

주요 파일 설명

[Controller]

<AttLectureController.java>

@RequestMapping("/attLectures")

public String showLecture(HttpServletRequest request, Model model) :

년도/학기별 총 학점, 수강 리스트 조회 카테고리 보기 위한 페이지로 이동한다.

@RequestMapping("/viewLectures")

public String showLectureList(HttpServletRequest request, Model model) :

년도/학기별 상세 수강 리스트 조회 페이지로 이동한다. HttpServletRequest 로 년도, 학기 파라미터를 받아 모델에 해당하는 데이터만 담아 보내준다.

@RequestMapping("/viewRegistLectures")

public String showRegistLectures(Model model) :

2020 년 1 학기 수강 신청 완료 강의 리스트 조회 페이지로 이동한다.

모델에 2020 년 1 학기에 들을 예정인 데이터만 담아 보내준다.

@RequestMapping("/registerLecture")

public String createoffer(Model model) :

수강 신청 페이지로 이동한다. 수강 신청 카테고리 클릭 시 아무 것도 입력하지 않았을 때의 화면을 띄울 때는 model 이 비어있게 되므로 빈 model 객체를 생성해 담아준다.

@RequestMapping("/dcreate")

public String dcreate(Model model, @Valid AttLecture lecture, BindingResult result) :

lecture 인자에서 데이터 바인딩이 되고, result 인자에 바인딩 한 후 검증된 결과가 담긴다. 검증된 결과 데이터에 에러가 있을 경우 수강 신청 페이지로 다시 이동하게 되고, 결과 데이터에 에러가 없을 경우 attLectureService 의 insert() 메서드를 호출해 강의를 데이터베이스에 insert 해준다. 그 후 수강 신청 성공 페이지로 이동한다.

<HomeController.java>

@RequestMapping(value = "/", method = RequestMethod.GET)

public String home(Locale locale, Model model) :

루트쪽으로 리퀘스트가 오면 불리게 되는 메소드이다. home 이라고 하는 view 를 리턴한다. home.jsp 가 호출된다.

<LoginController.java>

String showLogin(@RequestParam(value="error", required=false) String error, @RequestParam(value="logout", required=false) String logout, Model model) :

로그인 에러 발생시 넘겨줄 에러 메시지와 로그아웃 성공시 넘겨줄 메시지를 login뷰로 넘겨준다.

[DAO]

<AttLectureDao.java>

setDataSource(DataSource dataSource) :

@Autowired 로 dataSource 를 찾아서 주입해준다. 인자로 받은 dataSource 를 사용해 jdbcTemplate 인스턴스를 생성한다.

List<AttLecture> getLecture() :

sql문을 "select distinct att_year, att_term, sum(sub_credit) from lectures group by att_year, att_term order by att_year asc;" 로 하여 년도, 학기, 별로 그룹화 하여 년도/ 학기/ 이수 학점 합계 데이터를 가져온다.

List<AttLecture> getLectureByYearTerm(int att_year, int att_term) :

년도, 학기를 인자로 받아 해당 년도, 학기에 따른 강의 데이터를 가져와 리스트에 담아 리턴해 준다.

boolean insert(AttLecture lecture) : AttLecture 객체를 인자로 받아

데이터베이스에 insert하는 메서드이다. CRUD에서 C(create)에 해당한다.

boolean update(AttLecture lecture) : DB 를 update 하는 메서드이다.

[Model]

<AttLecture.java>

강의 정보를 담는 AttLecture 클래스이다. 년도(att_year), 와 학기(att_year)는 뷰에서 사용자가 입력 및 수정을 하지 못하도록 막아두었기에 에러 처리를 하지 않았다. 나머지 정보들인 강의 코드(sub_code), 강의명(sub_title), 구분(sup_part), 학점(sub_credit)은 각각의 타입에 맞게 @NotEmpty, @NotNull 어노테이션을 사용하여 공란이 있으면 안 됨, 공란이 있을 경우에는 사용자에게 보여줄 에러 메시지를 설정하였다.

[Service]

<AttLectureService.java>

List<AttLecture> getLecture() : 모든 강의 리스트를 받아오는 메서드이다.

void insert(AttLecture lecture) : 강의를 추가하는 메서드이다.

List<AttLecture> getLectureByYT(int att_year, int att_term) :

년도와 학기를 인자로 받아 해당 년도와 학기에 해당하는 강의들만
리스트로 받아오는 메서드이다.

[View]

<home.jsp>

웹 접속 시 가장 처음으로 보이는 홈 화면이다. 로그아웃 카테고리는 로그인 한 사용자에게만 나타나도록 하였다.

<login.jsp>

로그인 페이지이다. 아이디/비밀번호가 틀리거나 입력을 하지 않았을 때 상단에 빨간색으로 에러 메시지가 뜨도록 하였다.

<attLectures.jsp>

년도/학기 별 이수 학점을 조회하는 페이지이다. 조회 링크에 hidden 타입으로 파라미터를 전달해 해당 년도/학기 별 상세 이수 과목 리스트를 볼 수 있도록 하였다.

<offercreated.jsp>

수강 신청 성공 시 넘어가게 되는 페이지이다.

<viewLectures.jsp>

년도/학기 별 상세 이수 과목 리스트를 확인하는 페이지이다.

<registerLecture.jsp>

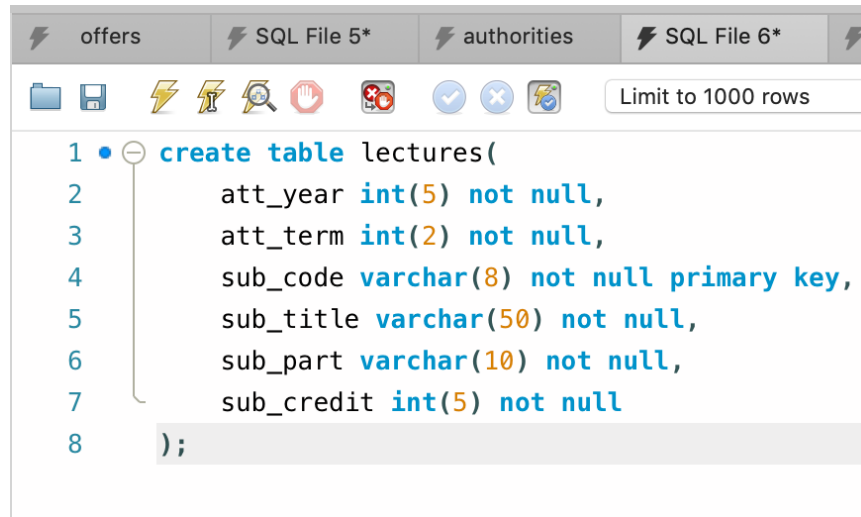
수강 신청 페이지이다. 년도/학기는 유저가 입력 및 수정할 수 없도록 readonly="true"로 지정하였다. <sf:errors/>로 정보 기입 누락 시 해당 정보에 관한 에러 메시지가 나타나도록 설정하였고, 데이터 바인딩을 하였기에 재입력 페이지로 돌아가도 이전에 입력해 놓았던 정보가 그대로 유지되도록 하였다.

<viewRegistLectures.jsp>

수강 신청 완료한 과목 리스트를 확인하는 페이지이다.

[MySQL 테이블]

<lectures.sql>

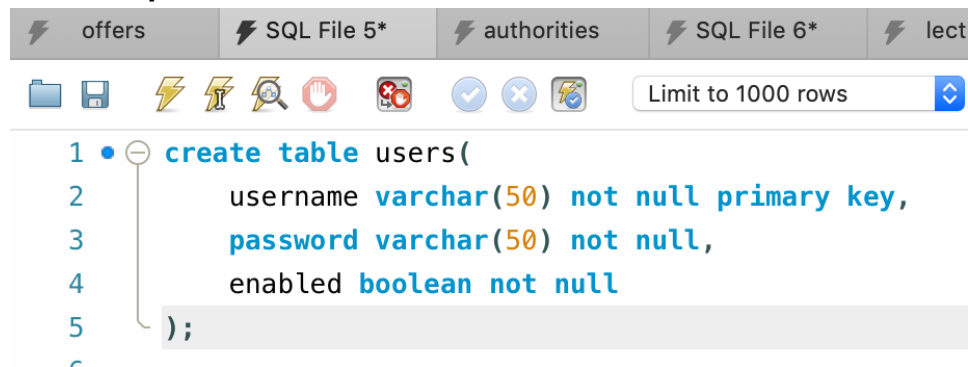


The screenshot shows a MySQL IDE window with a tab labeled 'SQL File 5*'. The SQL editor contains the following code:

```
1 • ○ create table lectures(  
2     att_year int(5) not null,  
3     att_term int(2) not null,  
4     sub_code varchar(8) not null primary key,  
5     sub_title varchar(50) not null,  
6     sub_part varchar(10) not null,  
7     sub_credit int(5) not null  
8 );
```

The interface includes a toolbar with icons for file operations, execution, and a 'Limit to 1000 rows' dropdown menu.

<users.sql>



The screenshot shows a MySQL IDE window with a tab labeled 'SQL File 6*'. The SQL editor contains the following code:

```
1 • ○ create table users(  
2     username varchar(50) not null primary key,  
3     password varchar(50) not null,  
4     enabled boolean not null  
5 );
```

The interface includes a toolbar with icons for file operations, execution, and a 'Limit to 1000 rows' dropdown menu.