

JSP PROJECT

MOVIE WEBSITE



JSP PROJECT MOVIE WEBSITE

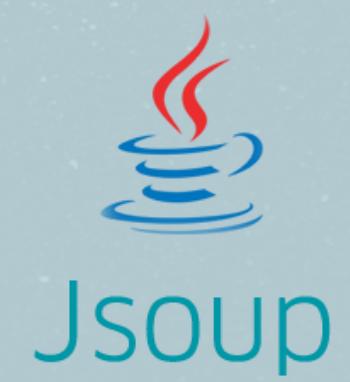
- 개요
- 사용된 스킬
- 역할 분담
- 조원별 일정
- 프론트설계도
- 백엔드 설계도 - ER다이어그램 / 패키지구조도
- 메인 기능
- 서블릿
- 느낀점 보완할 점



예약시스템 및 상영관 구현에 대한 도전

영화정보, 검색 등 풍부한 API 활용

스킬





권태형

조장(GIT취합), 관리자 페이지, 개봉예정작

이정준

크롤링 이용한 영화 검색, 박스오피스, 넷플릭스 순위, 영화정보 출력, 게시판(평점,리뷰)

이경석

영화 예매 [영화상영정보별 출력 및 예매 (달력 등), 상영관 설계]

유슬비

발표, 메인페이지, 마이페이지(예매조회,취소), 회원제, 기초 CSS

권태형

- 3/23 | 주제선정 , DB설계
- 3/24 | 기초 MVC 설계 , 기초 DAO,DTO 생성
- 3/27 | 상영정보 추가 페이지 생성, 기초 CSS 및 입력창 구성
- 3/28 | 기본 생성 규칙 select 구성, 선택한 정보와 겹치는 상영정보들 출력
- 3/29 | 상영정보 등록 설계 및 구현
- 3/30 | 관리자페이지 달력제작
- 3/31 | 달력내에 상영중인 영화 정보 가져오기
- 4/3 | 해당 날짜의 상영정보들을 모달로 띄우기
- 4/4 | 모달내의 상영정보 1개의 상세버튼 클릭시 해당 상영정보 출력
- 4/5 | 크롤링으로 영화 상세표시 시에 영화 정보 출력
- 4/6 | 크롤링이 아닌 API 이용하여 영화 포스터 출력으로 수정
- 4/7 | 상영정보 전체 출력 페이지에 관리자를 위한 애매율 및 영화당 매출 차트 생성
- 4/10 | 버튼으로 차트 변경 되도록 수정, 구체적인 CSS,
기초DB 생성을 위한 랜덤함수 구현
- 4/11 | 취합 및 기능 테스트로 프로젝트 전체의 기능 보완

이경식

- 3/23 | 주제선정 , DB설계
- 3/24 | 기초 MVC 설계 , 기초 DAO,DTO 생성
- 3/27 | 예매, 좌석 페이지 디자인 피그마
- 3/28 | 상영중인 영화 리스트 출력 + css
- 3/29 | 현재 날짜를 기준으로 달력 출력 + css
- 3/30 | 상영중인 영화의 상영관 출력 + css
- 3/31 | 예매페이지 오류 수정
- 4/03 | 좌석 페이지 인원선택 기능 및 좌석 예약기능 제작중 + css
- 4/04 | 좌석 예약 기능 제작중...
- 4/05 | 좌석 예약 기능 완성
- 4/06 | 결제 api 설정
- 4/07 | 좌석+결제페이지에 외부 api가져와 영화 이미지 설정
- 4/10 | 통합 테스트 및 보수
- 4/11 | 보수 및 발표준비

이정준

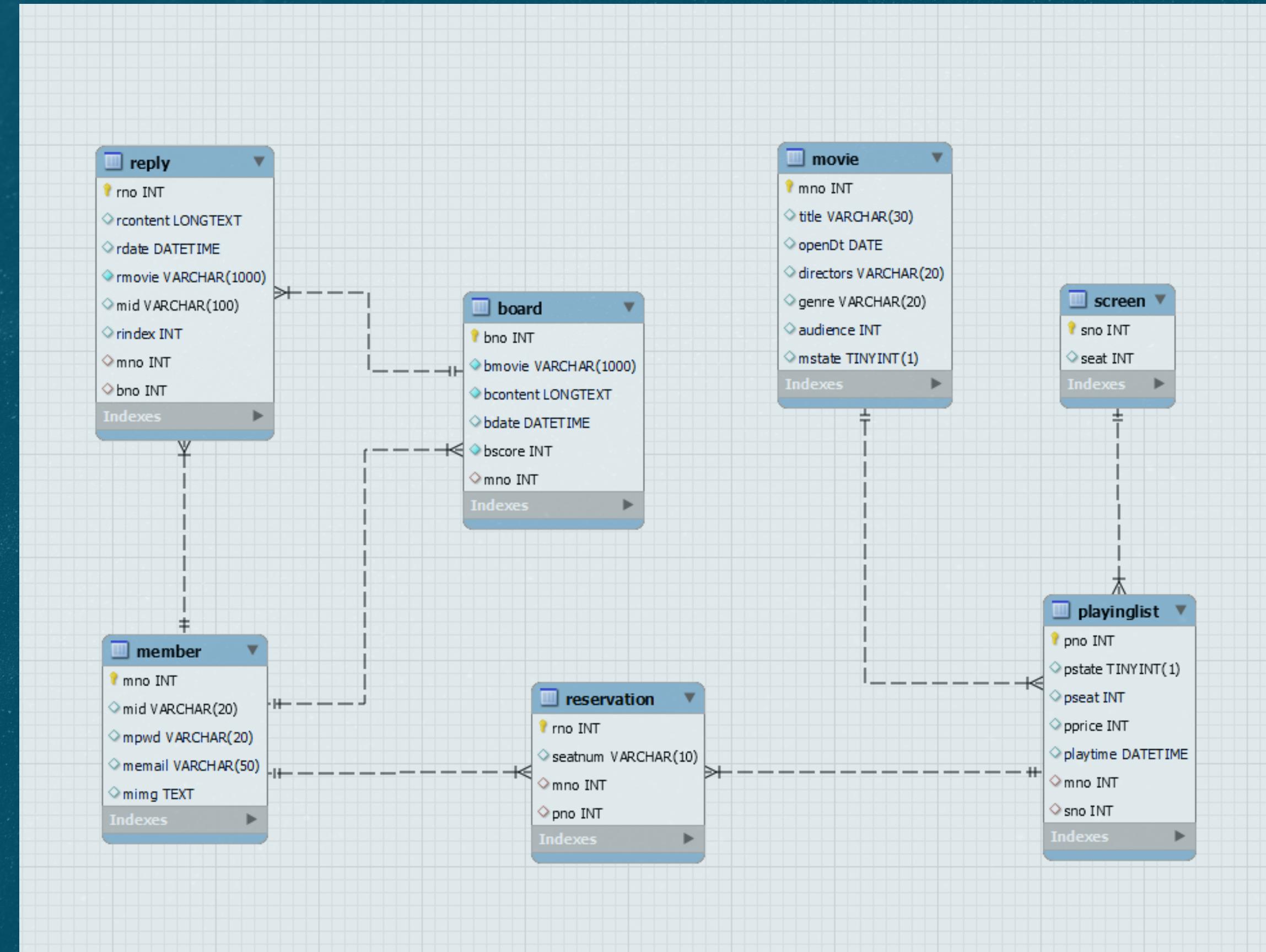
- 3/23 | 주제선정 , DB설계
- 3/24 | 기초 MVC 설계 , 기초 DAO,DTO 생성
- 3/27 | 크롤링할 사이트 선정 및 Jsoup 학습
- 3/28 | Jsoup 크롤링 구현
- 3/29 | Jsoup 크롤링 구현
- 3/30 | Jsoup 검색 구현
- 3/31 | Jsoup 검색 구현
- 4/3 | Jsoup 영화 상세보기 페이지 구현
- 4/4 | Jsoup 영화 상세보기 페이지 구현
- 4/5 | 상세보기 페이지의 리뷰
- 4/6 | 상세보기 페이지의 리뷰 댓글
- 4/7 | 리뷰게시판
- 4/10 | CSS 및 유효성검사
- 4/11 | 보수 및 발표준비

유슬비

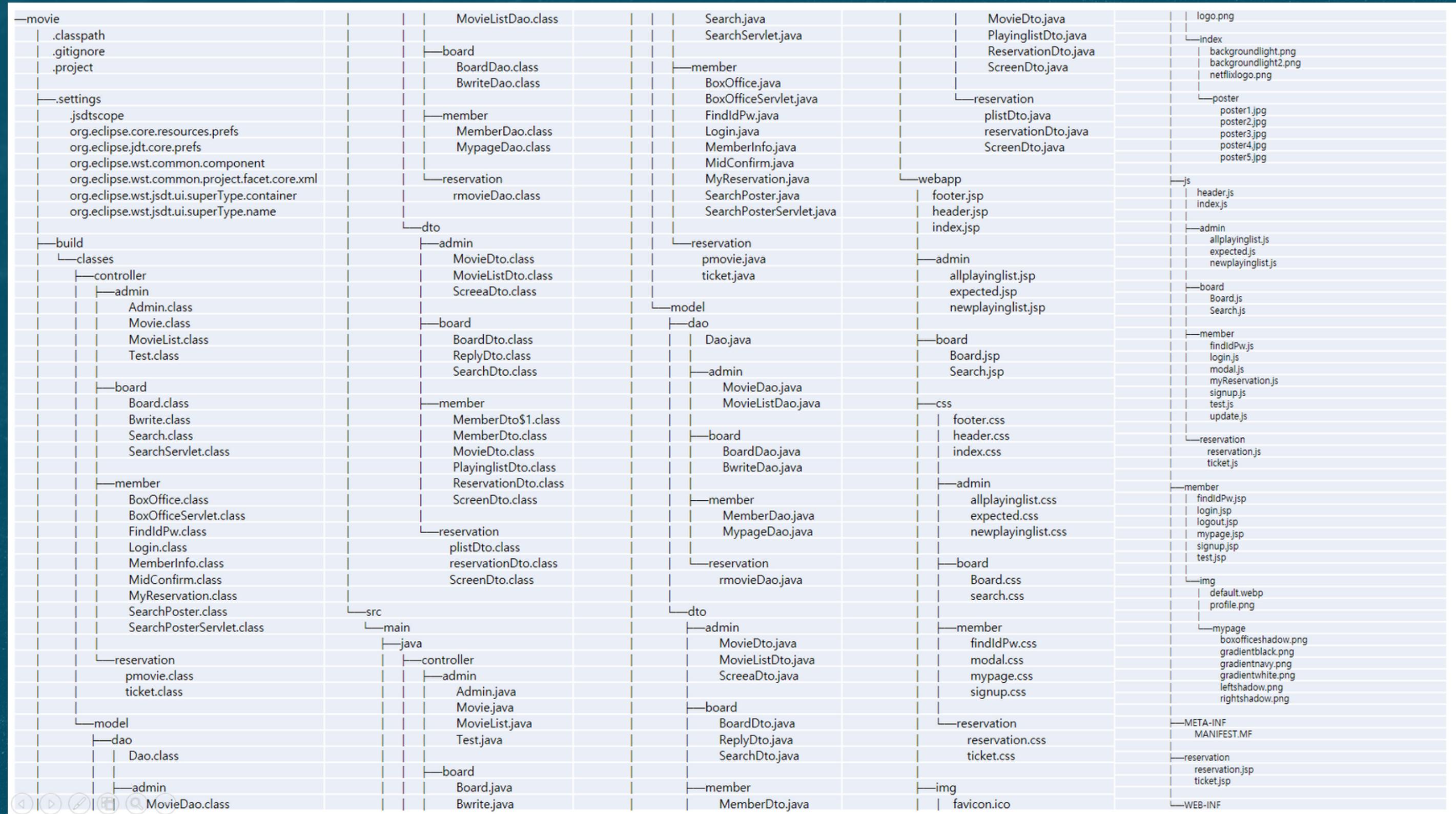
- 3/23 | 주제선정 , DB설계
- 3/24 | 기초 MVC 설계 , 기초 DAO,DTO 생성
- 3/27 | 로고제작,색상테마,폰트선정
- 3/28 | 프로필이미지,드롭다운,박스오피스css
- 3/29 | 아이디찾기. 비밀번호찾기 이메일발송
- 3/30 | 회원정보수정, 회원탈퇴, 회원가입 유효성검사
- 3/31 | 메인페이지 전날 박스오피스 출력(크롤링)
- 4/03 | 이미지슬라이더
- 4/04 | 마이페이지 예매내역출력
- 4/05 | 예매내역css , 예매내역 영화포스터출력(크롤링)
- 4/06 | 예매내역 슬라이더 , 메인페이지 css
- 4/07 | 넷플릭스10출력(크롤링,슬라이더구현) , 개봉예정작출력(API)
- 4/10 | 통합 테스트 및 보수
- 4/11 | 보수 및 발표준비

MYSQL ER 다이어그램 구성도

member / movie / playlist / screen
reservation / board / reply



MVC 구조도

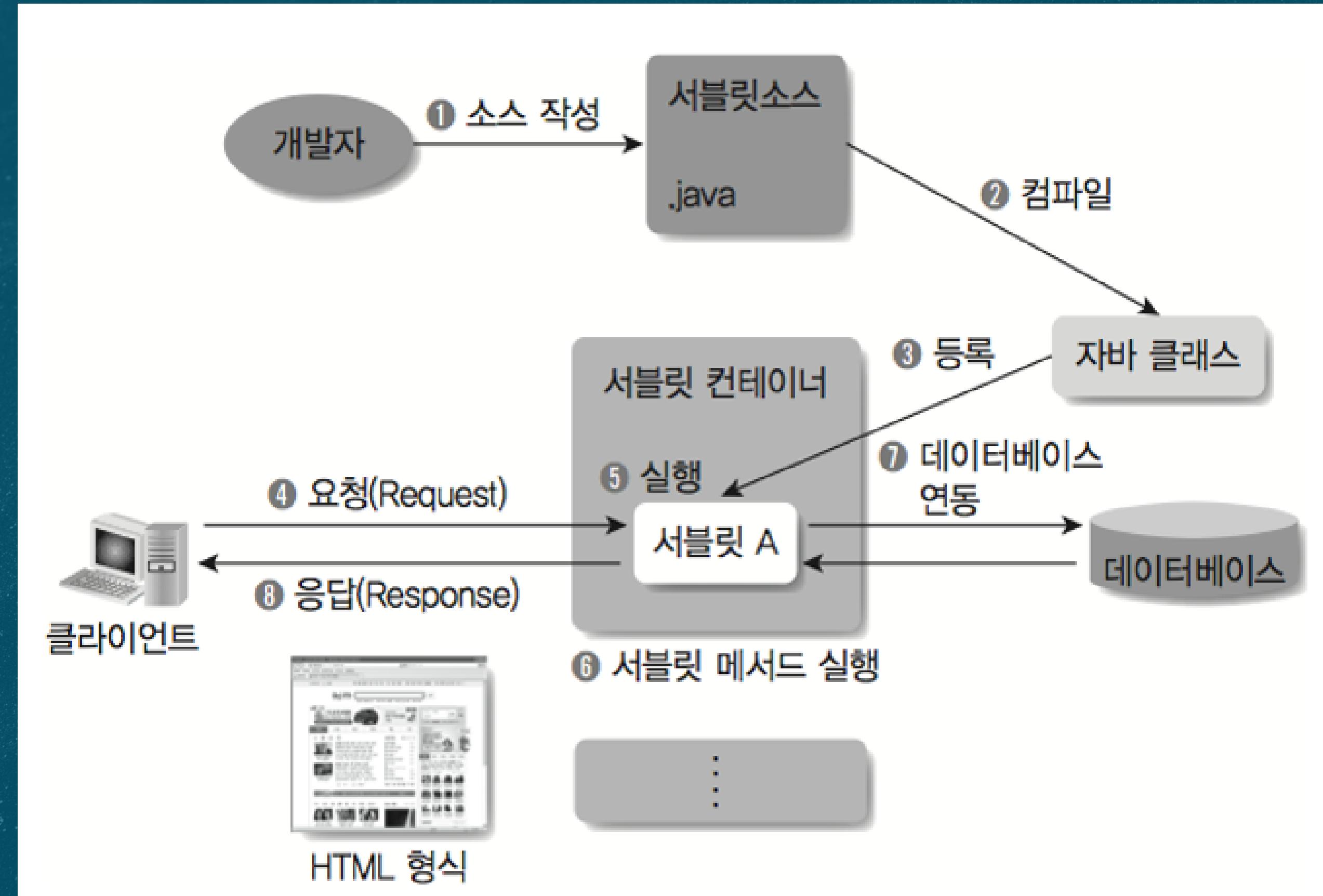


컨트롤 구조도

CINEVERSE



서블릿



첫실행시

```
@WebServlet("/Board/comment")
public class Board extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Board() {
        System.out.println("board/comment 실행");
    }

    public void init(ServletConfig config) throws ServletException{
        System.out.println("서블릿 init 실행");
    }

    public void destroy() {
        System.out.println("서블릿 destroy 실행");
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        response.setCharacterEncoding("UTF-8");

        int type = Integer.parseInt(request.getParameter("type"));
        System.out.println("doget 호출중");
    }
}
```

board/comment 실행
서블릿 init 실행
doget 호출중
연동성공

요청이 끝나면 Thread Pool에 이동하거나 종료(GC)

```
dto : MemberDto [mno=4, mid=zxcvb, mpwd=null, memail=admin@admin.com, mimg=null]
json : {"mno":4,"mid":"zxcvb","mpwd":null,"memail":"admin@admin.com","mimg":null}
doget 호출중
mno : 4
dto : MemberDto [mno=4, mid=zxcvb, mpwd=null, memail=admin@admin.com, mimg=null]
json : {"mno":4,"mid":"zxcvb","mpwd":null,"memail":"admin@admin.com","mimg":null}
doget 호출중
mno : 4
dto : MemberDto [mno=4, mid=zxcvb, mpwd=null, memail=admin@admin.com, mimg=null]
json : {"mno":4,"mid":"zxcvb","mpwd":null,"memail":"admin@admin.com","mimg":null}
doget 호출중
mno : 4
dto : MemberDto [mno=4, mid=zxcvb, mpwd=null, memail=admin@admin.com, mimg=null]
json : {"mno":4,"mid":"zxcvb","mpwd":null,"memail":"admin@admin.com","mimg":null}
doget 호출중
```

서블릿의 코드가 변경되거나 GC가 발생하면 `destroy()` 실행

```
4월 11, 2023 2:41:56 오후 org.apache.catalina.core.StandardContext reload
정보: 이름이 [/movie]인 컨텍스트를 다시 로드하는 작업이 시작되었습니다.

서블릿 destroy 실행
#월 11, 2023 2:41:56 오후 org.apache.catalina.loader.WebappClassLoaderBase clearReferencesJdbc
경고: 웹 애플리케이션 [movie]이(가) JDBC 드라이버 [com.mysql.cj.jdbc.Driver]을(를) 등록했지만, 웹 애플리케이션이 종지될 때, 해당 JDBC 드라이버의 등록을 제거하지 못했습니다.
4월 11, 2023 2:41:56 오후 org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads
경고: 웹 애플리케이션 [movie]이(가) [mysql-cj-abandoned-connection-cleanup](이)라는 이름의 쓰레드를 시작시킨 것으로 보이지만, 해당 쓰레드를 종지시키지 못했습니다. 이는 메.
java.base@17.0.5/java.lang.Object.wait(Native Method)
java.base@17.0.5/java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:155)
com.mysql.cj.jdbc.AbandonedConnectionCleanupThread.run(AbandonedConnectionCleanupThread.java:91)
java.base@17.0.5/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136)
java.base@17.0.5/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)
java.base@17.0.5/java.lang.Thread.run(Thread.java:833)
4월 11, 2023 2:41:57 오후 org.apache.jasper.servlet.TldScanner scanJars
정보: 적어도 하나의 JAR가 TLD들을 찾기 위해 스캔되었으나 아무 것도 찾지 못했습니다. 스캔했으나 TLD가 없는 JAR들의 전체 목록을 보시려면, 로그 레벨을 디버그 레벨로 설정하십시오. 스캔 과정
4월 11, 2023 2:41:57 오후 org.apache.catalina.core.StandardContext reload
정보: 이름이 [/movie]인 컨텍스트를 다시 로드하는 것을 완료했습니다.

4월 11, 2023 2:41:58 오후 org.apache.catalina.loader.WebappClassLoaderBase checkStateForResourceLoading
정보: 불허되는 접근: 이 웹 애플리케이션 인스턴스는 이미 종지되었습니다. []을(를) 로드할 수 없습니다. 디버그 목적 및 불허되는 접근을 발생시킨 해당 쓰레드를 종료시키기 위한 시도로서, 다음 스
java.lang.IllegalStateException: 불허되는 접근: 이 웹 애플리케이션 인스턴스는 이미 종지되었습니다. []을(를) 로드할 수 없습니다. 디버그 목적 및 불허되는 접근을 발생시킨 해당 쓰
    at org.apache.catalina.loader.WebappClassLoaderBase.checkStateForResourceLoading(WebappClassLoaderBase.java:1432)
    at org.apache.catalina.loader.WebappClassLoaderBase.getResource(WebappClassLoaderBase.java:1057)
    at com.mysql.cj.jdbc.AbandonedConnectionCleanupThread.checkThreadContextClassLoader(AbandonedConnectionCleanupThread.java:123)
    at com.mysql.cj.jdbc.AbandonedConnectionCleanupThread.run(AbandonedConnectionCleanupThread.java:90)
    at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136)
    at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)
    at java.base/java.lang.Thread.run(Thread.java:833)
```

종료된 상태라면 요청시 새로운 인스턴스를
생성하고 init() 실행

```
java.lang.IllegalStateException: 불허되는 접근: 이 웹 애플리케이션 인스턴스는 이미 종지되었습니다. []를(를) 로드할 수 없습니다. 디버그 목적 및 불허되는 접근을 발생시킨 해당
        at org.apache.catalina.loader.WebappClassLoaderBase.checkStateForResourceLoading(WebappClassLoaderBase.java:1432)
        at org.apache.catalina.loader.WebappClassLoaderBase.getResource(WebappClassLoaderBase.java:1057)
        at com.mysql.cj.jdbc.AbandonedConnectionCleanupThread.checkThreadContextClassLoader(AbandonedConnectionCleanupThread.java:123)
        at com.mysql.cj.jdbc.AbandonedConnectionCleanupThread.run(AbandonedConnectionCleanupThread.java:90)
        at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136)
        at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)
        at java.base/java.lang.Thread.run(Thread.java:833)
```

mno : 4

연동성공

dto : MemberDto [mno=4, mid=zxcvb, mpwd=null, memail=admin@admin.com, mimg=null]
json : {"mno":4, "mid":"zxcvb", "mpwd":null, "memail":"admin@admin.com", "mimg":null}

board/commet 실행

서블릿 init 실행

doget 호출중

연동성공

주요기능

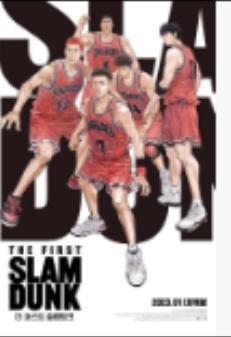
더 퍼스트 슬램덩크



스즈메의 문단속



웅남이



더 퍼스트 슬램덩크



2023년 04월 10일

4월

10

오늘

< >

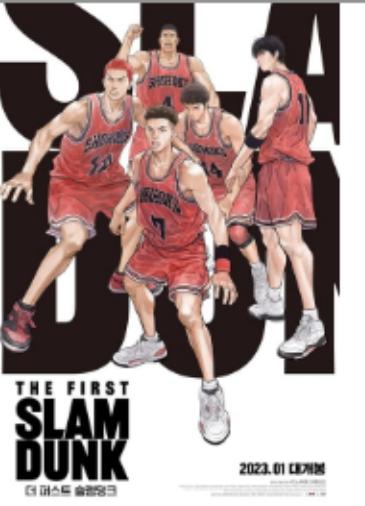
11 화 수 12 13 목 14 금 15 토 16 일

더 퍼스트 슬램덩크

22:00

110 / 110 2관

인원/좌석 선택



더 퍼스트 슬램덩크

2관

2023-04-10 | 22:00

성인

+ 0 -

청소년

+ 0 -

시니어

+ 0 -

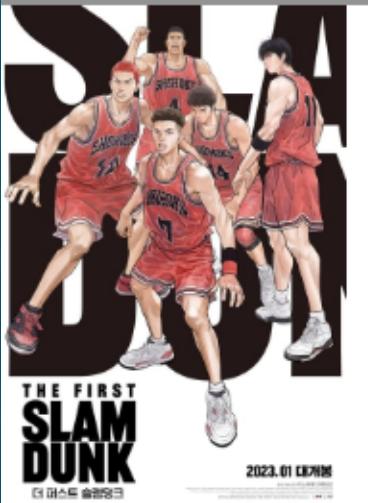
S C R E E N

a	a1	a2	a3	a4
b	b1	b2	b3	b4
c	c1	c2	c3	c4
d	d1	d2	d3	d4
e	e1	e2	e3	e4
f	f1	f2	f3	f4
g	g1	g2	g3	g4
h	h1	h2	h3	h4
i	i1	i2	i3	i4

a5	a6	a7	a8	a9	a10	a11	a12
b5	b6	b7	b8	b9	b10	b11	b12
c5	c6	c7	c8	c9	c10	c11	c12
d5	d6	d7	d8	d9	d10	d11	d12
e5	e6	e7	e8	e9	e10	e11	e12
f5	f6	f7	f8	f9	f10	f11	f12
g5	g6	g7	g8	g9	g10	g11	g12
h5	h6	h7	h8	h9	h10	h11	h12
i5	i6	i7	i8	i9	i10	i11	i12

a13	a14	a15	a16
b13	b14	b15	b16
c13	c14	c15	c16
d13	d14	d15	d16
e13	e14	e15	e16
f13	f14	f15	f16
g13	g14	g15	g16
h13	h14	h15	h16
i13	i14	i15	i16

인원/좌석 선택



더 퍼스트 슬램덩크

크

2023-04-10 | 22:00

성인

+ 2 -

청소년

+ 2 -

시니어

+ 0 -

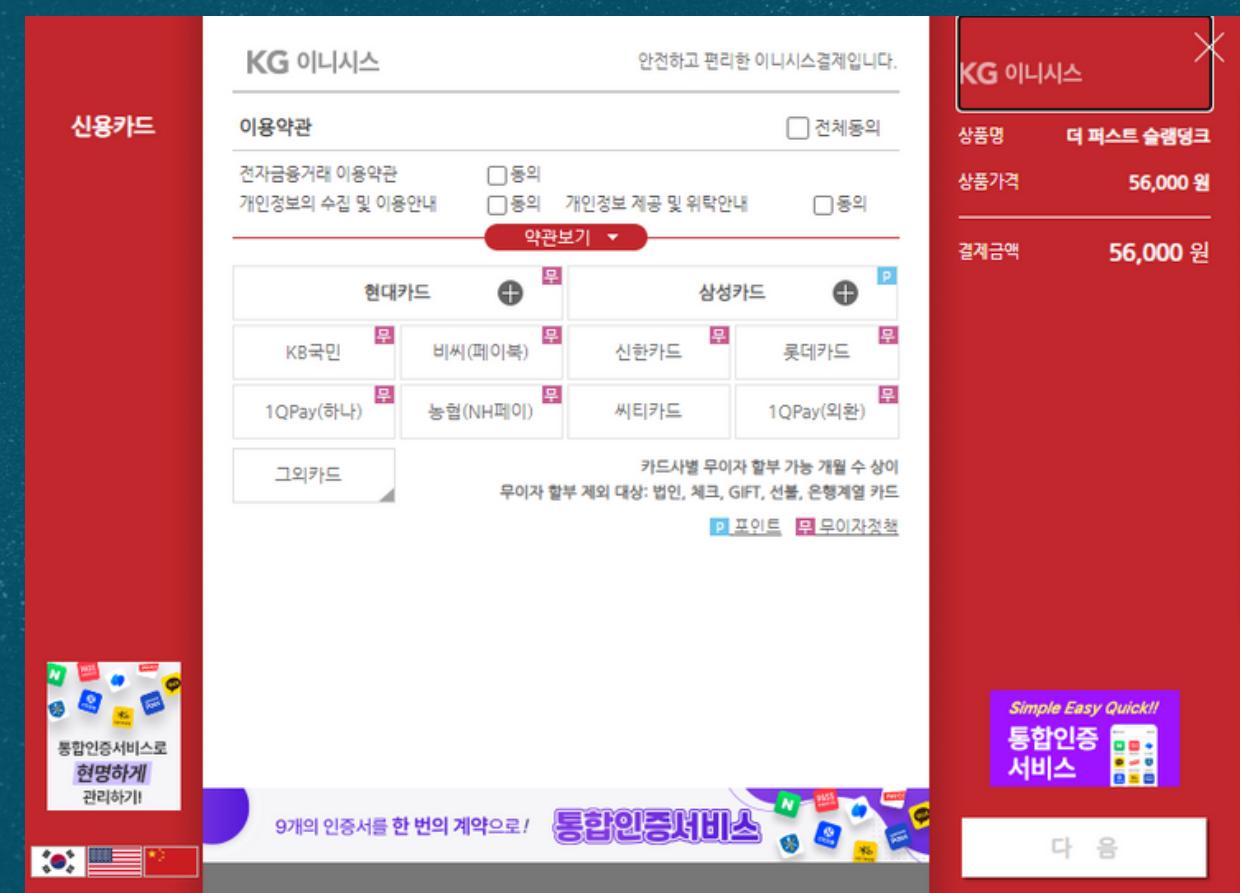
2관

S C R E E N

a	a1	a2	a3	a4
b	b1	b2	b3	b4
c	c1	c2	c3	c4
d	d1	d2	d3	d4
e	e1	e2	e3	e4
f	f1	f2	f3	f4
g	g1	g2	g3	g4
h	h1	h2	h3	h4
i	i1	i2	i3	i4

a5	a6	a7	a8	a9	a10	a11	a12
b5	b6	b7	b8	b9	b10	b11	b12
c5	c6	c7	c8	c9	c10	c11	c12
d5	d6	d7	d8	d9	d10	d11	d12
e5	e6	e7	e8	e9	e10	e11	e12
f5	f6	f7	f8	f9	f10	f11	f12
g5	g6	g7	g8	g9	g10	g11	g12
h5	h6	h7	h8	h9	h10	h11	h12
i5	i6	i7	i8	i9	i10	i11	i12

a13	a14	a15	a16
b13	b14	b15	b16
c13	c14	c15	c16
d13	d14	d15	d16
e13	e14	e15	e16
f13	f14	f15	f16
g13	g14	g15	g16
h13	h14	h15	h16
i13	i14	i15	i16





상영리스트 출력

AJAX

서블릿

DAO

```

let pmovie=[]; //상영중인 전체 영화
$.ajax({
  url:"/movie/playing/moive",
  method:"get",
  success:(r)=>{
    let html = ``;
    pmovie=r; //모든 상영중 영화를 저장후 출력
    pmovie.forEach((o)=>{
      html += `<div onclick="select_Pmovie(${o.mno})" class="movie_select_${o.mno} movie_select_one">
        <div class="movie_img_${o.mno} movie_img"></div>
        <div class="one_movie_title m_${o.mno}"> ${o.title} </div>
      </div>`;
    })
    document.querySelector('.movie_title').innerHTML = html;
    img_print() // 상영중인 영화를 먼저 추출한뒤 영화제목을 뽑아야함
  })
})
}
  
```

```

//상영중인 영화 제목 출력
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
  ArrayList<plistDto> result = rmovieDao.getInstance().Playing_movie();
  System.out.println("result : "+result);
  ObjectMapper objMapper = new ObjectMapper();

  String jsonArray = objMapper.writeValueAsString(result);

  response.setCharacterEncoding("UTF-8");
  response.setContentType("application/json");
  response.getWriter().print(jsonArray );
}
  
```

```

//현재 상영중인 영화 리스트
public ArrayList<plistDto> Playing_movie(){

  /* ----- 상영중인 영화 리스트 -----*/
  String sql = "select m.mno, m.title from movie m natural join playinglist p where pstate =1 group by mno ";
  ArrayList<plistDto> plist = new ArrayList<>();

  try {
    ps=con.prepareStatement(sql);
    rs=ps.executeQuery();

    while (rs.next()) {
      // 상영중인 영화 리스트
      plist.add( new plistDto(rs.getInt(1), rs.getString(2) ) );
    }//while e

    return plist;
  } catch (SQLException e) {System.err.println(e);}
  return null;
}
  
```



```

/* -----날짜 구하기 ----- */
//현재 년도, 월, 일 구하기
let year = (new Date()).getFullYear()
let month = new Date().getMonth() //0~11 => 1~12
let date = new Date().getDate();

//말일, 요일
let last = new Date(year,month+1,0).getDate(); //+1인체로
let week = ['일', '월', '화', '수', '목', '금', '토'];
let week_day = week[new Date(year,month,date).getDay()]

let monthCount = 0;
let month_kor = month+1+'월'; // 최초 '0월' 한번은 뜨고 시작
/*----- 달력 함수 -----*/
//현재날짜에서 하루씩 증가하는 함수
function dateCalculate(){
    date++;
    if(date > last){ //말일이 지나면
        date = 1; //1일부터
        monthCount++; //
        month+= monthCount;
        month_kor = month+1+'월'

        if(month>=13){ // 13월이된다면 1년++
            month%=12;
            year++;
        }
    }else{ //2일부턴 월이 안나오게
        month_kor = '';
    }
    week_day = week[new Date(year,month,date).getDay()];
}

```

```

//날짜 출력
dayPrint()
function dayPrint(){

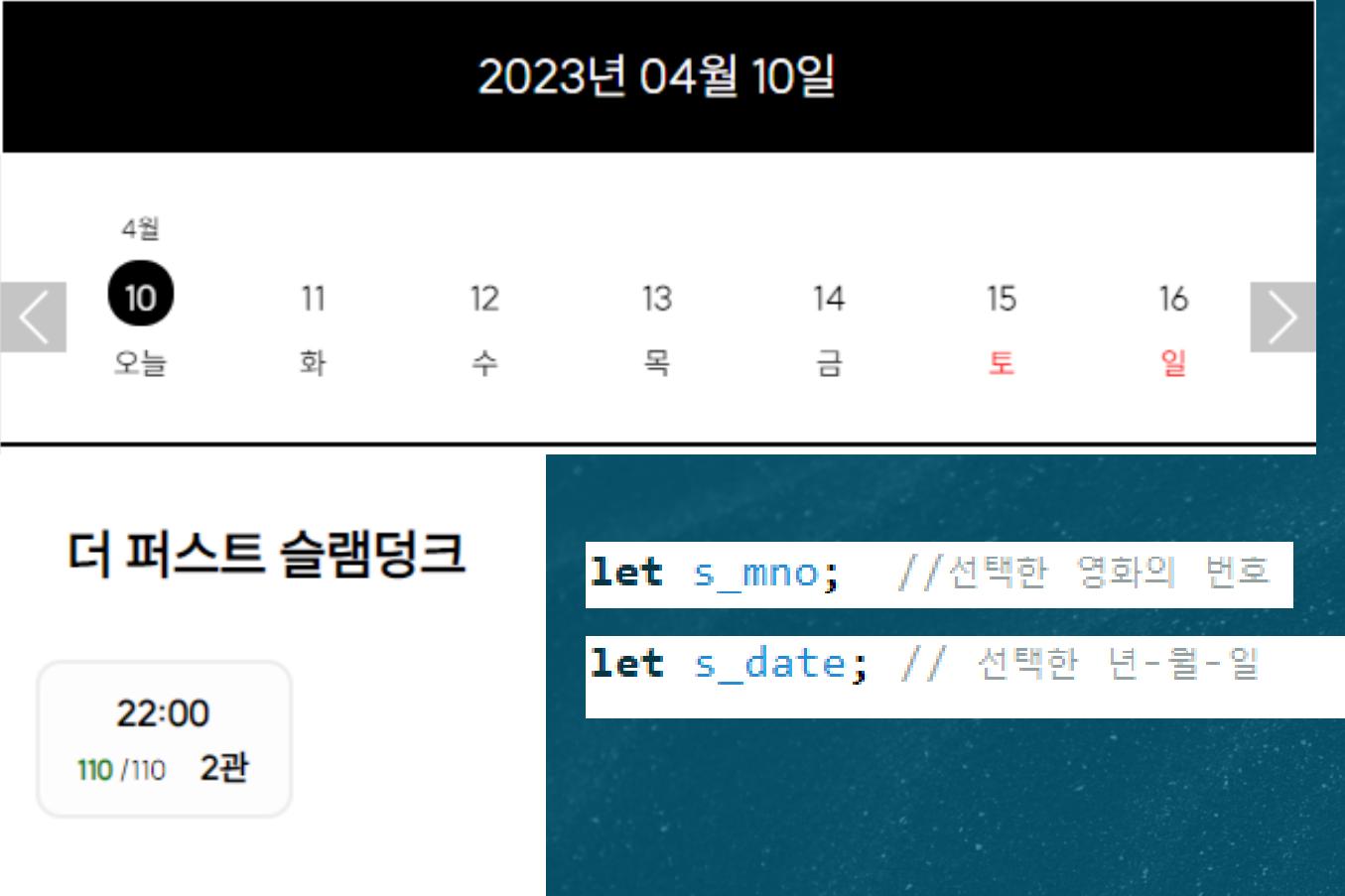
    let html = `<div class="carousel-inner">`;

    for(let i = 0 ; i < 3 ; i++){ //총 3주 출력
        if( i == 0 ){
            html += `<div class="carousel-item active">
                <ul class="date_week_li">`;
        }else{
            html += `<div class="carousel-item">
                <ul class="date_week_li">`;
        }
        for(let j = 0 ; j < 7 ; j++){ // 한번에 7일만 보이게

            html+= `<li onclick="select_date(${year}, ${month}, ${date})">
                <span class="date_month">${month_kor}</span>
                <span class="date_day D${month}_${date}"> ${date} </span>
                <span class="date_week_day" ${week_day === '토'|| week_day === '일' ? 'style="color:red"' : 'style="color:black'}>
                    ${date == new Date().getDate() ? '오늘' : week_day}
                </span>
            </li>`;
            dateCalculate(); // 1일씩 증가하는 함수
        }
        html +=`</ul>
            </div>` //ul과 캐러셀 하나 닫기
    }
    html +=`</div>
        <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="prev">
            <span class="carousel-control-prev-icon" aria-hidden="true"></span>
            <span class="visually-hidden">Previous</span>
        </button>
        <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="visually-hidden">Next</span>
        </button>
    `;

    document.querySelector('#carouselExampleControls').innerHTML = html;
}

```



AJAX

```
----- 상영관 출력 -----
function screen_print(){
    let html = ``;

    document.querySelector('.s_movie_title').innerHTML = s_movie

    $.ajax({
        url: "/movie/playing/moive",
        method: "put",
        data: { mno:s_mno , s_date:s_date },
        success:(r)=>{
            console.log(r)
            if(r==null){html=<div class="none_date">다른 날짜를 확인해주세요</div>}
            else{
                //이용가능한 좌석이 0 이면
                html += `<div class="date_box" ${ r.aseat>0 ? 'onclick="select_screen()"' : 'onclick="no()"' }> <!-- 상영관 박스 -->
                    <div class="date_time">${r.playtime.split(' ')[1].substr(0,5)}</div> <!-- 상영시간 -->
                    <div class="date_text">
                        <span class="date_seat">${r.aseat}</span>/${r.s_seat}</span> <!-- 남은좌석 -->
                        <span class="date_sno">${r.sno}</span> <!-- 상영관 위치 -->
                    </div>
                </div>
            }
            a_seat = r.aseat; seat = r.s_seat; s_sno = r.sno; s_pno = r.pno;
        }
        document.querySelector('.date_form').innerHTML = html
    })//success e
}
}

//선택한 영화+달력의 날짜는 상영관 출력
protected void doPut(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setCharacterEncoding("UTF-8");
    request.setCharacterEncoding("UTF-8");

    int mno = Integer.parseInt( request.getParameter("mno") );
    String s_date = request.getParameter("s_date");

    plistDto result = rmovieDao.getInstance().screen_print(mno,s_date );
    System.out.println("click result : "+result);

    ObjectMapper objMapper = new ObjectMapper();
    String jsonArray = objMapper.writeValueAsString(result);

    response.setCharacterEncoding("UTF-8");
    response.setContentType("application/json");
    response.getWriter().print(jsonArray );
}
```

서블릿

```
//상영관 정보 출력
public plistDto screen_print(int mno, String s_date){

    String sql ="select p.playtime , s.seat - p.pseat , s.sno ,s.seat , p.pno"
        + " from movie m natural join playinglist p natural join screen s "
        + " where mno = "+mno +" and playtime like '%"+s_date+"%'";
    try {
        ps=con.prepareStatement(sql);
        rs = ps.executeQuery();
        while(rs.next()) {
            plistDto pDto = new plistDto(
                rs.getString(1), rs.getInt(2), rs.getInt(3),
                rs.getInt(4) , rs.getInt(5));
            return pDto;
        }
    }catch(Exception e) {System.err.println(e);}

    return null;
}
```

DAO

```
//상영관 정보 출력
public plistDto screen_print(int mno, String s_date){

    String sql ="select p.playtime , s.seat - p.pseat , s.sno ,s.seat , p.pno"
        + " from movie m natural join playinglist p natural join screen s "
        + " where mno = "+mno +" and playtime like '%"+s_date+"%'";
    try {
        ps=con.prepareStatement(sql);
        rs = ps.executeQuery();
        while(rs.next()) {
            plistDto pDto = new plistDto(
                rs.getString(1), rs.getInt(2), rs.getInt(3),
                rs.getInt(4) , rs.getInt(5));
            return pDto;
        }
    }catch(Exception e) {System.err.println(e);}

    return null;
}
```

/MOVIE/RESERVATION/TICKET.JSP?PNO=3

```
<%
    int pno = Integer.parseInt( request.getParameter("pno"));
%>
<input type="hidden" value=<%=pno%> class="pno">
```

인원/좌석 선택



더 퍼스트 슬램덩크
2023-04-10 | 22:00 2관

성인 + 2 - 청소년 + 2 - 시니어 + 0 -

SCREEN

a	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	a16
b	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15	b16
c	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16
d	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15	d16
e	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	e13	e14	e15	e16
f	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16
g	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	g13	g14	g15	g16
h	h1	h2	h3	h4	h5	h6	h7	h8	h9	h10	h11	h12	h13	h14	h15	h16
i	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	i15	i16

총 합계 56,000원 결제하기

```
/*-----좌석 선택란-----*/
select_screen();
function select_screen(){
    let html = '';
    for(index = 97; index<=105 ; index++ ){ //아스키코드 a~i //fromCharCode(number) 숫자를 아스키코드로
        html += `<div class="seat_text">
            <span class="col"> ${String.fromCharCode(index)} </span>

            <div class="left">
                for(let i =1; i <= 4 ; i++){
                    html += `<span class="seat" id="s${index}_${i}" onclick="click_seat(${index}, ${i})">
                        ${String.fromCharCode(index)+i} </span>
                }
                html += `</div>

                html += `<div class="center">
                    for(let j =5; j <= 12 ; j++){
                        html += `<span class="seat" id="s${index}_${j}" onclick="click_seat(${index}, ${j})">
                            ${String.fromCharCode(index)+j} </span>
                    }
                html += `</div>

                html += `<div class="right">

```

인원수 체크

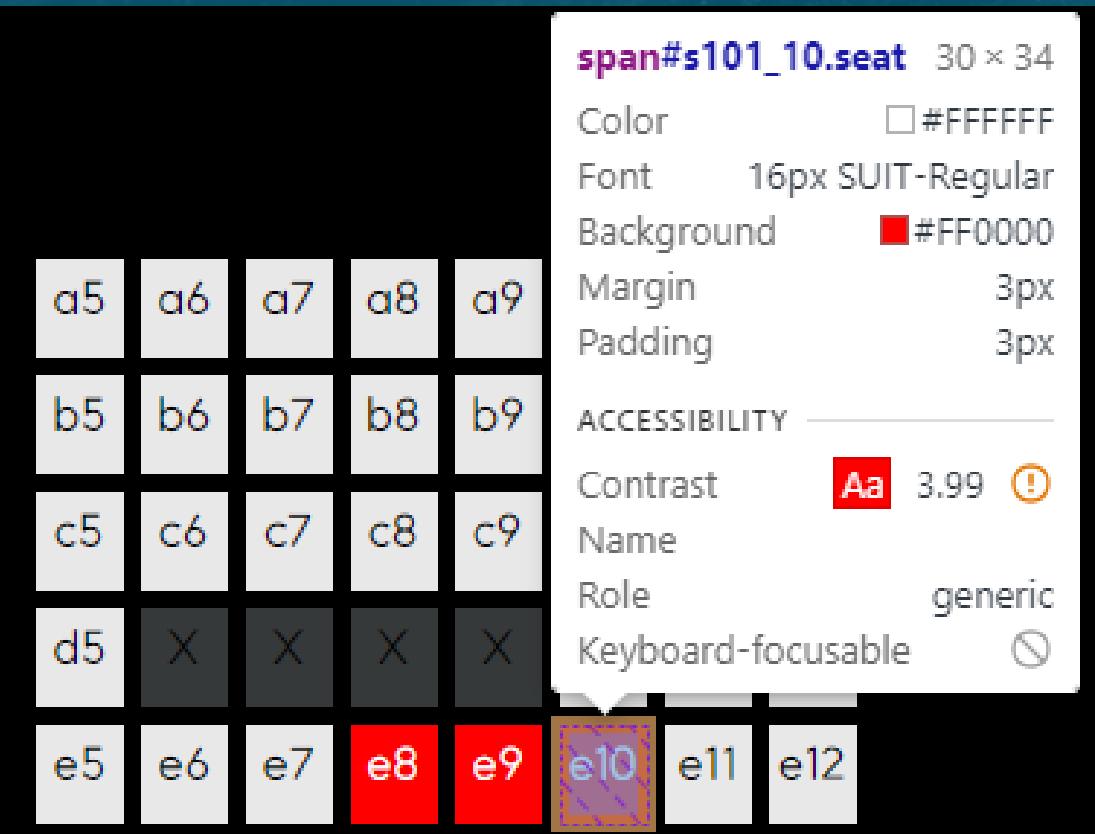


```
let a_num = 0 // adult num 성인 인원  
let t_num = 0 // teen num 청소년 인원  
let b_num = 0 // baby num 시니어 인원  
  
let seat_num=0 // 선택한 좌석
```

```
/* 인원 유효성 검사 (최대 8명 )*/  
function p_max(){  
    if(a_num+t_num+b_num >= 8){  
        alert('총 8명까지 예매할 수 있습니다!');  
        return;  
    }  
}
```

```
/* ----- 영화 인원 증가 ----- */  
function p_up(human){  
    p_max()  
    if(human == 1){  
        a_num++  
        document.querySelector('.a_num').innerHTML= a_num;  
    }  
    else if (human == 2){  
        t_num++;  
        document.querySelector('.t_num').innerHTML= t_num;  
    }  
    else if (human == 3 ){  
        b_num++;  
        document.querySelector('.b_num').innerHTML= b_num;  
    }  
    pay_money() // 결제금액 계산  
}  
  
/* ----- 영화 인원 감소 ----- */  
function p_down(human){  
  
    if(human == 1){  
        if(a_num == 0 ){  
            alert('0명 이하로 감소할 수 없습니다.'); return;  
        }  
        a_num--  
        document.querySelector('.a_num').innerHTML= a_num;  
    }  
    else if (human == 2){  
        if(t_num == 0 ){  
            alert('0명 이하로 감소할 수 없습니다.'); return;  
        }  
        t_num--;  
        document.querySelector('.t_num').innerHTML= t_num;  
    }  
    else if (human == 3 ){  
        if(b_num == 0 ){  
            alert('0명 이하로 감소할 수 없습니다.'); return;  
        }  
        b_num--;  
        document.querySelector('.b_num').innerHTML= b_num;  
    }  
    pay_money() // 결제금액 계산  
}
```

좌석 선택과 해제 CSS, 전역변수에 선택좌석 담기 이미 예매된 좌석 막기, 좌석예매



```

let seatString = ''
s_seat.forEach( (s,i) =>{
    console.log("좌석: "+s + " 인덱스: "+ i)
    seatString += s ;
    if( i == ( s_seat.length-1) ){
    } else{
        seatString += ",";
    }
})

$.ajax({
    url:"/movie/ticket",
    method:"post",
    data: { "pno" : pno , "s_seat" : seatString },
    async:"false",
    success:(r)=>{
        if(r=='true'){
            alert('예약되었습니다! ')
            location.href=`/movie/member/mypage.jsp?page=0`
        }
    }//success
})//$.ajax e
}//else e
})// rsp

```

```

let pno = document.querySelector('.pno').value
let s_seat=[]; // 선택한 좌석 select_seat

```

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    //로그인 세션을 int로
    Object Omno = request.getSession().getAttribute("login");
    int mno = Integer.parseInt((String.valueOf(Omno)));

    //상영번호
    int pno = Integer.parseInt( request.getParameter("pno") );

    //좌석
    String s_seat = request.getParameter("s_seat");
    String[] seat = s_seat.split(",");

    boolean result = rmovieDao.getInstance().reservation_ticket(seat, mno, pno);

    response.getWriter().print(result);
}

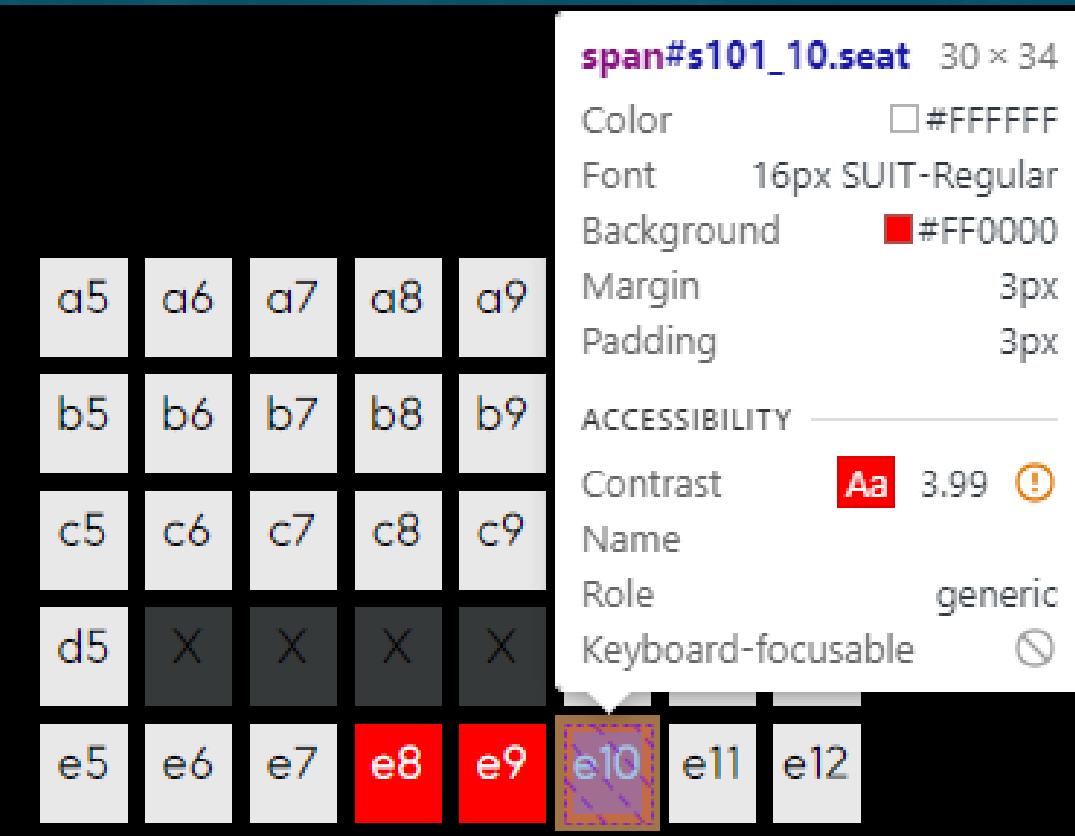
//결제하기
public boolean reservation_ticket(String seatnum[], int mno, int pno) {
    //좌석을 2개이상 예약할 수 있으므로 같은 멤버, 같은 상영관에 다른좌석만 insert
    String sql = "";
    try {

        for(int i = 0 ; i <= seatnum.length-1 ; i++) {
            sql = " insert into reservation(seatnum,mno,pno) "
                  + " values ('"+seatnum[i] +"', "+mno+", "+pno+")";
            ps=con.prepareStatement(sql);
            ps.executeUpdate();
        }
        //예약테이블 insert후 상영중인영화 테이블 좌석 업데이트
        sql = "update playinglist set pseat=pseat+"+seatnum.length+" where pno = "+pno;

        ps=con.prepareStatement(sql);
        ps.executeUpdate();

        return true;
    }catch(Exception e) {System.err.println(e);}
    return false;
}

```



```
//----- 예약된 좌석 출력 -----
```

```
function r_seat(){
    $.ajax({
        url:"/movie/ticket",
        method:"put",
        data: {"pno":pno},
        async:"false",
        success:(r)=>{

            //예약된 좌석은 비활성화
            r.forEach((o,i)=>{
                document.querySelector(`#${o.seatnum.substr(0,1).charCodeAt(0)}_${o.seatnum.substr(1,2)}`).disabled = true;
                document.querySelector(`#${o.seatnum.substr(0,1).charCodeAt(0)}_${o.seatnum.substr(1,2)})`).style.backgroundColor = "#363939";
                document.querySelector(`#${o.seatnum.substr(0,1).charCodeAt(0)}_${o.seatnum.substr(1,2)})`).style.cursor = "default";
                document.querySelector(`#${o.seatnum.substr(0,1).charCodeAt(0)}_${o.seatnum.substr(1,2)})`).onclick = null;
                document.querySelector(`#${o.seatnum.substr(0,1).charCodeAt(0)}_${o.seatnum.substr(1,2)})`).innerHTML = "X";
            })
        }
    })
}
```

느낀점 , 보안할 점

권태형

느낀점

팀원들이 API 및 크롤링등으로 작성한 메소드에 각자 값을 넣어주면 원하는 정보를 가져가고 가져와 사용할 수 있다는 것이 팀원들과의 프로젝트에서 굉장히 중요한 부분이라고 느꼈다

보안할 점

상영정보에 영화당 상영시간을 추가하여 상영관, 상영시간에 겹치는 것이 없게하여 겹치면 상영 정보 등록 불가, 겹치지 않으면 상영 정보 등록 가능으로 구현

이경석

느낀점

해당 기능에대한 작동원리와 오류 등 내가 몰랐던 부분을 같이 고민하고 해결하는 방향이 좋았으며 DB설계와 팀원간의 소통이 무엇보다 중요한지 깨달았다.

보안할 점

처음 크롤링, 외부 api 기능을 사용하고, 좌석 기능을 만드는데 시간을 많이 소모하였으며 jsoup에 대한 관심이 많아졌다.

이정준

느낀점

협업을 하면서 의견을 조율하고 소통하는 것이 즐거웠고 크롤링 구현에서 작업이 늦어졌는데 협업과 일정을 생각해서 더 빠르게 코딩할 수 있도록 공부를 더 해야겠다고 생각했다.

보안할 점

Jsoup라이브러리를 사용했는데 사용법에 익숙해지기까지 시간이 꽤 걸렸다. 코드를 보는 눈을 길러서 다음 새로운 라이브러리의 사용시 더 빠르게 파악하고 싶다.
그리고 크롤링으로 불러온 이미지의 로딩이 상당히 길었는데 이 부분을 보안하고싶다.

유슬비

느낀점

서로 배려해주신 덕분에 탈없이 잘 마무리 된 것 같다.
팀원들에게 새롭게 많은 것들을 배울 수 있어서 좋았다.

보안할 점

소켓을 사용하여 서블릿의 속도를 보완하면 좋을 것 같다.

THANK YOU
VERY MUCH!