









▼ kr.bit.config

- >  RootApplicationContext.java
- >  ServletApplicationContext.java
- >  SpringConfigClass.java

스프링 초기 설정 패키지

1. 프로젝트 작업시 사용할 Bean들을 정의하는 클래스
로그인 여부에 따라 기능에 차이가 있어야하므로 user빈을 Session영역에 저장한다.
2. 스캔할 패키지지정(controller, dao, service), 인터셉터 등록
3. 스프링 설정을 위한 AbstractAnnotation~클래스를 상속받아 구현






▼ kr.bit.controller

- >  BoardController.java
- >  HomeController.java
- >  MainController.java
- >  RestTestController.java
- >  UserController.java

컨트롤러





1. BoardService 자동주입, 로그인 정보 세션에 저장, board_info_idx를 통해 게시물 조회, 작성 수정 삭제
2. 서버 실행 시 main으로 이동
3. Main으로 이동
4. RestTestController: UserMapper의 id중복확인과 연결 / UserService를 사용 후 참, 거짓반환
5. 회원가입과 로그인에 대한 매핑 — 유효성검사(modelAttribute로 들어온 BindingResult에 에러여부 확인, @initBinder추가하면 추가로 validator실행 — UserService를 이용할 수 있도록 @autowired를 이용해 주입받고 join_pro에서 service를 이용해 호출

▼ board

-  delete.jsp
-  main.jsp
-  modify.jsp
-  read.jsp
-  write.jsp






게시판

1. 게시물 삭제시 확인창 팝업, 게시판 메인 페이지로 이동
2. 게시판 메인 페이지-> 게시물 리스트 조회
3. 게시물 수정 페이지
4. 게시물 제목을 선택하면 해당 게시글을 볼 수 있는 페이지
5. 게시물 작성 페이지 — 유효성 검사

▼  kr.bit.service
 >  BoardService.java
 >  TopMenuService.java
 >  UserService.java

컨트롤러와 DAO 사이에서 비즈니스 로직(사용자 니즈에 맞게 구현)을 수행하는 서비스 패키지

1. Content빈을 매개로 BoardDao 함수 실행
2. TopMenuDao 자동주입하여 상단메뉴를 가져옴
3. User빈을 매개로 UserDao의 함수 실행

▼  kr.bit.beans
 >  BoardInfo.java
 >  Content.java
 >  Page.java
 >  User.java





DB에 만든 테이블 관련 Bean패키지 (Bean멤버변수=DB테이블 속성이름)

1. 게시판 번호와 게시판 이름 설정, Getter&Setter 생성
2. 게시글 테이블에 해당하는 빈: 게시글 번호, 제목 내용 글쓴이 작성일 설정
3. 게시판의 최소, 최대 페이지 번호, 이전 다음페이지, 전체페이지개수, 현재페이지 번호
4. 사용자 테이블에 해당하는 빈: 사용자 정보에 대한 변수 설정& 유효성검사

▼  kr.bit.validator
 >  UserValidator.java





사용자 비밀번호 및 아이디 중복 확인 검사 패키지

1. Validator인터페이스 상속받고, 입력한 비밀번호 동일 여부를 검사 후 이에 대한 에러메시지를 properties에 등록하고, userController에도 등록 회원가입시만 아이디 중복확인하고 로그인 때는 검사하지 않도록 빈 이름을 joinBean으로 한정

- ▼  kr.bit.dao
 - >  BoardDao.java
 - >  TopMenuDao.java
 - >  UserDao.java





Data Access Object : DB에 직접 연결

1. BoardMapper 자동주입 받아 매퍼의 addContent함수의 content객체 매개변수 넣어서 실행. 게시판 번호에 해당하는 게시글을 List로 받아와 리턴
2. TopMenuMapper 자동주입 받아 매퍼에서 반환된 값을 List에 저장, 반환
3. UserMapper 자동주입 받아 User빈을 매개로 회원조회, 가입 정보수정

- ▼  kr.bit.interceptor
 - >  LoginInterceptor.java
 - >  TopMenuInterceptor.java
 - >  WriterInterceptor.java

인터셉터 패키지: controller호출 전후에 수행되어 controller제어

1. 로그인여부에 따라 접근할 수 있는 페이지가 다르므로, 로그인 하지 않았을 경우, user/not_login 페이지로 이동. loginBean 생성자를 통해 받아서 로그인 여부 판단. ServletApp에 인터셉터 등록
2. TopMenu는 모든 화면이 뜨므로, interceptor를 이용해 모든 요청에 대해 topMenuList를 가져오고, request영역에 attribute를 등록("team") 이후 로그인 여부에 따라 뜰 수 있는 화면이 다르도록 설정할거임
3. 게시글 작성자가 아닌 사용자가 수정, 삭제하지 못하도록 구현. 로그인 사용자와 게시글 작성자의 idx를 비교하여 같지 않으면 접근 차단하고 not_writer페이지로 이동

- ▼  kr.bit.mapper
 - >  BoardMapper.java
 - >  TopMenuMapper.java
 - >  UserMapper.java

SQL호출하기 위한 Mapper인터페이스 패키지

1. 자동생성값을 content_idx에 저장, 게시글 작성, 수정
2. 상단메뉴를 클릭했을 때 각 게시판에 맞는 게시판 메인페이지로 연결 (Board_info_table에서 idx, name을 가져와 BoardInfo리스트로 반환. ServletApp에 매퍼 등록
3. 사용자 정보를 조회하여 아이디 중복확인(RestTestController), 신규회원 추가, 수정하는 sql문 쿼리문 등록



bottom_menu.jsp

top_menu.jsp

반복되는 화면(상단 메뉴바, 하단부) 페이지를 저장한 폴더

1. 하단부 : 사업장 기본 정보(팀이름 팀원명 사이트명)
2. 상단 메뉴 : jstl의 taglib을 이용해 root경로를 지정하여 링크 설정- 인터셉터 걸거임 (로그인 되었을때랑 되지 않았을때 다른 화면을 보이도록 설정할거임)



join_success.jsp

join.jsp

login_fail.jsp

login_success.jsp

login.jsp

logout.jsp

modify_success.jsp

modify.jsp

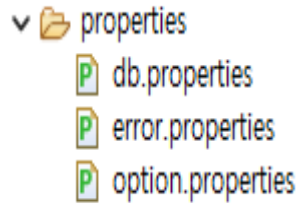
not_login.jsp

회원관리를 위한 폴더

1. 가입완료시 페이지 , user/login으로 하이퍼 링크
2. 회원가입 페이지, 아이디중복확인 - 스크립트 통해 중복확인 가능. User_idx반환값을 위해 hidden태그 추가 하고 이 값으로 Validator에서 유효성 검사
3. 로그인 실패 페이지: UserController에서 로그인 실패시 매핑. C태그를 이용해 fail변수가 true이면 경고를 띄우고, controller에서 이를 login.jsp로 전달한다.
4. 로그인 성공 페이지: UserController에서 로그인 성공시 매핑
5. 로그인페이지 (로그인 여부에 따라 기능차이가 있어야하므로, UserBean을 세션영역에 저장 후 RootApp에 추가)
6. 로그아웃 페이지
7. 수정이 성공했을 때 페이지
8. 회원정보수정페이지 : 이름 아이디 변경불가(readonly)
9. 로그인 하지 않았을 때, 로그인 시 사용할 수 있는 페이지에 접근할 경우 뜨는 페이지

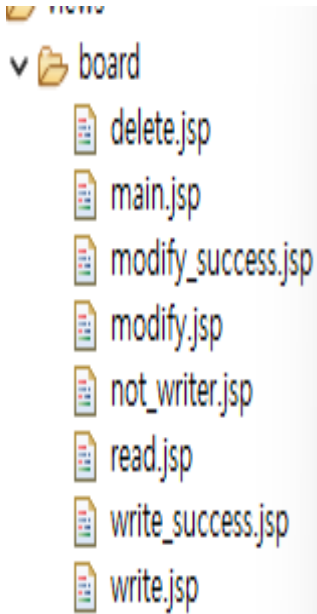
main.jsp

첫화면 : HomeController에서 넘김



고정된 상수값을 저장할 properties파일 폴더

1. DB와 연결하기 위한 정보를 저장(ServletApp에 접속코드와 연결)
2. 유효성 검사에 대한 에러메세지 세팅, ServletApp에등록
3. 한 페이지당 페이지 개수와, 페이지에서 보여지는 인덱스 번호 개수 지정



게시판

1. 게시글 삭제시 확인창 팝업, 게시판 메인 페이지로 이동
2. 게시판 메인 페이지-> 게시글 리스트 조회
3. 게시글 수정시 확인창 팝업, 게시글 수정 페이지로 이동
4. 게시글 수정 페이지
5. writerInterCeptor에서 게시글 작성자와 로그인 사용자가 다를 시 반환되는 페이지
6. 게시글 제목을 선택하면 board_info_idx와 content_idx로 해당 게시글을 볼수 있는 페이지
7. 게시글 작성 완료시 뜨는 페이지
8. 게시글 작성 페이지 : 글 제목과 내용을 쓰지 않으면 오류가 뜨도록 Content빈에서 유효성 검사