**로고학습일지**

**kt ds University 자바 기반의 데이터 사이언티스트 양성과정**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 08. 17 (금) | 장소 | kt ds University B관 201호 | **시 간** | 09:00~18:00 |
| 학습범위 |  | | | | |
| 작 성 자 | 신예리 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 |  |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| <JSP>  List 🡪 Detail  > <a></a> 페이지와 페이지를 연결해주는 태그  -href : hyper link reference  - <a href=<http://localhost:8080/board/detail>>  ${boardList.get(0).getTitle()}  </a>  🡺 <a href=”/board/detail”> 이렇게 사용해도 무방  ><http://localhost:8080/board/detail?boardId=BD-2018081615-000141>  ┗> Query Parameter / Query String ★★  -위의 정보가 HttpServletRequest의 객체인 request에 들어감!! (Client가 Server에 Request할 때)  🡪 <a href=”/board/detail?boardId=${boardList.get(0).getBoardId()}”>  >Servlet의 doGet  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {    // Query String (Parameter) 값을 얻어온다  // ?boardId=BD-2018080615-000141  String boardId = request.getParameter("boardId");  System.out.println(boardId);    BoardVO boardVO = this.boardService.readOneBoard(boardId);    String path = "/WEB-INF/view/board/detail.jsp";  RequestDispatcher rd = request.getRequestDispatcher(path);  rd.forward(request, response);  }  > <h1>${boardVO.title} </h1> 🡺 .title이 boardVO의 getTitle()을 호출  == ${boardVO.getTitle()}와 같은 값을 가져옴  ※<hr /> 🡺 가로줄 -------------------  브라우저에게 다른 URL을 요청할 것을 응답으로 전송  >response.sendRedirect("/board/detail?boardId=" + boardId);  ┗> 302 Found : 브라우저에게 다른 URL로 이동하라는 명령을 내림  <textarea name="content" placeholder="내용을 입력하세요." ></textarea> : 긴 글을 적을 때  -오페라, 크롬 크기 조절 O (explorer X)  Form Parameter  >Form Element or Form Tag  -<form method="post" action="/board/write"> 🡪 method : 어떤 방식 / action : 어디로 보낼 것인지  </form> (submit을 클릭하면 post 방식으로 "/board/write"로 데이터를 전송하겠다.)  -글 입력과 같이 입력을 통해 데이터를 넣는 영역 (input, textarea….)  -각 입력 데이터의 name과 값을 그대로 가져옴. 🡪 request.getParameter(name)으로 값 가져와  🡺🡺🡺🡺 웹에서 데이터를 보내는 방법 2가지 : 1. Query Parameter 2. Form Parameter  ※200 OK : 모든 작업이 끝났다는 의미    >doGet vs doPost  -deGet : Browser (브라우저 주소 뒤에 값을 넣어 보내는 것) , <a> / method=”get”  -dePost : form / method=”post”  -doPost 사용하는 이유? ★★★  Get은 쿼리 스트링(url)을 통해 보내겠다는 의미. 하지만 url에는 길이 제한이 있다. 255 ( 🡪 1byte)  Post는 2GB까지 가능! 블로그, 게시물의 글과 같이 장문의 데이터는 무조건 post!  >한글이나 특수문자 적으면 깨짐 (영어는 노상관) : request할 때 발생  : Characterset : UTF-8 (다국어 지원)  request.setCharacterEncoding("UTF-8");  response.setCharacterEncoding("UTF-8"); : response할 때 또 깨질 수 있으니 둘 다 설정해줄 것!  🡪모든 Servlet에 적어주어야 한다. (서블릿이 2-300개 이더라도)  🡺 Filter를 web.xml과 Servlet 사이에 적용하여 인코딩 (Filter Mapping : /\* 🡨 모든 파일을 filtering 해주겠다.)  ※조회수 update : 따로 service를 만들 필요가 없다.  Business 업무 : Task들이 모여서 업무가 된다.  >Task : Dao의 하나의 method  조회의 업무 1. 조회수 증가 2. 가져오기 🡪 기존의 readOneBoard에 추가  JSTL  JSTL을 위해서는 2개의 파일이 필요하다  ><http://mvnrepository.com>  1> jstl 검색 > 나온 것중에 많이 사용되는 것 javax.servlet » jstl > 많이 사용되는 것 1.2 클릭 > jar 다운  2> standard 검색 > 2. Standard (taglibs » standard) 클릭 > 1.1.2 jar 다운  ==> 다운받은 두 파일 복사 board\src\main\webapp\WEB-INF\lib에 복붙  🡪 .jsp파일에서 맨위에 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %> 추가  ┗namespace 🡪 c태그라고 불림  >forEach (items와 var 필수)  <c:forEach items="" var=""> for (String boardList : boardVO) {  </c:forEach> }  🡺 items=”” :: String boardList  var=”” :: boardVO  ※IP 주소 : 192.168.201.25  ※post 방식이지만 파라미터를 사용할 수 있다 ☺ |

|  |  |
| --- | --- |
| 특이사항 | 주말 과제 : 게시판 처음부터 만들기 |