JavaWeb 案例DAY02

1. 创建一个Servlet,理解Servlet的生命周期

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

直接使用案例1项目

步骤二：定义一个类，实现Servlet接口

在包com.bailiban.servlet中定义一个类Demo2Servlet 实现Servlet接口,并重写Servlet生命周期相关方法

|  |
| --- |
| **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** javax.servlet.Servlet;  **import** javax.servlet.ServletConfig;  **import** javax.servlet.ServletException;  **import** javax.servlet.ServletRequest;  **import** javax.servlet.ServletResponse;  /\*  \* Servlet的生命周期  \*/  **public** **class** Demo2Servlet **implements** Servlet {  /\*  \* 初始化方法  \* 在servlet创建时执行，只会执行一次  \*/  @Override  **public** **void** init(ServletConfig config) **throws** ServletException {    System.***out***.println("我出生了...");    }  /\*  \* 提供服务的方法 ，每访问一次servlet就会执行一次 多次执行  \*/  @Override  **public** **void** service(ServletRequest req, ServletResponse res) **throws** ServletException, IOException {  System.***out***.println("我响应请求了...");  }  /\*  \* 销毁方法  \* 在服务器正常关闭执行 只执行一次  \*/  @Override  **public** **void** destroy() {  System.***out***.println("我挂了... ");  }  @Override  **public** ServletConfig getServletConfig() {  // **TODO** Auto-generated method stub  **return** **null**;  }  @Override  **public** String getServletInfo() {  // **TODO** Auto-generated method stub  **return** **null**;  }  } |

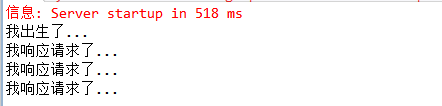
步骤三：在web.xml中配置Servlet

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* version=*"3.1"*>  <display-name>hello</display-name>  <welcome-file-list>  <welcome-file>index.html</welcome-file>  <welcome-file>index.htm</welcome-file>  <welcome-file>index.jsp</welcome-file>  <welcome-file>default.html</welcome-file>  <welcome-file>default.htm</welcome-file>  <welcome-file>default.jsp</welcome-file>  </welcome-file-list>  <!-- 配置Servlet-->  <servlet>  <!-- servlet的名字 自己取得 -->  <servlet-name>demo2</servlet-name>  <!-- servlet类全限定名 -->  <servlet-class>com.bailiban.servlet.Demo2Servlet</servlet-class>  </servlet>  <!-- 配置Servlet访问路径 -->  <servlet-mapping>  <servlet-name>demo2</servlet-name>  <!-- 配置访问路径 -->  <url-pattern>/demo2</url-pattern>  </servlet-mapping>  </web-app> |

步骤四：浏览器访问

在浏览器地址输入：[http://localhost:8080/hello/demo2](http://localhost:8080/hello/demo1)

查看生命周期方法执行特点



正常关闭服务器查看destroy()方法执行



1. 分别以继承GenericServlet和HttpServlet方式 创建Servlet

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

创建项目servlet\_demo,项目使用Servlet3.0注解方式创建Servlet

步骤二：定义一个类，继承GenericServlet抽象类

在包com.bailiban.servlet中定义一个类Demo1Servlet,继承GenericServlet抽象类,重写service方法，其他方法不用重写

|  |
| --- |
| **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** javax.servlet.GenericServlet;  **import** javax.servlet.ServletException;  **import** javax.servlet.ServletRequest;  **import** javax.servlet.ServletResponse;  **import** javax.servlet.annotation.WebServlet;  /\*  \* 继承 GenericServlet 方式 创建Servlet  \*/  @WebServlet("/demo1")  **public** **class** Demo1Servlet **extends** GenericServlet {  @Override  **public** **void** service(ServletRequest req, ServletResponse res) **throws** ServletException, IOException {  System.***out***.println("继承 GenericServlet 方式 创建Servlet");    }  } |

步骤三：浏览器访问

在浏览器地址输入：<http://localhost:8080/servlet_demo/demo1>

控制台打印



步骤四：定义一个类，继承HttpServlet抽象类

在包com.bailiban.servlet中定义一个类Demo2Servlet,继承HttpServlet抽象类,重写doGet方法 和doPost方法，其他方法不用重写

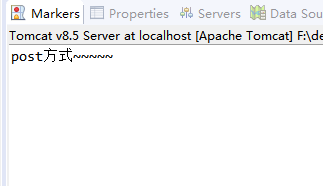
|  |
| --- |
| **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  /\*\*  \* 继承 HttpServlet方式创建servlet  \*/  @WebServlet("/demo2")  **public** **class** Demo2Servlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  //如果请求方式为get执行该方法  @Override  **protected** **void** doGet(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {    System.***out***.println("get方式~~~~~");  }  //如果请求方式为post执行该方法  @Override  **protected** **void** doPost(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  System.***out***.println("post方式~~~~~");  }  } |

步骤五：在web项目根目录创建demo2.html文件

该文件中包含一个表单，表单的提交地址为该servlet, 提交方式为post

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>demo2</title>  </head>  <body>  <form action=*"/servlet\_demo/demo2"* method=*"post"*>  用户名: <input type=*"text"* name = *"username"* /><br/>  <input type=*"submit"* value=*"提交"* />    </form>  </body>  </html> |

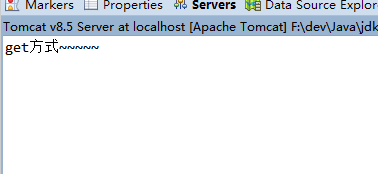
将项目部署到Tomcat并启动，提交表单执行该Servlet的doPost方法



交表单的提交方式改为get

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>demo2</title>  </head>  <body>  <form action=*"/servlet\_demo/demo2"* method=*"get"*>  用户名: <input type=*"text"* name = *"username"* /><br/>  <input type=*"submit"* value=*"提交"* />    </form>  </body>  </html> |

提交表单执行该Servlet的doGet方法



1. 完成用户登录综合案例

需求：

1.编写login.html登录页面

2.用户输入用户名和密码，将数据提交到LoginServlet

3.在LoginServlet中接收用户名和密码

登录成功重定向到SuccessServlet 显示欢迎信息

登录失败将错误信息转发到ErrorServlet 显示

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

创建项目login\_demo

步骤二：编写login.html登录页面

该登录界面的表单中包含用户名和密码两个输入框

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>登录界面</title>  </head>  <body>  <form action=*""* method=*"post"*>  用户名: <input type=*"text"* placeholder=*"请输入用户名"* name = *"username"* /><br/>  密码： <input type=*"password"* placeholder=*"请输入用密码"* name = *"password"* /><br/>  <input type=*"submit"* value=*"登录"* />    </form>  </body>  </html> |

步骤三：创建LoginServlet

在包com.bailiban.servlet中创建LoginServlet,设置访问路径为/login

|  |
| --- |
| **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  /\*\*  \* Servlet implementation class LoginServlet  \*/  @WebServlet("/login")  **public** **class** LoginServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** LoginServlet() {  **super**();  }    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {        }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  //无论是get还是post请求 都执行doGet方法 避免写重复的代码  doGet(request, response);  }  } |

将login.html表单的action属性设置为

<form action=*"/login\_demo/login"* method=*"post"*>

完善doGet方法

|  |
| --- |
| **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //使用request对象解决 post 提交中文乱码  request.setCharacterEncoding("utf-8");  //使用request对象 接收表单提交的数据  //getParameter(String name) name为表单空件的name属性值  String username = request.getParameter("username");  String password = request.getParameter("password");  // System.out.println(username+"--"+password);  //判定用户名和密码是否正确  **if**("admin".equals(username)&&"svse".equals(password)) {  //如果登录成功 重定向到 SuccessServlet  //request.getContextPath() 获取虚拟目录  response.sendRedirect(request.getContextPath()+"/success");    }**else** {  //出现错误 向request域中放错误信息  request.setAttribute("msg", "用户名为："+username+" 密码为:"+password+" 的用户登录失败");  //转发到ErrorServlet 显示错误提示  request.getRequestDispatcher("/error").forward(request, response);    }    } |

步骤四：创建SuccessServlet和ErrorServlet

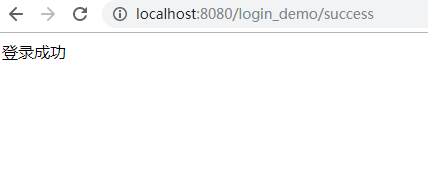
在包com.bailiban.servlet分别创建SuccessServlet和ErrorServlet

|  |
| --- |
| **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  /\*\*  \* Servlet implementation class SuccessServlet  \*/  @WebServlet("/success")  **public** **class** SuccessServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;    **public** SuccessServlet() {  **super**();  // **TODO** Auto-generated constructor stub  }  **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //解决输出乱码  response.setContentType("text/html;charset=utf-8");  //向浏览器输出错误信息  response.getWriter().write("登录成功");  }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  // **TODO** Auto-generated method stub  doGet(request, response);  }  }  **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  /\*\*  \* Servlet implementation class ErrorServlet  \*/  @WebServlet("/error")  **public** **class** ErrorServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;      **public** ErrorServlet() {  **super**();  // **TODO** Auto-generated constructor stub  }    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //解决输出乱码  response.setContentType("text/html;charset=utf-8");  //从 request域中 取出 错误信息  String msg = (String) request.getAttribute("msg");  //向浏览器输出错误信息  response.getWriter().write(msg);  }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  // **TODO** Auto-generated method stub  doGet(request, response);  }  } |

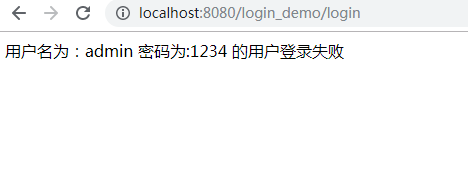
步骤五：浏览器访问

在浏览器地址输入：http://localhost:8080/login\_demo/login.html

登录成功跳转



登录失败 转发



1. 企业应用：完善用户登录综合案例

需求：

在案例5的基础上，完善用户登录，根据输入的用户名和密码在数据库中查找对应的用户，如果用户存在登录成功，如果用户不存在登录失败，项目采用三层结构设计

* 参考答案

步骤一：创建数据库环境

在mysql数据库中创建名为javaweb\_day02的数据库， 执行下面的SQL语句创建用户表，初始化数据

|  |
| --- |
| DROP TABLE IF EXISTS t\_user ;  CREATE TABLE t\_user(    id INT PRIMARY KEY AUTO\_INCREMENT,  username VARCHAR(32) UNIQUE NOT NULL,  PASSWORD VARCHAR(32) NOT NULL  );  INSERT INTO t\_user VALUES (NULL ,'wangcai','888');  INSERT INTO t\_user VALUES (NULL ,'ruhua','666');  INSERT INTO t\_user VALUES (NULL ,'cuihua','555'); |

步骤二：创建实体类 User

创建包com.bailiban.domain,在该包下创建User类

|  |
| --- |
| **package** com.bailiban.domain;  **public** **class** User {  **private** **int** id; //用户编号  **private** String username;//用户名  **private** String password; //密码      **public** User() {  **super**();  }    **public** User(String username, String password) {  **super**();  **this**.username = username;  **this**.password = password;  }  **public** **int** getId() {  **return** id;  }  **public** **void** setId(**int** id) {  **this**.id = id;  }  **public** String getUsername() {  **return** username;  }  **public** **void** setUsername(String username) {  **this**.username = username;  }  **public** String getPassword() {  **return** password;  }  **public** **void** setPassword(String password) {  **this**.password = password;  }      } |

步骤三：编写工具类JDBCUtils

创建包com.bailiban.utils,在该包下创建JDBCUtils类

|  |
| --- |
| **package** com.bailiban.utils;  **import** java.io.IOException;  **import** java.io.InputStream;  **import** java.sql.Connection;  **import** java.sql.SQLException;  **import** java.util.Properties;  **import** javax.sql.DataSource;  **import** com.alibaba.druid.pool.DruidDataSourceFactory;  /\*\*  \* JDBC工具类 使用Durid连接池  \*/  **public** **class** JDBCUtils {  **private** **static** DataSource *ds*;  // 静态代码块  **static** {  **try** {  // 1.加载配置文件  Properties pro = **new** Properties();  // 使用ClassLoader加载配置文件，获取字节输入流  InputStream is = JDBCUtils.**class**.getClassLoader().getResourceAsStream("db.properties");  pro.load(is);  // 2.初始化连接池对象  *ds* = DruidDataSourceFactory.*createDataSource*(pro);  } **catch** (IOException e) {  e.printStackTrace();  } **catch** (Exception e) {  e.printStackTrace();  }  }  /\*\*  \* 获取连接池对象  \*/  **public** **static** DataSource getDataSource() {  **return** *ds*;  }  } |

步骤四：编写dao层

创建包com.bailiban.dao,在该包下创建UserDao接口

|  |
| --- |
| **package** com.bailiban.dao;  **import** com.bailiban.domain.User;  **public** **interface** UserDao {  /\*\*  \* 登录方法  \* **@param** loginUser 只有用户名和密码  \* **@return** user包含用户全部数据,没有查询到，返回null  \*/  User login(User loginUser);  } |

创建包com.bailiban.dao.impl,在该包下创建UserDaoImpl实现类

|  |
| --- |
| **package** com.bailiban.dao.impl;  **import** org.springframework.dao.DataAccessException;  **import** org.springframework.jdbc.core.BeanPropertyRowMapper;  **import** org.springframework.jdbc.core.JdbcTemplate;  **import** com.bailiban.dao.UserDao;  **import** com.bailiban.domain.User;  **import** com.bailiban.utils.JDBCUtils;  /\*\*  \* 操作数据库中User表的类  \*/  **public** **class** UserDaoImpl **implements** UserDao {  //声明JDBCTemplate对象共用  **private** JdbcTemplate template = **new** JdbcTemplate(JDBCUtils.*getDataSource*());  /\*\*  \* 登录方法  \* **@param** loginUser 只有用户名和密码  \* **@return** user包含用户全部数据,没有查询到，返回null  \*/  @Override  **public** User login(User loginUser){  **try** {  //1.编写sql  String sql = "select \* from t\_user where username = ? and password = ?";  //2.调用query方法  User user = template.queryForObject(sql,  **new** BeanPropertyRowMapper<User>(User.**class**),  loginUser.getUsername(), loginUser.getPassword());  **return** user;  } **catch** (DataAccessException e) {  //打印调试信息  e.printStackTrace();  **return** **null**;  }  }  } |

步骤五：编写service层

创建包com.bailiban.service,在该包下创建UserService接口

|  |
| --- |
| **package** com.bailiban.service;  **import** com.bailiban.domain.User;  **public** **interface** UserService {  /\*\*  \* 登录方法  \* **@param** loginUser 只有用户名和密码  \* **@return** user包含用户全部数据,没有查询到，返回null  \*/  User login(User loginUser);  } |

创建包com.bailiban.service.impl,在该包下创建UserServiceImpl实现类

|  |
| --- |
| **package** com.bailiban.service.impl;  **import** com.bailiban.dao.UserDao;  **import** com.bailiban.dao.impl.UserDaoImpl;  **import** com.bailiban.domain.User;  **import** com.bailiban.service.UserService;  **public** **class** UserServiceImpl **implements** UserService {    //依赖dao  **private** UserDao userDao = **new** UserDaoImpl();  @Override  **public** User login(User loginUser) {  **return** userDao.login(loginUser);  }  } |

步骤六：编写LoginServlet中的代码

|  |
| --- |
| **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** com.bailiban.domain.User;  **import** com.bailiban.service.UserService;  **import** com.bailiban.service.impl.UserServiceImpl;  /\*\*  \* Servlet implementation class LoginServlet  \*/  @WebServlet("/login")  **public** **class** LoginServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** LoginServlet() {  **super**();  }    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //使用request对象解决 post 提交中文乱码  request.setCharacterEncoding("utf-8");  //使用request对象 接收表单提交的数据  //getParameter(String name) name为表单空件的name属性值  String username = request.getParameter("username");  String password = request.getParameter("password");  //接收到的数据封装进用户对象  User loginUser = **new** User(username, password);  //实例化UserService对象  UserService userService = **new** UserServiceImpl();  //调用登录方法 得到从数据库中查询的用户对象  User user = userService.login(loginUser);  //判断用户是否登录成功  **if**(user!=**null**) {  //如果登录成功 重定向到 SuccessServlet  //request.getContextPath() 获取虚拟目录  response.sendRedirect(request.getContextPath()+"/success");    }**else** {  //出现错误 向request域中放错误信息  request.setAttribute("msg", "用户名为："+username+" 密码为:"+password+" 的用户登录失败");  //转发到ErrorServlet 显示错误提示  request.getRequestDispatcher("/error").forward(request, response);    }    }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  //无论是get还是post请求 都执行doGet方法 避免写重复的代码  doGet(request, response);  }  } |

我们发现封装数据比较麻烦 我们可以使用commons-beanutils-1.8.0.jar包中的BeanUtils 简化数据的封装,再次优化LoginServlet中的代码

|  |
| --- |
| **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** java.lang.reflect.InvocationTargetException;  **import** java.util.Map;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** org.apache.commons.beanutils.BeanUtils;  **import** com.bailiban.domain.User;  **import** com.bailiban.service.UserService;  **import** com.bailiban.service.impl.UserServiceImpl;  /\*\*  \* Servlet implementation class LoginServlet  \*/  @WebServlet("/login")  **public** **class** LoginServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** LoginServlet() {  **super**();  }    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //使用request对象解决 post 提交中文乱码  request.setCharacterEncoding("utf-8");  // //使用request对象 接收表单提交的数据  // //getParameter(String name) name为表单空件的name属性值  // String username = request.getParameter("username");  // String password = request.getParameter("password");  // //接收到的数据封装进用户对象  // User loginUser = new User(username, password);  //使用BeanUtils优化参数的封装  //获取所有请求参数  Map<String, String[]> map = request.getParameterMap();  //将参数数据封装进实体类  User loginUser = **new** User();  **try** {  BeanUtils.*populate*(loginUser, map);  } **catch** (Exception e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  //实例化UserService对象  UserService userService = **new** UserServiceImpl();  //调用登录方法 得到从数据库中查询的用户对象  User user = userService.login(loginUser);  //判断用户是否登录成功  **if**(user!=**null**) {  //如果登录成功 重定向到 SuccessServlet  //request.getContextPath() 获取虚拟目录  response.sendRedirect(request.getContextPath()+"/success");    }**else** {  //出现错误 向request域中放错误信息  request.setAttribute("msg", "用户名为："+loginUser.getUsername()+" 密码为:"+loginUser.getPassword()+" 的用户登录失败");  //转发到ErrorServlet 显示错误提示  request.getRequestDispatcher("/error").forward(request, response);    }    }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  //无论是get还是post请求 都执行doGet方法 避免写重复的代码  doGet(request, response);  }  } |