1. Servlet核心（I）

**1. Servlet核心（I）**

**1.1. 请求方式**

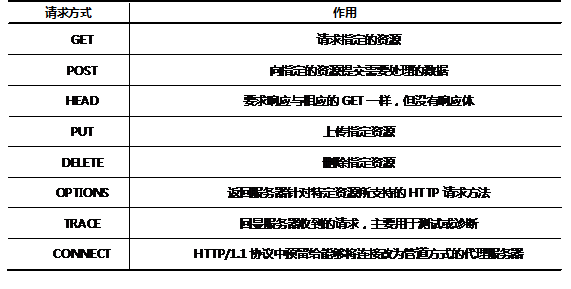
**1.1.1. 请求方式的种类**

请求数据包中的请求行的第一个参数就是请求方式，是客户端向Web服务器发送请求时的意向说明，以此告知服务器该如何处理及解析提交的这些数据，而客户端提交的数据也会因为采用的提交方式不同来处理。后续的一切解析都是基于请求方式的不同而不同。在RFC标准文档中有详细的规定来说明请求方式的种类及作用。

区分请求种类也会为浏览器采用不同的缓存方式处理后续请求提供一个依据，从而提升响应速度。所以对于编程人员来讲，使用正确的请求方式是Web应用的一个重要方面。

根据RFC文档的说明，请求的种类包括如 表 – 1 所示内容。

表 – 1 请求方式的种类



HTTP协议中定义了的这些方法（有时也叫“动作”）用来表示对指定数据的操作。有些方法（比如HEAD, GET, OPTIONS, and TRACE) 被定义为安全方法，这些方法针对的只是信息的返回，并不会改变服务器的状态（换句话说就是这些方法不会产生副作用）。不安全的方法（例如POST, PUT and DELETE) 应该用特殊的方式向用户展示，通常是按钮而不是链接，这样就可以使用户意识到可能要负的责任（例如一个按钮带来的资金交易。）

**1.1.2. GET请求方式的特点**

使用GET提交数据时，form中的数据将编码到URL中。在使用上，当且仅当请求幂等（字面意思是请求任意次返回同样的结果，本质是请求本身不会改变服务器数据和状态）时使用GET。重复访问时使用GET方法请求的页面，浏览器会使用缓存处理后续请求。当编程人员正确的使用GET后，浏览器会给出很好的缓存配合，响应速度更快。

原则上，服务器端处理GET和POST请求是没有分别的。但由于数据通过不同的方法编码，需要有不同的解码机制。所以，方法变化将导致处理请求的代码变化。比如对于CGI(通用网关接口：web服务器运行外部程序的规范，按CGI编写的程序可以扩展服务功能)，处理GET时通过环境变量获得参数，处理POST请求时则通过标准输入(stdin) 获得数据。

从使用经验上有如下总结：

* GET是把参数数据队列加到提交表单的ACTION属性所指的URL中，值和表单内各个字段一一对应，在URL中可以看到。
* GET传送的数据量较小，不能大于2KB。
* GET安全性非常低
* <form method="get" action="xxx?b=b">跟<form method="get" action="xxx">是一样的，也就是说，action页面后边带的参数列表会被忽视；而<form method="post" action="xxx?b=b">跟<form method="post" action="xxx">是不一样的。

如下情况中浏览器会发送GET类型的请求：

* 在地址栏输入一个地址
* 点击页面中的链接
* 表单的默认提交方式

**1.1.3. POST请求方式的特点**

POST方式提交的数据是在HTTP协议的Hearder中进行传输的。在使用上，当请求会改变服务器数据或状态时（更新数据，上传文件）应该使用POST。使用POST方法的Form提交时，浏览器基于POST将产生永久改变的假设，将让用户进行提交确认。

提交数据时，Form提交的第一步是创建数据集，并根据ENCTYPE对数据集进行编码。ENCTYPE有两个值：multipart/form-data，application/x-www-form-urlencoded（默认值），前者可同时用于GET、POST，后者只用于POST。进行数据传输时，对于GET方法，数据集使用content type application/x-www-form-urlencoded编码并附在URL后面，在这种模式下，数据严格限制为ASCII码；对于POST，使用content-type编码字符集并将其构造成消息发送

服务器端处理数据时，处理POST请求时则通过标准输入(stdin) 获得数据。

从使用经验上有如下总结：

* POST是将表单内各个字段与其内容放置在实体内容中内一起传送到ACTION属性所指的URL地址。用户看不到这个过程。
* POST传送的数据量较大，一般被默认为不受限制
* POST安全性较高

将表单的Method属性设置为POST时，浏览器会发送POST请求。

**1.2. Servlet如何运行**

**1.2.1. Servlet运行的步骤**

Servlet作为Web服务器的补充功能在运行时需要受到Servlet容器的管理，其运行的流程如下：

1. 浏览器依据IP建立与容器的连接
2. 浏览器将请求数据打包
3. 容器解析请求数据包，封装request和response对象
4. 容器依据请求资源路径找到Servlet创建对象
5. 容器将request和response对象作为参数传入service方法，并调用
6. 容器将响应数据打包发给浏览器
7. 浏览器取出结果，生成页面

**1.3. Servlet如何输出中文**

**1.3.1. 为什么返回的页面会有乱码**

Java语言在内存当中保存的每一个字符使用的都是Unicode字符集。一个中文字在内存中使用2个字节来表示。

编码：将Unicode字符集对应的字节数组转换成某种本地字符集（比如UTF-8，GBK）对应的字节数组的过程叫做编码

解码：将某种本地字符集对应的字节数组转换成Unicode字符集对应的字节数组的过程。

如果想输出就涉及到编码问题。编码和解码使用的字符集不一致就产生了乱码问题。

out.println（“”）这行代码是编码过程。现在这个out要输出的内容使用的是Unicode字符集，于是转换为默认的iso-8859-1。之所以使用这个格式是因为tomcat默认是英文的。所以调用out.println时，容器采用默认情况下的ISO-8859-1字符集去编码带有中文的输出内容时就产生了乱码。

**1.3.2. 如何解决输出内容的乱码问题**

在service（）方法中第一行的位置上添加如下代码：

1. response**.**setContentType**(**“text**/**html**;**charset**=**UTF**-8**”**);**

其中charset可以使用其他支持中文的字符集，如GBK。setContentType（）有两个作用：

1. 通知容器，在调用out.println方法输出时，使用指定的字符集
2. 生成一个消息头（content-type），通知浏览器，服务器返回的数据类型和字符集

使用该段代码修改默认的编码方式时，一定要保证在调用print之前编写，所以该段代码尽量放在service方法的第一行的位置。在charset之前使用的是分号隔开，如果写错，则会出现保存文件的界面，原因是浏览器不能识别消息头的值，于是让用户来处理。

能够输出带有中文内容页面的完整代码如下：

1. **import** java**.**io**.**IOException**;**
2. **import** java**.**io**.**PrintWriter**;**
3. **import** javax**.**servlet**.**ServletException**;**
4. **import** javax**.**servlet**.**http**.**HttpServlet**;**
5. **import** javax**.**servlet**.**http**.**HttpServletRequest**;**
6. **import** javax**.**servlet**.**http**.**HttpServletResponse**;**
7. **public** **class** HelloServlet **extends** HttpServlet**{**
8. **public** **void** service**(**HttpServletRequest request**,**
9. HttpServletResponse response**)throws** ServletException**,**
10. IOException**{**
11. response**.**setContentType**(**"text/html;charset=utf-8"**);**
12. //获得输出流
13. PrintWriter out **=** response**.**getWriter**();**
14. out**.**println**(**“达内科技欢迎你”**);**
15. //关闭流
16. out**.**close**();**
17. **}**
18. **}**

**1.4. Servlet如何处理表单**

**1.4.1. 如何获得请求参数值**

对于表单提交的数据，Servlet可以从容器构建的request对象中获取，如下两个方法可以在不同情况下获取表单数据：

* String getParameter（String paramName）
* String[] getParameterValues（String paramName）

**1.4.2. getParameter方法**

主要用于获取表单中控件的数据。其中参数名一定要与客户端表单中的控件name属性相一致。所以在构建表单各元素时，name属性一定要有。而name属性和id属性的区别就在于，id属性一般是作为客户端区分控件的标识，name属性是服务器端区分各控件的标识。如果参数名写错，则该方法会返回null。

用于提供用户输入的HTML页面代码如下：

1. **<**html**>**
2. **<**head**>**
3. **</**head**>**
4. **<**body**>**
5. **<**form action**=**"hello" method**=**"post"**>**
6. 用户名**:<**input name**=**"username"**/><**br**/>**
7. **<**input type**=**"submit" value**=**"提交"**/>**
8. **</**form**>**
9. **</**body**>**
10. **</**html**>**

使用该方法获取表单数据的完整代码如下：

1. **package** web**;**
2. **import** java**.**io**.**IOException**;**
3. **import** java**.**io**.**PrintWriter**;**
4. **import** javax**.**servlet**.**ServletException**;**
5. **import** javax**.**servlet**.**http**.**HttpServlet**;**
6. **import** javax**.**servlet**.**http**.**HttpServletRequest**;**
7. **import** javax**.**servlet**.**http**.**HttpServletResponse**;**
8. **public** **class** HelloServlet **extends** HttpServlet**{**
9. **public** **void** service**(**HttpServletRequest
10. request**,**HttpServletResponse response**)**
11. **throws** ServletException**,**IOException**{**
12. /\*
13. \* request.getParameter方法：
14. \*     1.作用：依据请求参数名获取请求参数值。
15. \*/
16. String username **=** request**.**getParameter**(**"username"**);**
17. //处理请求
18. String msg **=** "<div style='font-size:80px'>" **+**
19. "你好 " **+** username **+** "</div>"**;**
20. response**.**setContentType**(**"text/html;charset=utf-8"**);**
21. //获得输出流
22. PrintWriter out **=** response**.**getWriter**();**
23. out**.**println**(**msg**);**
24. //关闭流
25. out**.**close**();**
26. **}**
27. **}**

**1.4.3. getParameterValues方法**

用于获取表单中一组参数名相同的控件提交的数据组，如复选框，所以该方法返回的是字符串数组。如果用户没有选择这一组同名控件的任何一个，则该方法返回null。

用于提供用户输入用户名和爱好选项的HTML页面完整代码如下：

1. **<**html**>**
2. **<**head**>**
3. **</**head**>**
4. **<**body**>**
5. **<**form action**=**"hello" method**=**"post"**>**
6. 用户名**:<**input name**=**"username"**/><**br**/>**
7. 爱好**:**
8. 吃饭**<**input type**=**"checkbox" name**=**"hobby" value**=**"eating" checked**=**"checked"**/>**
9. 睡觉**<**input type**=**"checkbox" name**=**"hobby" value**=**"sleeping"**/>**
10. 打豆豆**<**input type**=**"checkbox" name**=**"hobby" value**=**"fighting"**/>**
11. **<**br**/>**
12. **<**input type**=**"submit" value**=**"提交"**/>**
13. **</**form**>**
14. **</**body**>**
15. **</**html**>**

使用该方法获取复选框值的完整代码如下：

1. **import** java**.**io**.**IOException**;**
2. **import** java**.**io**.**PrintWriter**;**
3. **import** javax**.**servlet**.**ServletException**;**
4. **import** javax**.**servlet**.**http**.**HttpServlet**;**
5. **import** javax**.**servlet**.**http**.**HttpServletRequest**;**
6. **import** javax**.**servlet**.**http**.**HttpServletResponse**;**
7. **public** **class** HelloServlet **extends** HttpServlet**{**
8. **public** **void** service**(**HttpServletRequest
9. request**,**HttpServletResponse response**)**     **throws** ServletException**,**IOException**{**
10. String username **=**     request**.**getParameter**(**"username"**);**
11. String**[]** hobby**=** request**.**getParameterValues**(**"hobby"**);**
12. //处理请求
13. String msg **=** "<div style='font-size:80px'>" **+**
14. "你好 " **+** username **+** "</div>"**;**
15. response**.**setContentType**(**"text/html;charset=utf-8"**);**
16. PrintWriter out **=** response**.**getWriter**();**
17. out**.**println**(**msg**);**
18. **for(int** i**=0;**i**<**hobby**.**length**;**i**++){**
19. out**.**println**(**"<h1>" **+** hobby**[**i**]** **+**"</h1>"**);**
20. **}**
21. //关闭流
22. out**.**close**();**
23. **}**
24. **}**

**1.4.4. 为什么表单提交中文会出现乱码**

当表单提交时，浏览器会对表单中的中文参数值进行编码，而使用的编码是打开页面时所使用的字符集，如当前页面使用的UTF-8的字符集进行显示的，那么表单提交的数据就会以UTF-8的方式进行编码后传输，而Web服务器在默认情况下对提交的表单数据会使用ISO-8859-1的字符集来解码，编码与解码的方式不一致就产生了表单提交时的中文乱码问题。

**1.4.5. 如何解决表单提交时的中文乱码问题**

**步骤一、确保表单所在的页面按照指定的字符集打开**

在HTML页面中使用meta标记可以确保浏览器按照指定的字符集进行解码页面，并限定表单提交时的数据编码方式

1. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8"**>**

在服务器端需要在调用getParameter方法读取参数之前，告诉浏览器如何解码，使用如下代码即可完成该设置：

1. request**.**setCharacterEncoding**(**"utf-8"**);**

注意该方法一定要要放在所有request.getParameter方法之前。

这种方式只针对POST请求有效。

**1.4.6. 解决GET方式提交时的中文乱码问题**

**步骤一、确保表单所在的页面按照指定的字符集打开**

在HTML页面中使用meta标记可以确保浏览器按照指定的字符集进行解码页面，并限定表单提交时的数据编码方式

1. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8"**>**

**步骤二、完成ISO-8859-1到UTF-8格式的转换**

1. String username **=** request**.**getParameter**(**“username”**);**
2. username **=** **new** String**(**username**.**getBytes**(**“iso**-8859-1**”**),**“UTF**-8**”**);**

由于GET方式提交的任何数据在服务器端一定会以ISO-8859-1的方式进行解码，所以服务器端可以先按ISO-8859-1的方式获取字节数组，在将该数组转变成UTF-8对应的字符串形式。

解决表单提交时的中文乱码问题的完整代码如下：

1. **import** java**.**io**.**IOException**;**
2. **import** java**.**io**.**PrintWriter**;**
3. **import** javax**.**servlet**.**ServletException**;**
4. **import** javax**.**servlet**.**http**.**HttpServlet**;**
5. **import** javax**.**servlet**.**http**.**HttpServletRequest**;**
6. **import** javax**.**servlet**.**http**.**HttpServletResponse**;**
7. **public** **class** HelloServlet **extends** HttpServlet**{**
8. **public** **void** service**(**HttpServletRequest     request**,**
9. HttpServletResponse response**)throws**
10. ServletException**,**IOException**{**
11. request**.**setCharacterEncoding**(**"utf-8"**);**
12. String username **=**     request**.**getParameter**(**"username"**);**
13. username**=new** String**(**username**.**getBytes**(**"iso-8859-1"**),**"utf-8"**);**
14. String msg **=** "<div style='font-size:80px'>" **+**
15. "你好 " **+** username **+** "</div>"**;**
16. response**.**setContentType**(**"text/html;charset=utf-8"**);**
17. PrintWriter out **=** response**.**getWriter**();**
18. out**.**println**(**msg**);**
19. out**.**close**();**
20. **}**
21. **}**

**1.5. Servlet如何访问数据库**

**1.5.1. 使用JDBC技术访问数据库**

**步骤一、将JDBC驱动（.jar文件）放到WEB-INF\lib下**

原因：ClassLoader找到字节码文件，然后加载到JVM的方法区中，变成一个对象。Tomcat都有自己的类加载器，会去WEB-INF下面lib中找字节码文件。毕竟jar包中都是字节码文件。

**步骤二、编写JDBC代码，需要注意异常的处理**

**JAVA SERVLETJSP DAY02**

1. Servlet获取请求参数值
2. 请求方式GET和POST的区别
3. 处理POST请求中的中文参数值
4. 处理GET请求中的中文参数值
5. 员工管理——添加员工信息
6. 员工管理——使用JDBC添加员工信息
7. 员工管理——使用JDBC查询所有员工信息

**1 Servlet获取请求参数值**

**1.1 问题**

在Servlet中获取客户端通过表单等形式提交的请求参数值，并输出显示。

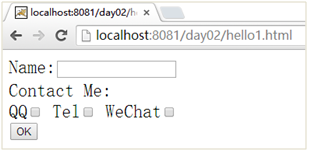
**1.2 方案**

使用request对象的getParameter（）方法获取请求中的name-value，使用request对象的getParameterValues（）方法获取请求中的name-values。

**1.3 步骤**

**步骤一：新建hello1.html页面**

新建hello1.html页面，内容如图-1所示：



图– 1

该页面表单内容如图-2所示：

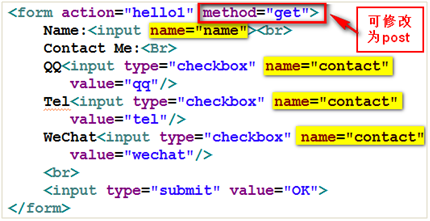


图 - 2

**步骤二：新建Hello1Servlet类**

新建Hello1Servlet.java文件，用于获取参数值及输出。

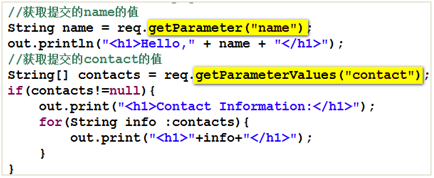


图 - 3

**步骤三：修改web.xml文件**

在web.xml文件中添加Hello1Servlet的映射



图 - 4

**步骤四：部署并访问工程**

部署工程后启动Tomcat，在浏览器中输入地址访问hello1.html页面，注意，本案例暂时只输入英文，查看服务器端获取参数值的输出结果。

**1.4 完整代码**

helo1.html文件的代码如下：

**<**html**>**

**<**head**>**

**</**head**>**

**<**body style**=**"font-size:24px"**>**

**<**form action**=**"hello1" method**=**"get"**>**

    Name**:<**input name**=**"name"**><**br**>**

    Contact Me**:<**Br**>**

    QQ**<**input type**=**"checkbox" name**=**"contact"

     value**=**"qq"**/>**

    Tel**<**input type**=**"checkbox" name**=**"contact"

     value**=**"tel"**/>**

    WeChat**<**input type**=**"checkbox" name**=**"contact"

     value**=**"wechat"**/>**

**<**br**>**

**<**input type**=**"submit" value**=**"OK"**>**

**</**form**>**

**</**body**>**

**</**html**>**

Hello1Servlet.java文件的代码如下：

package web**;**

import java**.**io**.**IOException**;**

import java**.**io**.**PrintWriter**;**

import javax**.**servlet**.**ServletException**;**

import javax**.**servlet**.**http**.**HttpServlet**;**

import javax**.**servlet**.**http**.**HttpServletRequest**;**

import javax**.**servlet**.**http**.**HttpServletResponse**;**

**public** **class** Hello1Servlet **extends** HttpServlet **{**

    /\*\*

     \* 获取请求中的name、contact两组参数值

     \* 分别使用getParameter和getParameterValues方法获取

     \*/

**protected** void service**(**HttpServletRequest req**,** HttpServletResponse resp**)**

**throws** ServletException**,** IOException **{**

        PrintWriter out **=** resp**.**getWriter**();**

        //获取提交的name的值

        String name **=** req**.**getParameter**(**"name"**);**

        out**.**println**(**"<h1>Hello," **+** name **+** "</h1>"**);**

        //获取提交的contact的值

        String**[]** contacts **=** req**.**getParameterValues**(**"contact"**);**

**if(**contacts**!=null){**

            out**.**print**(**"<h1>Contact Information:</h1>"**);**

**for(**String info **:**contacts**){**

                out**.**print**(**"<h1>"**+**info**+**"</h1>"**);**

**}**

**}**

        out**.**close**();**

**}**

**}**

web.xml文件的代码如下：

**<?**xml version**=**"1.0" encoding**=**"UTF-8"**?>**

**<**web**-**app version**=**"2.5"

    xmlns**=**"http://java.sun.com/xml/ns/javaee"

    xmlns**:**xsi**=**"http://www.w3.org/2001/XMLSchema-instance"

    xsi**:**schemaLocation**=**"http://java.sun.com/xml/ns/javaee

    http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"**>**

**<**servlet**>**

**<**servlet**-**name**>**hello1Servlet**</**servlet**-**name**>**

**<**servlet**-class>**web**.**Hello1Servlet**</**servlet**-class>**

**</**servlet**>**

**<**servlet**-**mapping**>**

**<**servlet**-**name**>**hello1Servlet**</**servlet**-**name**>**

**<**url**-**pattern**>**/hello1</url**-**pattern**>**

**</**servlet**-**mapping**>**

**</**web**-**app**>**

**2 请求方式GET和POST的区别**

**2.1 问题**

查看客户端在提交请求参数值时，GET请求方式和POST请求方式的根本区别，即参数值的传递方式的区别。

**2.2 方案**

使用MyEclipse自带的TCP/IP Monitor，以代理服务器的方式解析不同请求数据包中的请求行、消息头、主体的内容。

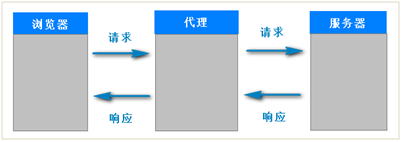


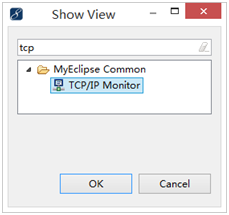
图 - 5

**2.3 步骤**

**步骤一：新建代理服务器并启动TCP/IP Monitor**

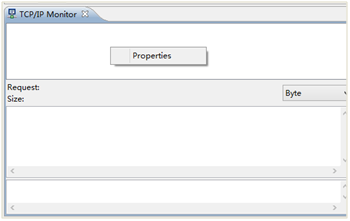
在MyEclipse中，按照如下步骤打开TCP/IP Monitor视图。

菜单“Windows”(“Show View”(“Other…”在搜索框中输入“TCP”就可看到该视图，选中后点击“OK”即可打开。如图-6：



图– 6

在该视图的空白处右键，点击“Properties”如图-7:



图– 7

在弹出的窗口中，按照如下图-8至图-10所示的顺序依次操作，完成Monitor的创建及启动。

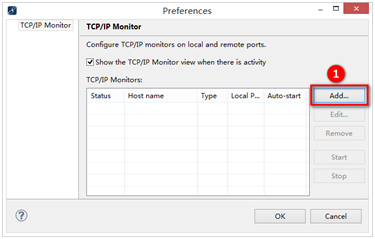


图-8

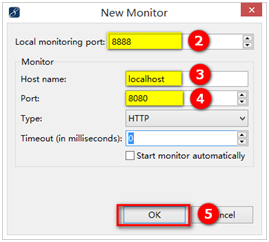


图-9

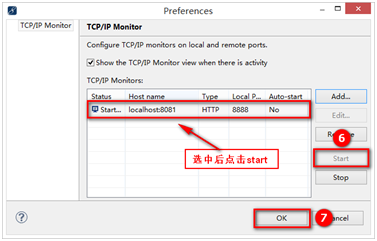


图-10

**步骤二：访问代理服务器，以GET方式提交表单数据**

使用上一个案例中的表单来模拟提交数据即可。本次注意表单的method值为get。

1. **<**form action**=**“hello1” method**=**“get“**>**

启动Tomcat后，输入如下地址http://localshot:8888/day02/hello1.html

http://pdf7.tarena.com.cn/tts8_source/ttsPage/JAVA/JSD_V05/SERVLETJSP/DAY02/CASE/01/index.files/image011.png

图– 11

填写表单数据后点击OK，查看Monitor中的数据包如图-12

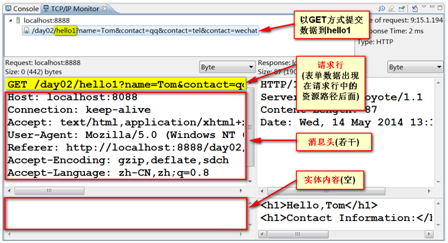


图 -12

**步骤三：访问代理服务器，以POST方式提交表单数据**

修改表单的method值为post。

1. **<**form action**=**“hello1” method**=**“post“**>**

重新部署后，输入代理服务器地址http://localshot:8888/day02/hello1.html填写表单数据后点击OK后查看Monitor中的数据包如图-13

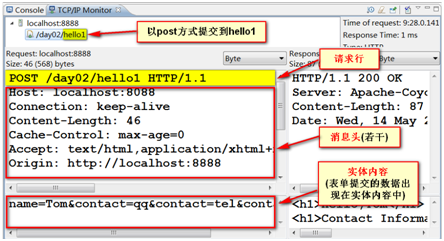


图- 13

**步骤四：对比代理服务器中的数据包数据**

对比图-12和图-13抓取的请求数据包可以看出，get方式和post方式在提交表单数据时的根本区别在于数据所处的位置。get方式参数值在请求资源路径的后面，从请求行汇总传递过来。post方式参数值出现在实体内容中，请求资源路径上不会出现任何多余信息，

**3 处理POST请求中的中文参数值**

**3.1 问题**

表单以post方式提交中文数据到服务器端会出现乱码的现象，增加对乱码的处理，使得POST方式能够正确提交数据并显示。

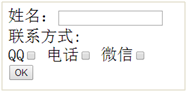
**3.2 方案**

为html页面增加meta标记，保证浏览器以支持中文编码方式打开页面。在服务器端的Servlet获取请求参数值时，按照与客户端相同的编码方式来解码，就可以实现正确获取，在输出流中同样设置与该编码方式相同格式来控制中文的输出显示。

**3.3 步骤**

**步骤一：新建hello2.html页面并增加meta标记**

新建hello2.html页面如下所示



图– 14

表单为post提交方式，文件源代码如下:

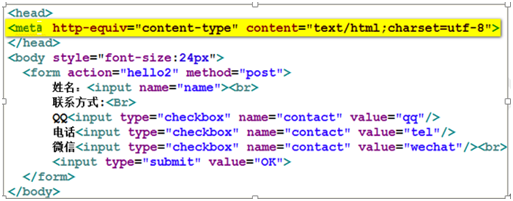


图 - 15

**步骤二：新建Hello2Servlet.java文件**

新建Hello2Servlet.java文件，添加request.setCharacterEncoding来设置取参数值之前的解码方式，为了保证输出中文，使用response.setContentType来设置content-type的消息头，从而控制浏览器的解码方式。

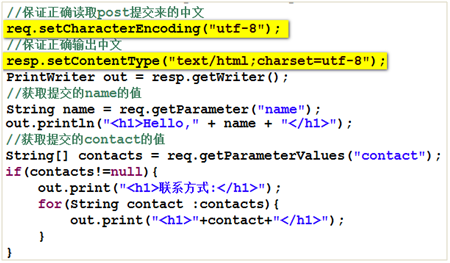


图 - 16

**步骤三：修改web.xml文件**

添加Hello2Servlet的映射，如图-17



图 - 17

**步骤四：部署并访问工程**

部署工程,启动服务，输入地址，并输入中文表单数据后查看结果。



图 - 18

**3.4 完整代码**

hello2.html完整代码如下：

**<**html**>**

**<**head**>**

**<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8"**>**

**</**head**>**

**<**body style**=**"font-size:24px"**>**

**<**form action**=**"hello2" method**=**"post"**>**

    姓名：**<**input name**=**"name"**><**br**>**

    联系方式**:<**Br**>**

    QQ**<**input type**=**"checkbox" name**=**"contact" value**=**"qq"**/>**

    电话**<**input type**=**"checkbox" name**=**"contact" value**=**"tel"**/>**

    微信**<**input type**=**"checkbox" name**=**"contact" value**=**"wechat"**/><**br**>**

**<**input type**=**"submit" value**=**"OK"**>**

**</**form**>**

**</**body**>**

**</**html**>**

Hello2servlet.java完整代码如下：

1. package web**;**
2. import java**.**io**.**IOException**;**
3. import java**.**io**.**PrintWriter**;**
4. import javax**.**servlet**.**ServletException**;**
5. import javax**.**servlet**.**http**.**HttpServlet**;**
6. import javax**.**servlet**.**http**.**HttpServletRequest**;**
7. import javax**.**servlet**.**http**.**HttpServletResponse**;**
8. **public** **class** Hello2Servlet **extends** HttpServlet **{**
9. **protected** void service**(**HttpServletRequest req**,** HttpServletResponse resp**)**
10. **throws** ServletException**,** IOException **{**
11. //保证正确读取post提交来的中文
12. req**.**setCharacterEncoding**(**"utf-8"**);**
13. //保证正确输出中文
14. resp**.**setContentType**(**"text/html;charset=utf-8"**);**
15. PrintWriter out **=** resp**.**getWriter**();**
16. //获取提交的name的值
17. String name **=** req**.**getParameter**(**"name"**);**
18. out**.**println**(**"<h1>Hello," **+** name **+** "</h1>"**);**
19. //获取提交的contact的值
20. String**[]** contacts **=** req**.**getParameterValues**(**"contact"**);**
21. **if(**contacts**!=null){**
22. out**.**print**(**"<h1>联系方式:</h1>"**);**
23. **for(**String contact **:**contacts**){**
24. out**.**print**(**"<h1>"**+**contact**+**"</h1>"**);**
25. **}**
26. **}**
27. out**.**close**();**
28. **}**
29. **}**

web.xml文件完整代码如下:

1. **<?**xml version**=**"1.0" encoding**=**"UTF-8"**?>**
2. **<**web**-**app version**=**"2.5"
3. xmlns**=**"http://java.sun.com/xml/ns/javaee"
4. xmlns**:**xsi**=**"http://www.w3.org/2001/XMLSchema-instance"
5. xsi**:**schemaLocation**=**"http://java.sun.com/xml/ns/javaee
6. http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"**>**
7. **<**servlet**>**
8. **<**servlet**-**name**>**hello2Servlet**</**servlet**-**name**>**
9. **<**servlet**-class>**web**.**Hello2Servlet**</**servlet**-class>**
10. **</**servlet**>**
11. **<**servlet**-**mapping**>**
12. **<**servlet**-**name**>**hello2Servlet**</**servlet**-**name**>**
13. **<**url**-**pattern**>**/hello2</url**-**pattern**>**
14. **</**servlet**-**mapping**>**
15. **</**web**-**app**>**

**4 处理GET请求中的中文参数值**

**4.1 问题**

表单以get方式提交中文数据到服务器端会出现乱码的现象，增加对乱码的处理，使得get方式能够正确提交数据并显示。

**4.2 方案**

为html页面增加meta标记，保证浏览器以支持中文编码方式打开页面。在服务器端的Servlet获取请求参数值时，按照iso-8859-1的方式获取字节码，再使用与客户端相同的编码方式来解码，

**4.3 步骤**

**步骤一：新建hello2.html页面并增加meta标记**

页面中的表单提交方式为get



图 - 19

**步骤二：新建Hello2Servlet.java文件**

表单提交到的Servlet中需要按照ios-8859-1的方式获取字节码，再将此字节码按照UTF-8方式的解码。

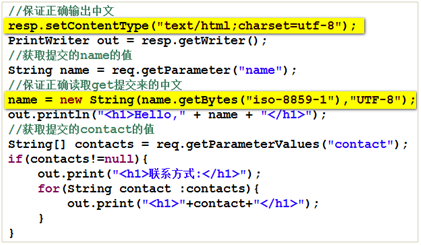


图 - 20

**步骤三：修改web.xml文件**

与图-17一致。

**步骤四：部署并访问工程**

部署并访问工程，输入中文姓名查看结果及地址栏。

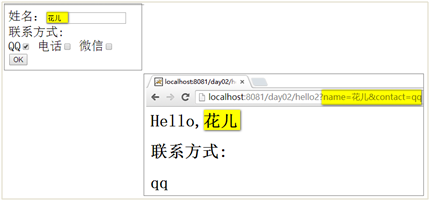


图 - 21

**4.4 完整代码**

hello2.html文件完整代码：

1. **<**html**>**
2. **<**head**>**
3. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8"**>**
4. **</**head**>**
5. **<**body style**=**"font-size:24px"**>**
6. **<**form action**=**"hello2" method**=**"get"**>**
7. 姓名：**<**input name**=**"name"**><**br**>**
8. 联系方式**:<**Br**>**
9. QQ**<**input type**=**"checkbox" name**=**"contact" value**=**"qq"**/>**
10. 电话**<**input type**=**"checkbox" name**=**"contact" value**=**"tel"**/>**
11. 微信**<**input type**=**"checkbox" name**=**"contact" value**=**"wechat"**/><**br**>**
12. **<**input type**=**"submit" value**=**"OK"**>**
13. **</**form**>**
14. **</**body**>**
15. **</**html**>**

Hello2Servlet.java文件代码如下：

1. package web**;**
2. import java**.**io**.**IOException**;**
3. import java**.**io**.**PrintWriter**;**
4. import javax**.**servlet**.**ServletException**;**
5. import javax**.**servlet**.**http**.**HttpServlet**;**
6. import javax**.**servlet**.**http**.**HttpServletRequest**;**
7. import javax**.**servlet**.**http**.**HttpServletResponse**;**
8. **public** **class** Hello2Servlet **extends** HttpServlet **{**
9. **protected** void service**(**HttpServletRequest req**,** HttpServletResponse resp**)**
10. **throws** ServletException**,** IOException **{**
11. //保证正确输出中文
12. resp**.**setContentType**(**"text/html;charset=utf-8"**);**
13. PrintWriter out **=** resp**.**getWriter**();**
14. //获取提交的name的值
15. String name **=** req**.**getParameter**(**"name"**);**
16. //保证正确读取get提交来的中文
17. name **=** **new** String**(**name**.**getBytes**(**"iso-8859-1"**),**"UTF-8"**);**
18. out**.**println**(**"<h1>Hello," **+** name **+** "</h1>"**);**
19. //获取提交的contact的值
20. String**[]** contacts **=** req**.**getParameterValues**(**"contact"**);**
21. **if(**contacts**!=null){**
22. out**.**print**(**"<h1>联系方式:</h1>"**);**
23. **for(**String contact **:**contacts**){**
24. out**.**print**(**"<h1>"**+**contact**+**"</h1>"**);**
25. **}**
26. **}**
27. out**.**close**();**
28. **}**
29. **}**

web.xml文件代码如下：

1. **<?**xml version**=**"1.0" encoding**=**"UTF-8"**?>**
2. **<**web**-**app version**=**"2.5"
3. xmlns**=**"http://java.sun.com/xml/ns/javaee"
4. xmlns**:**xsi**=**"http://www.w3.org/2001/XMLSchema-instance"
5. xsi**:**schemaLocation**=**"http://java.sun.com/xml/ns/javaee
6. http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"**>**
7. **<**servlet**>**
8. **<**servlet**-**name**>**hello2Servlet**</**servlet**-**name**>**
9. **<**servlet**-class>**web**.**Hello2Servlet**</**servlet**-class>**
10. **</**servlet**>**
11. **<**servlet**-**mapping**>**
12. **<**servlet**-**name**>**hello2Servlet**</**servlet**-**name**>**
13. **<**url**-**pattern**>**/hello2</url**-**pattern**>**
14. **</**servlet**-**mapping**>**
15. **</**web**-**app**>**

**5 员工管理——添加员工信息**

**5.1 问题**

编写工程实现收集员工姓名、薪水、年龄信息，将数据提交到服务器后，服务器能够正确获取参数值并输出显示，要求能够支持中文。

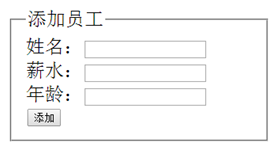
**5.2 方案**

制作表单用于获取用户输入，选择post方式提交数据，添加meta标签控制浏览器的编码方式。服务器端增加Servlet用于获取请求参数值，添加用于处理中文乱码的setCharaterEncoding，实现正确解码。使用setContentType实现中文的正确输出。

**5.3 步骤**

**步骤一：新建addEmp.html页面**

新建addEmp.html页面效果如图-22所示



图– 22

addEmp.html页面的表单如下：



图 - 23

**步骤二：新建AddEmpServlet类**

新建AddEmpServlet类，使用request.setCharacterEncoding("UTF-8")及response.setContentType("text/html;charset=UTF-8")来保证正确获取中文，正确输出中文。

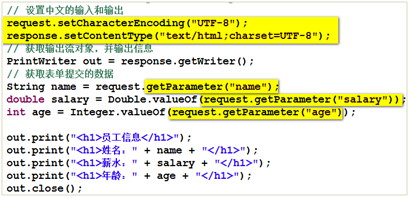


图- 24

**步骤三：修改web.xml文件**

修改web.xml文件，为AddEmpServlet添加映射，如图-25



图 - 25

**步骤四：部署及访问应用**

部署应用，启动Tomcat，输入访问addEmp.html的页面地址，输入中文，点击提交，查看运行结果。

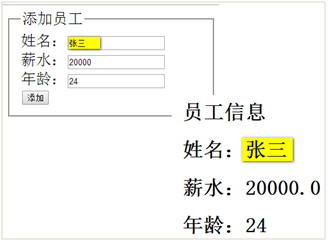


图 - 26

**5.4 完整代码**

addEmp.html文件完整代码：

**<**html**>**

**<**head**>**

**<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8"**>**

**</**head**>**

**<**body style**=**"font-size:24px"**>**

**<**form action**=**"add" method**=**"post"**>**

**<**fieldset**>**

**<**legend**>**添加员工**</**legend**>**

                姓名：**<**input name**=**"name"**/><**br**>**

                薪水：**<**input name**=**"salary"**/><**br**>**

                年龄：**<**input name**=**"age"**/><**br**>**

**<**input type**=**"submit" value**=**"添加"**/>**

**</**fieldset**>**

**</**form**>**

**</**body**>**

**</**html**>**

AddEmpServlet.java文件完整代码：

package web**;**

import java**.**io**.**IOException**;**

import java**.**io**.**PrintWriter**;**

import java**.**sql**.**Connection**;**

import java**.**sql**.**DriverManager**;**

import java**.**sql**.**PreparedStatement**;**

import java**.**sql**.**SQLException**;**

import javax**.**servlet**.**ServletException**;**

import javax**.**servlet**.**http**.**HttpServlet**;**

import javax**.**servlet**.**http**.**HttpServletRequest**;**

import javax**.**servlet**.**http**.**HttpServletResponse**;**

**public** **class** AddEmpServlet **extends** HttpServlet **{**

**public** void service**(**HttpServletRequest request**,** HttpServletResponse response**)**

**throws** ServletException**,** IOException **{**

        // 设置中文的输入和输出

        request**.**setCharacterEncoding**(**"UTF-8"**);**

        response**.**setContentType**(**"text/html;charset=UTF-8"**);**

        // 获取输出流对象，并输出信息

        PrintWriter out **=** response**.**getWriter**();**

        // 获取表单提交的数据

        String name **=** request**.**getParameter**(**"name"**);**

        double salary **=** Double**.**valueOf**(**request**.**getParameter**(**"salary"**));**

        int age **=** Integer**.**valueOf**(**request**.**getParameter**(**"age"**));**

        out**.**print**(**"<h1>员工信息</h1>"**);**

        out**.**print**(**"<h1>姓名：" **+** name **+** "</h1>"**);**

        out**.**print**(**"<h1>薪水：" **+** salary **+** "</h1>"**);**

        out**.**print**(**"<h1>年龄：" **+** age **+** "</h1>"**);**

        out**.**close**();**

**}**

**}**

web.xml文件完整代码：

1. **<?**xml version**=**"1.0" encoding**=**"UTF-8"**?>**
2. **<**web**-**app version**=**"2.5"
3. xmlns**=**"http://java.sun.com/xml/ns/javaee"
4. xmlns**:**xsi**=**"http://www.w3.org/2001/XMLSchema-instance"
5. xsi**:**schemaLocation**=**"http://java.sun.com/xml/ns/javaee
6. http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"**>**
7. **<**servlet**>**
8. **<**servlet**-**name**>**addEmpServlet**</**servlet**-**name**>**
9. **<**servlet**-class>**web**.**AddEmpServlet**</**servlet**-class>**
10. **</**servlet**>**
11. **<**servlet**-**mapping**>**
12. **<**servlet**-**name**>**addEmpServlet**</**servlet**-**name**>**
13. **<**url**-**pattern**>**/add</url**-**pattern**>**
14. **</**servlet**-**mapping**>**
15. **</**web**-**app**>**

**6 员工管理——使用JDBC添加员工信息**

**6.1 问题**

将客户端提交的员工信息，使用JDBC的技术插入到t\_emp表中。

**6.2 方案**

首先在Oracle中建立一个名字叫t\_emp的表，包括id，name，salary，age四个字段，id为主键字段，并实现自增。在处理表单信息的Servlet中，使用JDBC技术完成连接数据库，并使用获取的请求参数值构建insert语句，执行后完成员工信息的增加操作。操作成功或失败均返回给用户响应的提示信息。

**6.3 步骤**

**步骤一：使用MyEclipse的DB Browser视图连接Oracle数据库**

使用如下步骤打开DB Browser视图

菜单“window”(“Show View”(“Other…”在搜索栏输入DB，选择“DB Browser“后点击ok即可。在视图中右键选择new，创建一个新的连接，配置如图-27，图-28

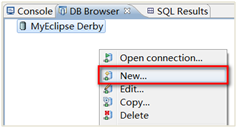


图 - 27

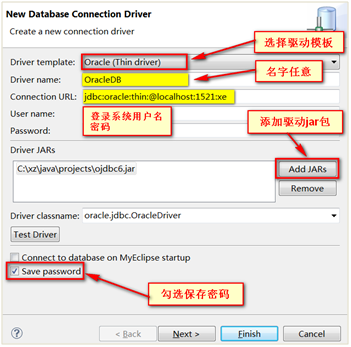


图- 28

连接数据库正确后，点击新创建的这个连接，右键 Open Connection即可查看连接情况如图-29

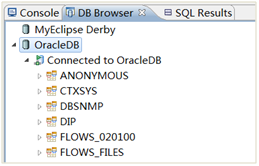
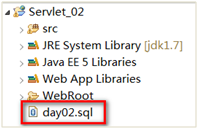


图- 29

**步骤二：执行SQL语句创建t\_emp表**

在工程节点下，右键新建一个名字叫做day02.sql的文件，在该文件中编写建表脚本。如图-30，图-31



图– 30

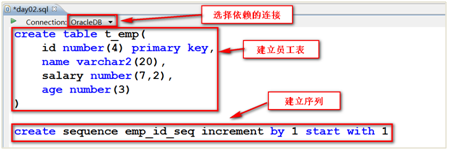
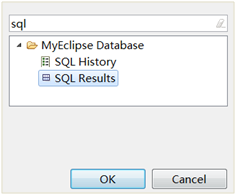


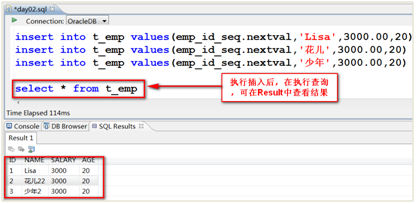
图- 31

在sql文件中，编写完sql语句后，为了执行，可以选中，然后点击连接左边的绿色按钮，即代表执行。为了查看表是否创建，可以在执行完以后到DB Browser视图中，打开对应的连接，查看下面的table节点下是否有更新。

表创建成功之后，继续在sql文件中编写insert语句，选中后执行。为了查看表内的数据，再编写select \* from t\_emp语句。结果的查看可以通过菜单“Window” ( “show View” ( “Other…”，在搜索栏中填写SQL后，选中SQL Result视图，点击确定，即可打开视图看见select语句的执行结果。过程如图-32，图-33所示



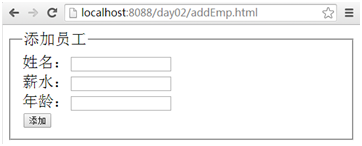
图– 32



图– 33

**步骤三：新建addEmp.html页面**

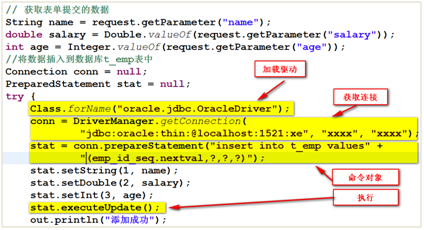
建立addEmp.html页面如下



图– 34

**步骤四：新建AddEmpServlet.java类**

编写有关数据库操作的代码，如图-35



图– 35

在编写JDBC代码时，需要加载驱动jar包，否则Class.forName会报错。在web Project中添加jar包，只需要将文件拷贝，在lib文件夹下右键粘贴即可，工程会自动增加对jar包的构建。

如图-36为添加了驱动jar包后效果

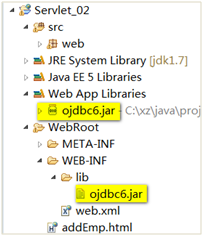


图- 36

**步骤五：添加AddEmpServlet的映射**



图-37

**步骤六：部署应用，访问addEmp.html页面，查看结果**

部署应用，访问addEmp.html页面填写中文数据后点击提交。可以到SQL Result界面中通过执行查询语句确认增加操作是否执行成功。结果参考图-38，图-39



图- 38

**6.4 完整代码**

addEmp.html文件完整代码：

**<**form action**=**"add" method**=**"post"**>**

**<**fieldset**>**

**<**legend**>**添加员工**</**legend**>**

                姓名：**<**input name**=**"name"**/><**br**>**

                薪水：**<**input name**=**"salary"**/><**br**>**

                年龄：**<**input name**=**"age"**/><**br**>**

**<**input type**=**"submit" value**=**"添加"**/>**

**</**fieldset**>**

**</**form**>**

AddEmpServlet.java文件完整代码:

package web**;**

import java**.**io**.**IOException**;**

import java**.**io**.**PrintWriter**;**

import java**.**sql**.**Connection**;**

import java**.**sql**.**DriverManager**;**

import java**.**sql**.**PreparedStatement**;**

import java**.**sql**.**SQLException**;**

import javax**.**servlet**.**ServletException**;**

import javax**.**servlet**.**http**.**HttpServlet**;**

import javax**.**servlet**.**http**.**HttpServletRequest**;**

import javax**.**servlet**.**http**.**HttpServletResponse**;**

**public** **class** AddEmpServlet **extends** HttpServlet **{**

**public** void service**(**HttpServletRequest request**,** HttpServletResponse response**)**

**throws** ServletException**,** IOException **{**

       request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

String name = request.getParameter("name");

double salary = Double.valueOf(request.getParameter("salary"));

int age = Integer.valueOf(request.getParameter("age"));

Connection con=null;

PreparedStatement ps=null;

try {

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql:///db1803","root","root");

ps = con.prepareStatement("insert into t\_emp(name,salary,age) values(?,?,?)");

ps.setString(1, name);

ps.setDouble(2, salary);

ps.setInt(3, age);

ps.executeUpdate();

out.println("添加成功！");

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

if(ps!=null){

try {

ps.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

if(con!=null){

try {

con.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

**}**

**}**

web.xml文件完整代码：

**<?**xml version**=**"1.0" encoding**=**"UTF-8"**?>**

**<**web**-**app version**=**"2.5"

    xmlns**=**"http://java.sun.com/xml/ns/javaee"

    xmlns**:**xsi**=**"http://www.w3.org/2001/XMLSchema-instance"

    xsi**:**schemaLocation**=**"http://java.sun.com/xml/ns/javaee

    http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"**>**

**<**servlet**>**

**<**servlet**-**name**>**addEmpServlet**</**servlet**-**name**>**

**<**servlet**-class>**web**.**AddEmpServlet**</**servlet**-class>**

**</**servlet**>**

**<**servlet**-**mapping**>**

**<**servlet**-**name**>**addEmpServlet**</**servlet**-**name**>**

**<**url**-**pattern**>**/add</url**-**pattern**>**

**</**servlet**-**mapping**>**

**</**web**-**app**>**

**7 员工管理——使用JDBC查询所有员工信息**

**7.1 问题**

客户端访问服务器后可以返回t\_emp表中所有数据的表格展示，效果如图-40：



图- 40

**7.2 方案**

在处理请求的Servlet中，使用JDBC技术实现数据库的访问，并且在遍历结果集时，将结果与html的表格行、列标签进行连接，生成符合表格样式的标记后输出到客户端。

**7.3 步骤**

**步骤一：新建ListEmpServlet.java文件**

在ListEmpServlet类中，使用JDBC技术连接数据库，并执行查询语句，对查询结果集遍历后，以表格形式输出。如图-41

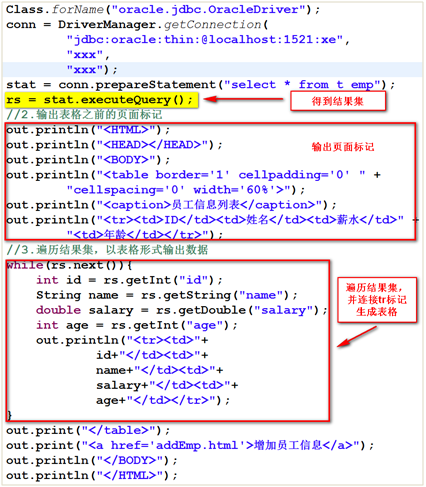


图- 41

**步骤二：添加ListEmpServlet映射**

修改web.xml文件，添加映射，如图-42



图- 42

**步骤三：部署及访问应用**

部署后，在浏览器地址栏输入地址http://localhost:8080/应用名/list,查看结果是否与图-40一致。

**7.4 完整代码**

ListEmpServlet.java文件完整代码：

package web**;**

import java**.**io**.**IOException**;**

import java**.**io**.**PrintWriter**;**

import java**.**sql**.**Connection**;**

import java**.**sql**.**DriverManager**;**

import java**.**sql**.**PreparedStatement**;**

import java**.**sql**.**ResultSet**;**

import java**.**sql**.**SQLException**;**

import javax**.**servlet**.**ServletException**;**

import javax**.**servlet**.**http**.**HttpServlet**;**

import javax**.**servlet**.**http**.**HttpServletRequest**;**

import javax**.**servlet**.**http**.**HttpServletResponse**;**

**public** **class** ListEmpServlet **extends** HttpServlet **{**

**public** void service**(**HttpServletRequest request**,** HttpServletResponse response**)**

**throws** ServletException**,** IOException **{**

        response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

Connection con=null;

PreparedStatement ps=null;

ResultSet rs=null;

try {

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql:///db1803","root","root");

ps = con.prepareStatement("select \* from t\_emp");

rs=ps.executeQuery();

out.println("<table border='1'>");

out.println("员工信息列表");

out.println("<tr><td>id</td><td>姓名</td><td>薪水</td><td>年龄</td></tr>");

while(rs.next()){

int id=rs.getInt("id");

String name = rs.getString("name");

double salary = rs.getDouble("salary");

int age = rs.getInt("age");

out.println("<tr><td>"+id+"</td><td>"+name+"</td><td>"+salary+"</td><td>"+age+"</td></tr>");

out.println("<a href='addEmp.html'>增加员工信息</a>");

}

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

if(ps!=null){

try {

ps.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

if(con!=null){

try {

con.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

**}**

**}**

web.xml文件完整代码：

**<?**xml version**=**"1.0" encoding**=**"UTF-8"**?>**

**<**web**-**app version**=**"2.5"

    xmlns**=**"http://java.sun.com/xml/ns/javaee"

    xmlns**:**xsi**=**"http://www.w3.org/2001/XMLSchema-instance"

    xsi**:**schemaLocation**=**"http://java.sun.com/xml/ns/javaee

    http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"**>**

**<**servlet**>**

**<**servlet**-**name**>**addEmpServlet**</**servlet**-**name**>**

**<**servlet**-class>**web**.**AddEmpServlet**</**servlet**-class>**

**</**servlet**>**

**<**servlet**>**

**<**servlet**-**name**>**listEmpServlet**</**servlet**-**name**>**

**<**servlet**-class>**web**.**ListEmpServlet**</**servlet**-class>**

**</**servlet**>**

**<**servlet**-**mapping**>**

**<**servlet**-**name**>**addEmpServlet**</**servlet**-**name**>**

**<**url**-**pattern**>**/add</url**-**pattern**>**

**</**servlet**-**mapping**>**

**<**servlet**-**mapping**>**

**<**servlet**-**name**>**listEmpServlet**</**servlet**-**name**>**

**<**url**-**pattern**>**/list</url**-**pattern**>**

**</**servlet**-**mapping**>**

**</**web**-**app**>**