

1、解析HttpServletRequest接口

HttpServletRequest模板类中的service方法中需要两个参数对象，分别是HttpServletRequest类型和HttpServletResponse类型的，分别探讨。

--研究: javax.servlet.http.HttpServletRequest接口

1、这是Servlet规范中的一个重要接口

2、继承关系:

```
public interface HttpServletRequest extends ServletRequest{}
```

3、该接口的实现类是web容器负责的，Tomcat服务器有自己的实现。

所以说跟前面讲的几个接口一样，程序员只需要面向接口调用，无需关系具体的实现内容

4、该接口封装的信息:

--封装的是请求协议中的全部内容

请求方式（这就是为什么该接口中有获取请求方式字符串返回的方法）

URL

协议版本号

表单提交的数据

...

5、HttpServletRequest接口类型的变量名一般定义为: request，表示请求。

一个HttpServletRequest对象表示一次请求，100个请求对应该类型对象100个

所以说request对象的生命周期是短暂的

--一次请求: 目前为止，这样理解: 在网页上点击超链接，最终到网页停下来，是一次完整的请求

6、该接口中的常用方法

--对请求中提交的表单数据操作的方法如下，（要注意下面几个方法是继承于ServletRequest接口中的方法）:

表单中提交的数据被封装到request对象中，该对象中的map集合存储这些数据: Map集合中的key是name, value是一个String类型是一维数组

--String getParameter(String name)

该方法通过key获取value一维数组中首元素（通常情况下，这个一维数组只存一个元素，所以该方法使用次数最多）

--String[] getParameterValues(String name)

该方法是通过Map集合中的key获取value（这个就可以获取到value一维数组中的其他元素，使用次数比上面少）

--Map getParameterMap()

该方法获取存储表单提交数据的那个Map集合对象

--Enumeration getParameterNames()

该方法获取Map集合对象中的key，存放到一个Enumeration类型的集合中（该集合前面见到过）

下面是HttpServletRequest接口常用方法中对提交的表单数据操作方法的示例

创建名为idea-servlet-http-8的web项目，配置好相应环境。

编写一个注册表单页面，register.html，设置为欢迎页面

register.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>register page</title>
</head>
<body>
  <form action="/idea-servlet-http-8/query" method="post">
    用户名
    <input type="text" name="username" >
    <br>
    密码
    <input type="password" name="password">
    <br>
    性别
    男<input type="radio" name="sex" value="f">
    女<input type="radio" name="sex" value="m">
    <br>
    爱好
    <input type="checkbox" name="hobby" value="sport">运动
    <input type="checkbox" name="hobby" value="music">音乐
    <input type="checkbox" name="hobby" value="dance">跳舞
    <input type="checkbox" name="hobby" value="study">学习
    <input type="checkbox" name="hobby" value="read">阅读
    <br>
    学历
    <select name="grade">
      <option value="primary">小学</option>
      <option value="high">高中</option>
      <option value="college">大学</option>
    </select>
    <br>
    简历
    <textarea rows="10" cols="60" name="introduce"></textarea>
    <br>
    <input type="submit" value="注册">
    <input type="reset" value="重置">
  </form>
</body>
</html>
```

localhost:8080/idea-servlet-http-8/

应用 百度 淘宝 京东 天猫 苏宁易购 杂七杂八 tensorflow 工具 GitHub java

用户名

密码

性别 男 ☒ 女 ☐

爱好 ☒ 运动 ☒ 音乐 ☐ 跳舞 ☐ 学习 ☐ 阅读

学历

ok

简历

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0">

  <welcome-file-list>
    <welcome-file>注册用户</welcome-file>
    <welcome-file>/register.html</welcome-file>
  </welcome-file-list>

  <servlet>
    <servlet-name>register</servlet-name>
    <servlet-class>com.servlet.httpServletRequest.RequestServlet</servlet-
class>
  </servlet>
  <servlet-mapping>
    <servlet-name>register</servlet-name>
    <url-pattern>/query</url-pattern>
  </servlet-mapping>
</web-app>
```

RequestServlet.java

```
package com.servlet.httpServletRequest;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.Enumeration;
import java.util.Map;
import java.util.Set;

public class RequestServlet extends HttpServlet {
```

```

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

    //获取表单提交的用户信息，这些数据被封装到request对象中
    //用request对象里的方法就可以获取这些信息了
    /*
        表单提交数据的格式如下，用POST请求方式，数据格式如下：

        username=admin&password=123&sex=m&hobby=sport&hobby=music&grade=college&introduce=ok

        而这些信息封装到request对象中实际是存到一个map集合中的，且集合中的value是一个数组类型

        如下：
        Map<String,String[]>
        key          value
        -----
        username      {"admin"}
        password       {"123"}
        sex            {"m"}
        hobby           {"sport","music"}
        grade          {"college"}
        introduce      {"ok"}
    */

    //测试方法: String getParameter(String name)
    String username = request.getParameter("username");
    String password = request.getParameter("password");
    String sex = request.getParameter("sex");
    String hobby = request.getParameter("hobby");
    String grade = request.getParameter("grade");
    String introduce = request.getParameter("introduce");
    System.out.println(username);
    System.out.println(password);
    System.out.println(sex);
    System.out.println(hobby);
    System.out.println(grade);
    System.out.println(introduce);

    System.out.println("=====");

    //测试方法: String[] getParameterValues(String name)
    //这里只有爱好hobby可以选择多个，所以下面只获取表单数据中爱好的数据
    String[] interest = request.getParameterValues("hobby");
    //遍历集合interest
    for (String value:
        interest) {
        System.out.println(value);
    }

    System.out.println("=====");

    //测试方法: Map getParameterMap()
    Map<String,String[]> data = request.getParameterMap();
    //获取data集合中的key并且存放进一个Set集合中
    Set keys1 = data.keySet();
    //遍历keys集合获取data中的value

```

```

        for (Object name:
            keys1) {
            System.out.println(data.get(name));
            String[] values = data.get(name);
            System.out.print(name+" = ");
            for (String value:
                values) {
                System.out.print(value+" ");
            }
            System.out.println();
        }

        System.out.println("=====");
        //从输出结果可以看出，data集合中的value是每个类型数据的数组对象
        //数组的toString方法输出如：[Ljava.lang.String;@7c910271，即[L+完整的类名;@+内存地址经过hash计算转换成16进制

        //测试方法：Enumeration getParameterNames()
        Enumeration names = request.getParameterNames();
        //遍历names集合获取其中Map集合中的value
        while (names.hasMoreElements()){
            String keys2 = (String)names.nextElement();
            String[] values = request.getParameterValues(keys2);
            System.out.print(keys2+" = ");
            for (String value:
                values ) {
                System.out.print(value+" ");
            }
            System.out.println();
        }

    }
}

```

输出如下

```

MeteoAdmin
123456
f
sport
college
ok

=====
sport
music
dance
study
read

=====
[Ljava.lang.String;@65272c20
username = MeteoAdmin
[Ljava.lang.String;@324f44a1
password = 123456
[Ljava.lang.String;@290ca47
sex = f

```

```
[Ljava.lang.String;@22b94d25  
hobby = sport music dance study read  
[Ljava.lang.String;@7e23237e  
grade = college  
[Ljava.lang.String;@36011a57  
introduce = ok
```

```
=====
```

```
username = MeteoAdmin  
password = 123456  
sex = f  
hobby = sport music dance study read  
grade = college  
introduce = ok
```