REPORT



|  |  |
| --- | --- |
| 과 목 : | 데이터베이스이론및실습 |
| 제출일자 : | 2024.06.04 |
| 담당교수 : | 권기현 |
| 학 과 : | 정보통신공학과 |
| 학 번 : | 201920479 |
| 이 름 : | 김행복 |

7-1 계산기 구현

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>계산기-useBean</title>  </head>  <body>  <h2>계산기-useBean</h2>  <hr>  <form method="post" action="calc.jsp">  <input type="text" name="n1" size="10"><select name ="op">  <option> + </option>  <option> - </option>  <option> \* </option>  <option> / </option>  </select>  <input type="text" name="n2" size="10">  <input type="submit" value="실행">  </form>  </body>  </html> |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <jsp:useBean id="calc" class="ch07.Calculator" />  <jsp:setProperty name="calc" property= "\*" />  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>Insert title here</title>  </head>  <body>  <h2>계산 결과 - JSP</h2>  <hr>  결과:<%=calc.calc()%>  </body>  </html> |

실행 결과

텍스트, 폰트, 화이트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

7-2 JSTL과 EL 종합 예제

|  |
| --- |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>JSTL 종합 예제</title>  </head>  <body>  <h2>JSTL 종합 예제</h2>    <hr>  <h3>set, out</h3>  <c:set var="product1" value="<h2>애플 아이폰</h2>" />  <c:set var="product2" value="삼성 갤럭시 노트" />  <%-- <c:set var="intArray" value="${[1, 2, 3, 4, 5]}" /> --%>  <c:set var="intArray" value="<%= new int[]{1,2,3,4,5} %>" />  <p>  product1(jstl):  <c:out value="${product1}" default="Not registerd" escapeXml="true" />  </p>  <p>product1(el):${product1}</p>  <p>intArray[2]: ${intArray[2]}</p>  <hr>  <h3>forEach: 배열 출력</h3>  <ul>  <c:forEach var="num" varStatus="i" items="${intArray}">  <li>${i.index} : ${num}</li>  </c:forEach>  </ul>  <hr>    <h3>if</h3>  <c:set var="checkout" value="true" />  <c:if test="${checkout}">  <p>주문 제품: ${product2}</p>  </c:if>  <c:if test="${!checkout}">  <p>주문 제품이 아님!!</p>  </c:if>    <c:if test="${!empty product2}">  <p>  <b>${product2} 이미 추가됨!!</b>  </p>  </c:if>  <hr>  <h3>choose, when, otherwise</h3>  <c:choose>  <c:when test="${checkout}">  <p>주문 제품: ${product2}</p>  </c:when>  <c:otherwise>  <p>주문 제품이 아님!!</p>  </c:otherwise>  </c:choose>  <hr>    <h3>forTokens</h3>  <c:forTokens var="city" items="Seoul|Tokyo|New York|Toronto" delims="|" varStatus="i">  <c:if test="${i.first }">도시 목록 : </c:if>  ${city}  <c:if test="${!i.last }">, </c:if>  </c:forTokens>    <hr>  </body>  </html> |

실행결과

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

8-1 계산기 구현

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>계산기-컨트롤러</title>  </head>  <body>  <h2>계산기-컨트롤러</h2>  <hr>  <form method="post" action="/jwbook/calcControl">  <input type="text" name="n1" size="10"><select name ="op">  <option> + </option>  <option> - </option>  <option> \* </option>  <option> / </option>  </select>  <input type="text" name="n2" size="10">  <input type="submit" value="실행">  </form>  </body>  </html> |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <jsp:useBean id="calc" class="ch07.Calculator" />  <jsp:setProperty name="calc" property= "\*" />  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>계산기-컨트롤러</title>  </head>  <body>  <h2>계산 결과-컨트롤러</h2>  <hr>  결과: ${result}  </body>  </html> |
| package ch08;  import java.io.IOException;  import java.io.PrintWriter;  import javax.servlet.ServletException;  import javax.servlet.annotation.WebServlet;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  @WebServlet("/calcControl")  public class Calculator extends HttpServlet {  private static final long serialVersionUID = 1L;    public Calculator() {  super();  }  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  int n1 = Integer.parseInt(request.getParameter("n1"));  int n2 = Integer.parseInt(request.getParameter("n2"));  String op = request.getParameter("op");    long result = 0;    switch(request.getParameter("op")) {  case "+" : result = n1+n2; break;  case "-" : result = n1-n2; break;  case "/" : result = n1/n2; break;  case "\*" : result = n1\*n2; break;  }    request.setAttribute("result", result);  getServletContext().getRequestDispatcher("/ch08/calcResult.jsp").forward(request, response);  }  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  doGet(request, response);  }  } |

실행 결과

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 화이트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

8-2 고급 컨트롤러 서블릿 구현

|  |
| --- |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>상품정보 조회</title>  </head>  <body>  <h2>상품정보 조회</h2>  <hr>  <ul>  <li>상품코드: ${p.id }</li>  <li>상품명: ${p.name }</li>  <li>제조사: ${p.maker }</li>  <li>가격: ${p.price }</li>  <li>등록일: ${p.date }</li>  </ul>  </body>  </html> |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>상품 목록</title>  </head>  <body>  <h2>상품 목록</h2>  <hr>  <table border="1">  <tr> <th>번호</th> <th>상품명</th> <th>가격</th> </tr>  <c:forEach var="p" varStatus="i" items="${products}">  <tr>  <td>${i.count}</td>  <td><a href="/jwbook/pcontrol?action=info&id=${p.id}">${p.name}</a></td>  <td>${p.price}</td>  </tr>  </c:forEach>  </table>  </body>  </html> |
| package ch08;  public class Product {  private String id;  private String name;  private String maker;  private int price;  private String date;    public Product(String id, String name, String maker, int price, String date)  {  this.id = id;  this.name = name;  this.maker = maker;  this.price = price;  this.date = date;  }    public String getId() {return id;}  public void setId(String id) {this.id = id;}    public String getName() {return name;}  public void setName(String name) {this.name = name;}    public String getMaker() {return maker;}  public void setMaker(String maker) {this.maker = maker;}    public int getPrice() {return price;}  public void setPrice(int price) {this.price = price;}    public String getDate() {return date;}  public void setDate(String date) {this.date = date;}    } |
| package ch08;  import java.io.IOException;  import javax.servlet.ServletConfig;  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 ProductController  \*/  @WebServlet("/pcontrol")  public class ProductController extends HttpServlet {  private static final long serialVersionUID = 1L;  ProductService service;  /\*\*  \* @see HttpServlet#HttpServlet()  \*/  public void init(ServletConfig config) throws ServletException {  super.init(config);  service = new ProductService();  }    /\*\*  \* @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)  \*/  protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  // TODO Auto-generated method stub  String action = request.getParameter("action");  String view ="";    if(action == null) {  getServletContext().getRequestDispatcher("/pcontrol?action=list").forward(request, response);  } else {  switch(action) {  case "list": view = list(request, response); break;  case "info": view = info(request, response); break;  }  getServletContext().getRequestDispatcher("/ch08/" + view).forward(request, response);  }  }  private String info(HttpServletRequest request, HttpServletResponse response) {  request.setAttribute("p", service.find(request.getParameter("id")));  return "productInfo.jsp";  }  private String list(HttpServletRequest request, HttpServletResponse response) {  request.setAttribute("products", service.findAll());  return "productList.jsp";  }  } |
| package ch08;  import java.util.ArrayList;  import java.util.HashMap;  import java.util.List;  import java.util.Map;  public class ProductService {  Map<String, Product> products = new HashMap<>();    public ProductService() {  Product p = new Product("101", "애플 사과폰 12", "애플전자", 1200000, "2020.12.12");  products.put("101", p);    p = new Product("102", "삼전 우주폰 21", "삼전전자", 1300000, "2021.2.2");  products.put("102", p);    p = new Product("103", "엘스 듀얼폰", "엘스전자", 1500000, "2021.3.2");  products.put("103", p);  }    public List<Product> findAll() {  return new ArrayList<>(products.values());  }    public Product find(String id) {  return products.get(id);  }  } |

실행 결과

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

9-2 JDBC 종합 실습: 학생 정보 조회와 등록

|  |
| --- |
| package ch09;  import java.sql.Date;  public class Student {  private int id;  private String username;  private String univ;  private Date birth;  private String email;    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 getUniv() {  return univ;  }  public void setUniv(String univ) {  this.univ = univ;  }  public Date getBirth() {  return birth;  }  public void setBirth(Date birth) {  this.birth = birth;  }  public String getEmail() {  return email;  }  public void setEmail(String email) {  this.email = email;  }  } |
| package ch09;  import java.io.IOException;  import javax.servlet.ServletConfig;  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;  @WebServlet("/studentControl")  public class studentControl extends HttpServlet {  private static final long serialVersionUID = 1L;    StudentDAO dao;    public void init(ServletConfig config) throws ServletException {  super.init(config);  dao = new StudentDAO();  }    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  request.setCharacterEncoding("utf-8");  String action = request.getParameter("action");  String view ="";    if(action == null) {  getServletContext().getRequestDispatcher("/studentControl?action=list").forward(request, response);  } else {  switch(action) {  case "list": view = list(request, response); break;  case "insert": view = insert(request, response); break;  }  getServletContext().getRequestDispatcher("/ch09/" + view).forward(request, response);  }  }    private String list(HttpServletRequest request, HttpServletResponse response) {  request.setAttribute("students", dao.getAll());  return "studentInfo.jsp";  }  public String insert(HttpServletRequest request, HttpServletResponse response) {  Student s = new Student();  try {  BeanUtils.populate(s, request.getParameterMap());  } catch (Exception e) {  e.printStackTrace();  }  dao.insert(s);  return list(request, response);  }  } |
| package ch09;  import java.sql.\*;  import java.util.ArrayList;  import java.util.List;  public class StudentDAO {  Connection conn = null;  PreparedStatement pstmt;    final String JDBC\_DRIVER = "oracle.jdbc.OracleDriver";  final String JDBC\_URL = "jdbc:oracle:thin:@localhost:1521:xe";    public void open() {  try {  Class.forName(JDBC\_DRIVER);  conn = DriverManager.getConnection(JDBC\_URL, "system", "pass");  } catch (Exception e) {  e.printStackTrace();  }  }    public void close() {  try {  pstmt.close();  conn.close();  } catch (SQLException e) {  e.printStackTrace();  }  }    public void insert(Student s) {  open();  String sql = "INSERT INTO student(id, username, univ, birth, email) values(student\_seq.NEXTVAL, ?,?,?,?)";    try {  pstmt = conn.prepareStatement(sql);  pstmt.setString(1, s.getUsername());  pstmt.setString(2, s.getUniv());  pstmt.setDate(3, s.getBirth());  pstmt.setString(4, s.getEmail());    pstmt.executeUpdate();  } catch(Exception e) {  e.printStackTrace();  } finally {  close();  }  }    public List<Student> getAll() {  open();  List<Student> students = new ArrayList<>();    try {  pstmt = conn.prepareStatement("select \* from student");  ResultSet rs = pstmt.executeQuery();    while(rs.next()) {  Student s = new Student();  s.setId(rs.getInt("id"));  s.setUsername(rs.getString("username"));  s.setUniv(rs.getString("univ"));  s.setBirth(rs.getDate("birth"));  s.setEmail(rs.getString("email"));    students.add(s);  }  } catch (Exception e) {  e.printStackTrace();  } finally {  close();  }  return students;  }  } |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>    <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>학생 정보</title>  </head>  <body>  <h2>학생정보</h2>[<a href="/jwbook/studentControl">새로고침</a>]  <hr>  <table border = "1">  <tr>  <th>id</th>  <th>이름</th>  <th>대학</th>  <th>생일</th>  <th>이메일</th>  </tr>  <c:forEach items = "${students}" var="s">  <tr>  <td>${s.id }</td>  <td>${s.username }</td>  <td>${s.univ }</td>  <td>${s.birth }</td>  <td>${s.email }</td>  </tr>  </c:forEach>  </table>  <hr>  <h2>학생 추가</h2>  <hr>  <form method="post" action="/jwbook/studentControl?action=insert">  <label>이름</label>  <input type="text" name="username"><br>  <label>대학</label>  <input type="text" name="univ"><br>  <label>생일</label>  <input type="text" name="birth"><br>  <label>이메일</label>  <input type="text" name="email"><br>  <button type="submit">등록</button>  </form>  </body>  </html> |

실행 결과

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

10-1 뉴스 기사 관리 웹 서비스

|  |
| --- |
| package ch10;  public class News {  private int aid;  private String title;  private String img;  private String ndate;  private String content;    public int getAid() {  return aid;  }  public void setAid(int aid) {  this.aid = aid;  }  public String getTitle() {  return title;  }  public void setTitle(String title) {  this.title = title;  }  public String getImg() {  return img;  }  public void setImg(String img) {  this.img = img;  }  public String getNdate() {  return ndate;  }  public void setNdate(String ndate) {  this.ndate = ndate;  }  public String getContent() {  return content;  }  public void setContent(String content) {  this.content = content;  }  } |
| package ch10;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  public class NewsDAO {  Connection conn = null;  PreparedStatement pstmt;    final String JDBC\_DRIVER = "oracle.jdbc.OracleDriver";  final String JDBC\_URL = "jdbc:oracle:thin:@localhost:1521:xe";    public Connection open() {  try {  Class.forName(JDBC\_DRIVER);  conn = DriverManager.getConnection(JDBC\_URL, "system", "pass");  } catch (Exception e) {  e.printStackTrace();  }  return conn;  }    public void close() {  try {  pstmt.close();  conn.close();  } catch (SQLException e) {  e.printStackTrace();  }  }    public void addNews(News n) throws SQLException {  Connection conn = open();  String sql = "INSERT INTO news (aid, title, img, ndate, content) VALUES (news\_seq.NEXTVAL, ?, ?, CURRENT\_TIMESTAMP, ?)";  PreparedStatement pstmt = conn.prepareStatement(sql);    try(conn; pstmt) {  pstmt.setString(1, n.getTitle());  pstmt.setString(2, n.getImg());  pstmt.setString(3, n.getContent());  pstmt.executeUpdate();  } catch(Exception e) {  e.printStackTrace();  } finally {  close();  }  }    public List<News> getAll() throws Exception {  Connection conn = open();  List<News> newsList = new ArrayList<>();    String sql = "SELECT aid, title, TO\_TIMESTAMP(ndate, 'YYYY-MM-DD HH24:MI:SS.FF') AS cdate FROM news";  PreparedStatement pstmt = conn.prepareStatement(sql);  ResultSet rs = pstmt.executeQuery();    try(conn; pstmt; rs) {  while(rs.next()) {  News n = new News();  n.setAid(rs.getInt("aid"));  n.setTitle(rs.getString("title"));  n.setNdate(rs.getString("cdate"));    newsList.add(n);  }  return newsList;  }  }    public News getNews(int aid) throws SQLException {  Connection conn = open();    News n = new News();    String sql = "SELECT aid, title, img, TO\_TIMESTAMP(ndate, 'YYYY-MM-DD HH24:MI:SS.FF') AS cdate, content FROM news WHERE aid = ?";    PreparedStatement pstmt = conn.prepareStatement(sql);  pstmt.setInt(1, aid);  ResultSet rs = pstmt.executeQuery();    rs.next();    try(conn; pstmt; rs) {  n.setAid(rs.getInt("aid"));  n.setTitle(rs.getString("title"));  n.setImg(rs.getString("img"));  n.setNdate(rs.getString("cdate"));  n.setContent(rs.getString("content"));  pstmt.executeQuery();  return n;  }  }    public void delNews(int aid) throws SQLException {  Connection conn = open();    String sql = "delete from news where aid=?";  PreparedStatement pstmt = conn.prepareStatement(sql);    try(conn; pstmt) {  pstmt.setInt(1, aid);  // 삭제된 뉴스 기사가 없을 경우  if(pstmt.executeUpdate() == 0) {  throw new SQLException("DB에러");  }  }  }  } |
| package ch10;  import java.io.IOException;  import java.lang.reflect.Method;  import java.sql.SQLException;  import java.util.List;  import javax.servlet.RequestDispatcher;  import javax.servlet.ServletConfig;  import javax.servlet.ServletContext;  import javax.servlet.ServletException;  import javax.servlet.annotation.MultipartConfig;  import javax.servlet.annotation.WebServlet;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import javax.servlet.http.Part;  import org.apache.commons.beanutils.BeanUtils;  /\*\*  \* Servlet implementation class NewsController  \*/  @WebServlet("/news.nhn")  @MultipartConfig(maxFileSize=1024\*1024\*2, location="/Users/happy/Work/tmp/img")  public class NewsController extends HttpServlet {  private static final long serialVersionUID = 1L;    private NewsDAO dao;  private ServletContext ctx;    // 웹 리소스 기본 경로 지정  private final String START\_PAGE = "ch10/newsList.jsp";    public void init(ServletConfig config) throws ServletException {  super.init(config);  dao = new NewsDAO();  ctx = getServletContext();  }    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  request.setCharacterEncoding("utf-8");  String action = request.getParameter("action");    dao = new NewsDAO();    // 자바 리플렉션을 사용해 if, switch 없이 요청에 따라 구현 메서드가 실행되도록 함.  Method m;  String view = null;    // action 파라미터 없이 접근한 경우  if (action == null) {  action = "listNews";  }    try {  // 현재 클래스에서 action 이름과 HttpServletRequest 를 파라미터로 하는 메서드 찾음  m = this.getClass().getMethod(action, HttpServletRequest.class);    // 메서드 실행후 리턴값 받아옴  view = (String)m.invoke(this, request);  } catch (NoSuchMethodException e) {  e.printStackTrace();  // 에러 로그를 남기고 view 를 로그인 화면으로 지정, 앞에서와 같이 redirection 사용도 가능.  ctx.log("요청 action 없음!!");  request.setAttribute("error", "action 파라미터가 잘못 되었습니다!!");  view = START\_PAGE;  } catch (Exception e) {  e.printStackTrace();  }    // POST 요청 처리후에는 리디렉션 방법으로 이동 할 수 있어야 함.  if(view.startsWith("redirect:/")) {  // redirect/ 문자열 이후 경로만 가지고 옴  String rview = view.substring("redirect:/".length());  response.sendRedirect(rview);  } else {  // 지정된 뷰로 포워딩, 포워딩시 컨텍스트경로는 필요없음.  RequestDispatcher dispatcher = request.getRequestDispatcher(view);  dispatcher.forward(request, response);  }  }    public String addNews(HttpServletRequest request) {  News n = new News();  try {  // 이미지 파일 저장  Part part = request.getPart("file");  String fileName = getFilename(part);  if(fileName != null && !fileName.isEmpty()){  part.write(fileName);  }  // 입력값을 News 객체로 매핑  BeanUtils.populate(n, request.getParameterMap());    // 이미지 파일 이름을 News 객체에도 저장  n.setImg("/img/"+fileName);  dao.addNews(n);  } catch (Exception e) {  e.printStackTrace();  ctx.log("뉴스 추가 과정에서 문제 발생!!");  request.setAttribute("error", "뉴스가 정상적으로 등록되지 않았습니다!!");  return listNews(request);  }    return "redirect:/news.nhn?action=listNews";    }  public String listNews(HttpServletRequest request) {  List<News> list;  try {  list = dao.getAll();  request.setAttribute("newslist", list);  } catch (Exception e) {  e.printStackTrace();  ctx.log("뉴스 목록 생성 과정에서 문제 발생!!");  request.setAttribute("error", "뉴스 목록이 정상적으로 처리되지 않았습니다!!");  }  return "ch10/newsList.jsp";  }    public String getNews(HttpServletRequest request) {  int aid = Integer.parseInt(request.getParameter("aid"));  try {  News n = dao.getNews(aid);  request.setAttribute("news", n);  } catch (SQLException e) {  e.printStackTrace();  ctx.log("뉴스를 가져오는 과정에서 문제 발생!!");  request.setAttribute("error", "뉴스를 정상적으로 가져오지 못했습니다!!");  }  return "ch10/newsView.jsp";  }    public String deleteNews(HttpServletRequest request) {  int aid = Integer.parseInt(request.getParameter("aid"));  try {  dao.delNews(aid);  } catch (SQLException e) {  e.printStackTrace();  ctx.log("뉴스 삭제 과정에서 문제 발생!!");  request.setAttribute("error", "뉴스가 정상적으로 삭제되지 않았습니다!!");  return listNews(request);  }  return "redirect:/news.nhn?action=listNews";  }    private String getFilename(Part part) {  String fileName = null;  // 파일이름이 들어있는 헤더 영역을 가지고 옴  String header = part.getHeader("content-disposition");  //part.getHeader -> form-data; name="img"; filename="사진5.jpg"  System.out.println("Header => "+header);  // 파일 이름이 들어있는 속성 부분의 시작위치를 가져와 쌍따옴표 사이의 값 부분만 가지고옴  int start = header.indexOf("filename=");  fileName = header.substring(start+10,header.length()-1);  ctx.log("파일명:"+fileName);  return fileName;  }  } |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl" crossorigin="anonymous">  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js" integrity="sha384-b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0" crossorigin="anonymous"></script>  <title>뉴스 관리 앱</title>  </head>  <body>  <div class="container w-75 mt-5 mx-auto">  <h2>뉴스 목록</h2>  <hr>  <ul class="list-group">  <c:forEach var="news" items="${newslist}" varStatus="status">  <li class="list-group-item list-group-item-action d-flex justify-content-between align-items-center"><a href="news.nhn?action=getNews&aid=${news.aid}" class="text-decoration-none">[${status.count}] ${news.title}, ${news.ndate}</a>  <a href="news.nhn?action=deleteNews&aid=${news.aid}"><span class="badge bg-secondary">&times;</span></a>  </li>  </c:forEach>  </ul>  <hr>  <c:if test="${error != null}">  <div class="alert alert-danger alert-dismissible fade show mt-3">  에러 발생: ${error}  <button type="button" class="btn-close" data-bs-dismiss="alert"></button>  </div>  </c:if>  <button class="btn btn-outline-info mb-3" type="button" data-bs-toggle="collapse" data-bs-target="#addForm" aria-expanded="false" aria-controls="addForm">뉴스 등록</button>  <div class="collapse" id="addForm">  <div class="card card-body">  <form method="post" action="/jwbook/news.nhn?action=addNews" enctype="multipart/form-data">  <label class="form-label">제목</label>  <input type="text" name="title" class="form-control">  <label class="form-label">이미지</label>  <input type="file" name="file" class="form-control">  <label class="form-label">기사내용</label>  <textarea cols="50" rows="5" name="content" class="form-control"></textarea>  <button type="submit" class="btn btn-success mt-3">저장</button>  </form>  </div>  </div>  </div>  </body>  </html> |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl" crossorigin="anonymous">  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js" integrity="sha384-b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0" crossorigin="anonymous"></script>  <title>뉴스 관리 앱</title>  </head>  <body>  <div class="container w-75 mt-5 mx-auto">  <h2>${news.title}</h2>  <hr>  <div class="card w-75 mx-auto">  <img class="card-img-top" src="${news.img}">  <div class="card-body">  <h4 class="card-title">Date: ${news.ndate}</h4>  <p class="card-text">Content: ${news.content}</p>  </div>  </div>  <hr>  <a href="javascript:history.back()" class="btn btn-primary"><< Back</a>    </div>  </body>  </html> |

실행결과

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 하늘, 들판이(가) 표시된 사진

자동 생성된 설명