SW 프레임워크 응용실습

[ 종합실습 ]



과목명 : SW 프레임워크 - 01

담당교수 : 장성호 교수님

학과 : IT경영학과

학번: 2017316006

이름: 김규정

제출일: 2019-06-19

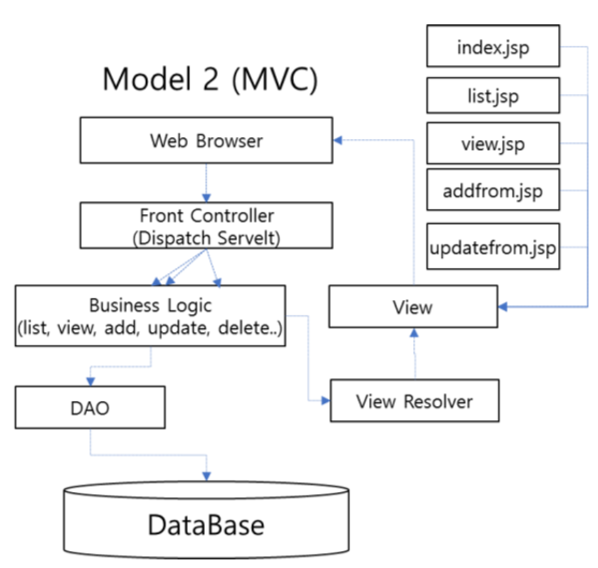
**목 차**

**Ⅰ. 웹사이트 구성 및 제작 과정을 설명하기 위한 개요**

**Ⅱ. 제작과정 설명**

**Ⅲ. 검색기능 구현**

**Ⅰ. 웹사이트 구성 및 제작 과정을 설명하기 위한 개요**



모델2 : MVC 및 표준프레임워크 기반으로 구현되었을 때의 구조

- DB는 DAO를 통해서 접근하게 되며, 처리 결과는 Model을 통해서 View인 JSP를 통해 출력하게 된다.

여기의 JSP들은 화면출력을 위한 템플릿 역할만을 하게 되며, 직접 실행이 되지 않는다.

실행이 되지 않는 WEB-INF밑에 위치를 시키고, 실행이 된다고 하더라도 템플릿 내용만 출력된다.

필요한 기능이 증가되는 경우에 모듈이 증가되는 것이 아니고, 현재의 구조도 그대로 유지하게 되어 대용량의

시스템 구축에 적합하고 변경, 유지보수가 편리하다.

|  |  |  |
| --- | --- | --- |
| 구분 | 코드명 | 설명 |
| [src/main/java] | | |
| Controller | lab.ShoppingController.java | @ModelAttribute를 활용한 model 바인딩과 폼입력과 출력을 설명 |
| Service | ShoppingService.java | 상품관리를 위한 인터페이스 |
| ShoppingServiceImpl.java | 상품관리 구현 |
| DAO | ShoppingDAO.java | 상품관리 데이터베이스 처리 |
| Model | ShoppingVO.java | 상품관리 클래스(VO) |
| [src/main/resources] | | |
| Spring | META-INF/spring/  context-common.xml | Annotation 처리를 위한 component scan (Service, DAO 용) |
| META-INF/spring/  context-datasource.xml | DBCP를 활용한 데이터 풀 제공 |
| META-INF/spring/  context-mybatis.xml | Spring과 MyBatis 연결설정 |
| META-INF/spring/jdbc.properties | DB연결정보 설정 |
| MyBatis | META-INF/sqlmap/  sql-mybatis-config.xml | MyBatis 환경설정 |
| META-INF/sqlmap/mappers/ lab-dao-class.xml | SQL 설정파일인 Mapper정의 |
| Log4j | log4j2.xml | 로그출력 |
| [src/main/webapp] | | |
| SpringMVC | WEB-INF/config/springmvc/ context-srvlet.xm | Spring MVC 설정 |
| View (JSP) | WEB-INF/jsp/shopping/ shoppingAddForm.jsp | 상품정보 입력 폼 출력 |
| WEB-INF/jsp/shopping/ shoppingList.jsp | 상품정보 목록 출력 |
| WEB-INF/jsp/shopping/  shoppingModifyForm.jsp | 상품정보 수정 폼 출력 |
| WEB-INF/jsp/shopping /shoppingView.jsp | 상품정보 상세보기 출력 |
| Web 설정 | WEB-INF/web.xml | Spring ApplicationContext 및 Dispatcher Servlet 설정 |
| Index | Index.jsp | 실행을 위한 링크 |

**내 상품**

**: 의류가게**

**-> 의류를 관리하기 위한 홈페이지를 만듦 = 옷 아이디, 옷 이름, 옷 가격, 옷 색상, 옷 수량**

**Ⅱ. 제작과정 설명**

**나의 상품 : 옷**

**[과제 2에서 구현한 내용]**

1. eGovFrame Web Project를 생성

2. MySQL JDBC 드라이버 설정

스크린샷이(가) 표시된 사진

자동 생성된 설명

3. Spring 설정

- context-common.xml: Annotation 활용을 위한 component-scan 설정

- context-datasource.xml: DB연결

- context-mybatis.xml: Spring에서 MyBatis 활용을 위한 sqlsession Bean 설정

- jdbc.properties: DB연결 환경 변수

4. MyBatis 설정

5. MyBatis SQL Mapper설정 -> mappers/lab-dao-class.xml

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

: DAO에서 DB 입력, 수정, 삭제, 조회를 수행하는 SQL을 설정하고, SQL실행에 필요한 입력 파라미터와 결과 값을 반환하는 오브젝트인 resultmap을 설정한다.

6. Spring 활용을 위한 웹 어플리케이션 설정

7. Controller annotation 위한 component-scan 및 View Resolver 설정을 위한 context-servlet.xml 작성

8. Model 작성, DAO작성 (shoppingDAO.java), Service작성, Service Implemetation작성

스크린샷이(가) 표시된 사진

자동 생성된 설명

: MyBatis에서 파라미터와 resultmap정의에 필요한 VO 클래스를 정의

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

: MyBatis에서 정의한 SQL ID와 실행에 필요한 파라미터를 정의하여 입력, 수정, 삭제 작업을 위한 메소드를 정의하며, 서비스에서 호출되어 실행된다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

: 서비스에서 작성할 메소드에 대한 인터페이스를 정의

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

: 인터페이스에서 정의된 메소드를 구현하며, DAO의 메소드를 호출한다.

**[과제 3 화면 구현을 위해 새로 추가한 내용]**

9. Controller 작성

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

22~26줄 : 상품 정보를 입력하기 위한 부분을 출력

28~36줄 : 신규로 상품정보를 입력하고, 그 상품 정보의 리스트를 출력

38~44줄 : 상품정보를 리스트로 조회하고 그 상품 정보를 리스트로 출력

46~54줄 : 선택된 상품 정보들을 조회하고 그 정보를 출력

56~64줄 : 상품 수정을 위한 폼을 출력

66~74줄 : 수정된 상품 정보들을 반영하고 그 상품정보들을 조회하여 화면으로 출력

76~83줄 : 선택된 상품 정보를 삭제해주고 그 상품 목록들을 출력

10. View (JSP) 작성

|  |  |  |
| --- | --- | --- |
| View (JSP) | WEB-INF/jsp/shopping/ shoppingAddForm.jsp | 상품정보 입력 폼 출력 |
| WEB-INF/jsp/shopping/ shoppingList.jsp | 상품정보 목록 출력 |
| WEB-INF/jsp/shopping/  shoppingModifyForm.jsp | 상품정보 수정 폼 출력 |
| WEB-INF/jsp/shopping /shoppingView.jsp | 상품정보 상세보기 출력 |

: View는 controller에서 서비스 메소드를 실행한 후에 결과를 화면에 보여줌.

**WEB-INF/jsp/shopping/ shoppingAddForm.jsp: 상품 정보 입력 폼 출력**

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

: 입력 폼은 신규 추가 버튼을 눌렀을 때 보여지게 되며, submit이 될 때 지정된 action으로 input에 입력된 값

이 클래스로 바인딩이 되어 controller의 @ModelAttribte로 매핑되어 전달 됨.

**WEB-INF/jsp/shopping/ shoppingList.jsp: 상품 정보 목록 출력**

스크린샷이(가) 표시된 사진

자동 생성된 설명스크린샷이(가) 표시된 사진

자동 생성된 설명

: List 출력의 경우, SQL 실행결과로 출력된 레코드가 클래스에 담겨서 리스트 형태로 출력됨.

따라서 List에서 클래스를 하나씩 꺼내고 메소드를 view 태그에 매핑하여 출력함.

**WEB-INF/jsp/shopping/ shoppingModifyForm.jsp: 상품 정보 수정 폼 출력**

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

: 수정 폼과 입력 폼의 출력은 유사

: input 태그에 value 값을 주어 초기 값을 정의하여, 출력하고 사용자가 수정 가능함.

**WEB-INF/jsp/shopping/ shoppingView.jsp: 상품 정보 상세보기 -> 수정, 삭제하도록 링크 제공**

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

: 상품정보 조회화면은 선택된 단일 정보가 select 되어 화면에 출력됨.

: 조회화면에서 선택된 정보를 수정하거나 삭제할 수 있도록 링크 제공함.

**Index.jsp**

스크린샷이(가) 표시된 사진

자동 생성된 설명

**Ⅲ. 검색 기능 구현**

|  |  |  |
| --- | --- | --- |
| 구분 | 코드명 | 설명 |
| [src/main/java] | | |
| Controller  Service | ShoppingController.java | 검색조건, 페이징 처리 위한 파라미터 추가 |
| ShoppingService.java | 리스트출력 파라미터 MAP으로 변경 |
| ShoppingServiceImpl.java | 리스트출력 파라미터 MAP으로 변경 |
| DAO | ShoppingDAO.java | 리스트출력 파라미터 MAP으로 변경 |
| Model | SearchCritetia.java | 검색조건 VO추가 |
| [src/main/resources] | | |
| MyBatis | META-INF/sqlmap/mappers/  lab-dao-class.xml | 검색과 페이징 반영 List SQL 수정, 페이지위한 Count SQL 추가 |
| [src/main/webapp] | | |
| View (JSP) | WEB-INF/jsp/shopping/ shoppingList.jsp | 상품정보 목록 출력 |

**SearchCritetia.java**

스크린샷이(가) 표시된 사진

자동 생성된 설명

: 검색 조건이 여러 개 존재하는 경우 VO를 따로 추가하는 것이 효율적임

: 검색 조건과 검색어 정보를 관리하는 VO를 따로 추가.

**ShoppingDAO.java**

스크린샷이(가) 표시된 사진

자동 생성된 설명

: DAO의 파라미터를 MAP의 형태로 변경

: 전체 ROW 수를 구하는 메소드 추가

**ShoppingService.java**

스크린샷이(가) 표시된 사진

자동 생성된 설명

**ShoppingServiceImpl.java**

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

: Service, serviceImpl 도 파라미터를 MAP 형태로 변경하고 메소드를 추가함.

**META-INF/sqlmap/mappers/lab-dao-class.xml**

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

: 검색 구현을 위한 MYSQL JDBC 드라이버

**ShoppingController.java**

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 사람이(가) 표시된 사진

자동 생성된 설명

:검색과 관련된 부분은 @ModelAttribute를 활용한 부분과 @RequestParam 부분,

: 이 둘은 Map을 통해서 동시에 정보를 전달 받음 -> 즉, searchCriteriaVO 와 commandMap Map에 같은 값 받음.

: @RequestParam Map<String, Object> commandMap으로 파라미터를 받게 되면 파라미터가 모두 key와 value값으로 매핑이 되어 Map으로 저장됨.

: @ModelAttribute는 JSP에서 form 객체로 연결하기 위해 필요한 부분임.

: 전달 받은 검색조건에 페이징 정보를 추가하여 서비스에 전달해야 되기 때문에 편의상 중복되지만 추가적으로 Map으로 파라미터를 전달 받았다.

: 전달 받은 검색조건에 페이징 처리를 위한 설정정보와 전체 출력건수를 View로 전달하여 페이징을 구성하게 된다.

**WEB-INF/jsp/shopping/ shoppingList.jsp**

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

: 표준프레임워크에서 제공하는 pagination기능을 활용하면 태그라이브러리 형태로 제공이 됨

-> <ui:pagination paginationInfo = "${paginationInfo}" type="image" jsFunction="linkPage" /> 호출

-> 페이징 구현

첫번째 네모 – 페이지 리스트에 링크가 걸려서 페이지가 눌렸을 때 실행됨

두번째 네모 – 검색 값 입력 창과 실행버튼 생성

세번째 네모 - Controller로부터 전달받은 페이징 정보를 활용하여 페이지 리스트를 생성하는 태그 라이브러리