1. HTTP
2. 概述

Hyper Text Transfer Protocol，超文本传输协议。该协议定义了客户端和服务器端通信的格式。

1. 特点
2. 基于TCP/IP 的高级协议
3. 默认端口号：80
4. 基于请求/响应模型：一次请求对应一次响应
5. 无状态的：每次请求之间相互独立，不能交互数据
6. 请求消息数据格式
7. 请求行

（1）格式

请求方式 请求URL 请求协议/版本

GET /login.html HTTP/1.1

（2）GET和POST的区别：

GET：

请求参数在请求行中，跟在URL之后

请求的URL长度有限制

POST：

请求参数在请求体中

请求的URL长度没有限制

1. 请求头

（1）概述

客户端浏览器通过请求头来告诉服务器一些信息。如浏览器的类型，浏览器能够接收的数据类型…

（2）请求头格式

键值对的形式。

（3）请求头举例

User-Agent:

浏览器告诉服务器，该浏览器的版本信息

Referer:

浏览器告诉服务器，该请求的地址信息

作用：

防盗链，防止其他网站盗用链接

统计作用，统计请求的数量

1. 请求空行

用于分割POST请求的请求头和请求体的

1. 请求体（正文）

封装POST请求消息的请求参数的

1. Request
2. Tomcat中的请求和响应流程
3. Tomcat服务器会根据请求URL中的资源路径，创建对应的Servlet接口的实现类的对象。
4. Tomcat服务器会创建request和response对象，request对象中封装请求消息数据
5. Tomcat将request和response两个对象传递给Servlet中的service方法，并且调用service方法
6. 通过request对象获取请求消息数据，通过response对象设置响应消息数据
7. 服务器在对浏览器作出响应之前，会从response对象中取出响应消息数据
8. request对象继承结构

ServletRequest 接口

| 继承

HttpServletRequest 接口

| 实现

org.apache.catalina.connector.ReuqestFacade类（Tomcat）

由RequestFacade类来实现HttpServletRequest接口，由Tomcat进行编写。

1. request功能
2. 获取请求行数据

方法：

获取请求方式：

String getMethod()

获取虚拟目录：

String getContextPath()

获取Servlet路径：  
String getServletPath()

获取get方式请求参数：

String getQueryString()

获取请求URL：如 http://localhost/demo

StringBuffer getRequestURL()

获取请求URI：如 /demo

String getRequestURI()

获取协议及版本号：  
String getProtocol()

获取客户机IP地址：  
String getRemoteAddr()

1. 获取请求头数据
2. 通过请求头名称获取请求头的值

String getHeader(String name)

1. 获取所有请求头名称

Enumeration<String> getHeaderNames()

1. 使用示例

Enumeration<String> headerNames = request.getHeaderNames();

while(headerNames.hasMoreElements())){

String name = headerNames.nextElement();

String value = request.getHeader(name);

System.out.println(name + “-“ + value);

}

1. 获取请求体数据

只有请求方式为POST时，才有请求体，在请求体中封装了POST请求的请求参数

获取数据步骤：

1. 获取流对象

BufferedReader getReader()

获取字符输入流，仅限字符

ServletInputStream getInputStream()

获取字节输入流，可以操作所有数据类型

1. 获取流对象后，再从该对象中获取数据
2. 使用示例

BufferedReader br = request.getReader();

String line = null;

while((line = br.readLine()) != null)

System.out.println(line);

1. 获取请求参数通用方式
2. 根据参数名称获取参数值

String getParameter(String name);

如：返回

username=a&password=123

1. 根据参数名称获取参数值的数组

String[] getParameterValues(String name)

1. 获取所有请求的参数名称

Enumeration<String> getParameterNames()

1. 获取所有参数的map集合

Map<String, String[]> getParameterMap()

返回的Map的参数为键值对，值为字符串数组，解决了复选框等一个键对应多个值的问题。

Map<String, String[]> parameterMap = request.getParameterMap();

//获取键

Set<String> keyset = parameterMap.keySet();

for(String name : keyset){

//通过键获取值的数组

String[] values = parameterMap.get(name);

System.out.println(name);

for(String value : values)

System.out.println(value);

}

1. 获取请求参数时中文乱码问题
2. get请求方式

Tomcat 8 中已经解决了get请求方式的中文乱码问题，不需要我们自己来解决。

1. Post请求方式

在获取参数时，设置request的编码为utf-8即可：

request.setCharacterEncoding(“utf-8”);

1. Request 的请求转发
2. 步骤

通过 request 对象获取请求转发器对象：

ReqestDispatcher getRequestDispatcher(String path);

使用请求转发器对象RequestDispatcher调用方法forward进行转发：

forward(ServletRequest request, ServletResponse response);

1. 特点

浏览器地址栏路径不变化

只能转发到当前服务器内存资源中

转发并不是两次请求，而只是一次请求

1. Request的共享数据
2. 域对象

一个有作用范围的对象，可以在范围内共享数据

1. request域

代表一次请求的范围，一般用于请求转发的多个资源中共享数据。当我们将请求转发时，可以使用request域共享数据。

1. 方法

存储数据，设置属性

void setAttribute(String name, Object obj)

通过键获取值

Object getAttribute(String name)

通过键移除键值对

void removeAttribute(String name)

1. request域使用

在前一个Servlet中存储数据到request域中：

request.setAttribute(“msg”, “hello”);

在转发的Servlet中获取request域中的数据：

Object msg = request.getAttribute(“msg”);

1. 用户登录实例
2. 需求分析
3. 编写login.html登录页面
4. 使用Druid数据库连接池技术，操作mysql
5. 使用JdbcTemplate技术封装JDBC
6. 登录成功时跳转到SuccessServlet展示：登录成功！用户名，欢迎你！
7. 登录失败时跳转到FailServlet展示:登录失败，用户名或密码错误
8. 具体思路
9. 点击登录按钮后，逻辑转到LoginServlet中，在LoginServlet中，我们需要设置编码以避免中文情况，获取username和password的内容，并将username和password封装为一个User对象。调用数据库查询的类Userdao的login方法查询，获取返回值User对象。判断返回值是否为null，如果是，则为登录失败，转到FailServlet中；如果不是，则登录成功，将用户信息存到request域中，转发给SuccessServlet。
10. 数据库查询类Userdao的login方法通过User对象进行查询，执行sql语句。
11. 具体操作
12. 创建项目，导入HTML页面，druid配置文件，以及依赖的Jar包
13. 创建数据库环境，创建对应的数据库，表
14. 创建对应的用户类User
15. 创建操作数据库的类UserDao
16. 创建JDBC的工具类JDBCUtils，创建连接池，链接数据库
17. 创建测试类UserDaoTest，测试UserDao中的login方法是否可行。

7， 编写LoginServlet类，重写doGet()，doPost()方法，在doGet()方法中写入逻辑。

1. html中的form表单的action路径的写法：虚拟目录+Servlet的资源路径
2. 编写FailServlet类和SuccessServlet类，分别对应于登录失败和登录成功的页面。
3. BeabUtils
4. 概述

BeabUtils工具类，用于简化数据的封装。当我们需要将多个数据整合为一个对象时，可以使用BeanUtils工具类，直接封装为对象。

1. 使用

Map<String, String[]> map = req.getParameterrMap();

User loginUser = new User();

//整体封装数据

BeanUtils.populate(loginUser, map);

1. JavaBean
2. 概述

JavaBean为标准的Java类。

1. 要求
2. 类必须为public修饰
3. 必须提供空参的构造器
4. 成员变量必须使用private修饰
5. 提供公共setter和getter方法
6. 方法
7. setProperty

设置属性，所谓的属性不是成员变量，属性是通过get和set方法获取的。setProperty()方法设置的也是属性的值，而不是成员变量的值。

1. getProperty

获取属性的值。

1. populate

popilate(Object obj, Map map)，将map集合的键值对信息，封装到对应的JavaBean对象中来。