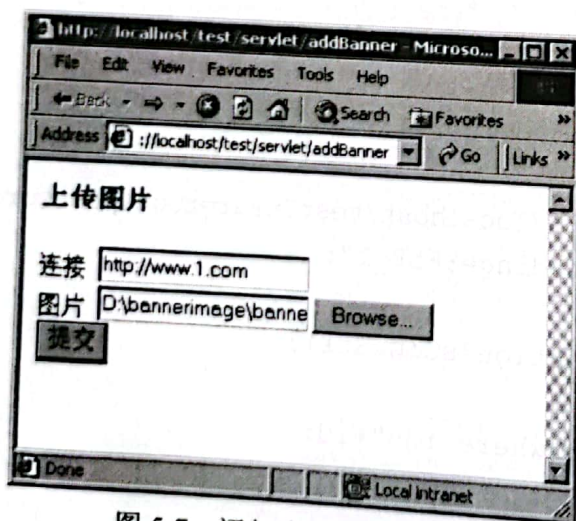


图 5-7 添加广告条的页面

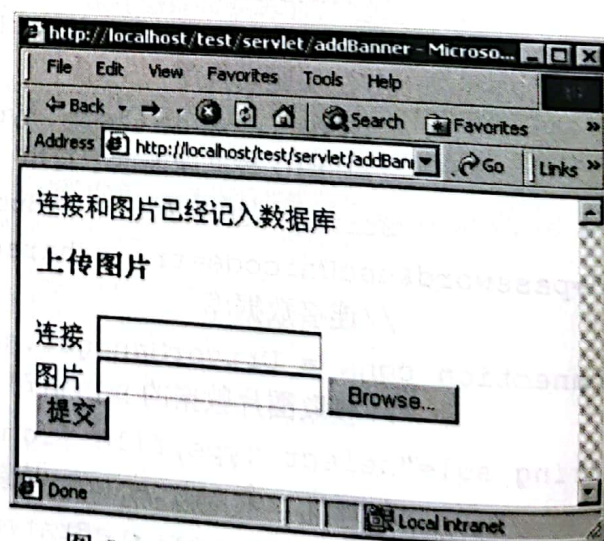


图 5-8 广告条添加成功后的页面

连续添加几个广告条后，运行 showBanner 程序，在连续刷新时添加的广告条会随机地出现在页面上，如图 5-9 所示。



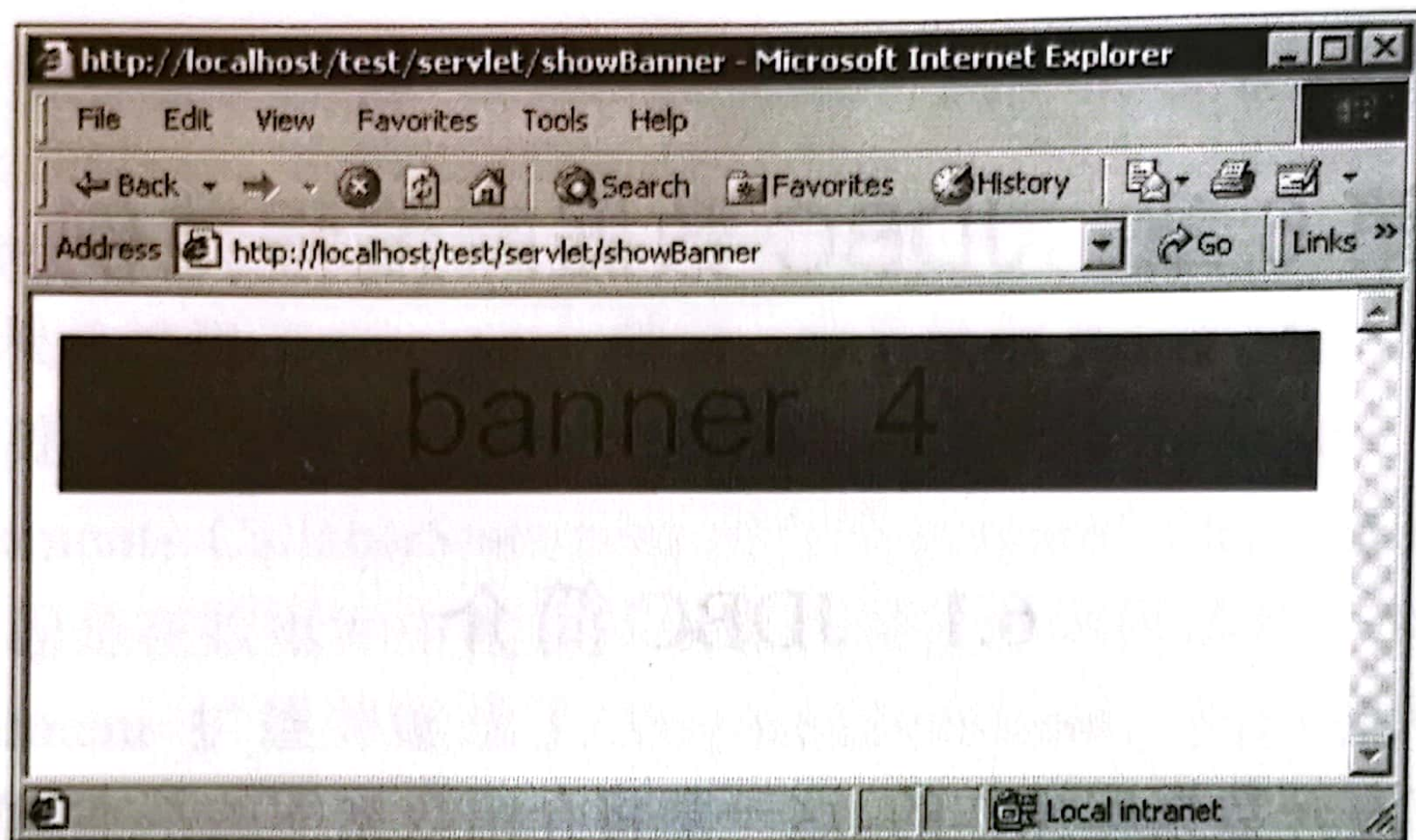


图 5-9 广告条的随机显示

