

WebApp PJT

2019.03

손찬욱

목차

1. 개요

2. 상세설계

3. 설계하기

4. 서버 사이드 작업

5. 프론트 엔드 작업

개요

목적

- 간단한 **영화 후기 관리** 웹앱을 만들어 봅니다.
- 이를 통해 웹에서의 데이터 전송과 저장에 대한 전반적인 처리를 실습해 봅니다.
- 서버사이드에서는 DB설계 과정부터 데이터 저장과 출력을 배우고, 프론트 엔드에서는 데이터 전송/요청 및 UI 구성에 대한 기본적인 개념을 습득할 수 있습니다.

상세설계

내용

- 총 6개의 값을 입력 받는다.
 - 영화 제목
 - 영화 평점
 - 영화 후기
 - 영화 사이트 링크 (선택)
 - 이미지 (선택)
- 값을 서버에 전송해 DB에 저장한다. 이 때 등록시간을 함께 등록한다.
- 저장된 영화 후기 글의 목록을 출력한다.
- 영화 후기 글의 제목을 클릭하면 영화 후기 글의 내용이 출력한다.
- 후기 등록 버튼을 통해 영화 후기를 등록할 수 있다.

영화관람 후기

등록



완벽한 타인

★★★★★

등록일: 2018-10-28 15:27:42

[관련 링크](#)

[수정](#) [삭제](#)



벽 속에 숨은 마법시계

★★★★★

등록일: 2018-10-28 15:09:21

[수정](#) [삭제](#)

영화관람 후기

등록

제목

평점

0

후기

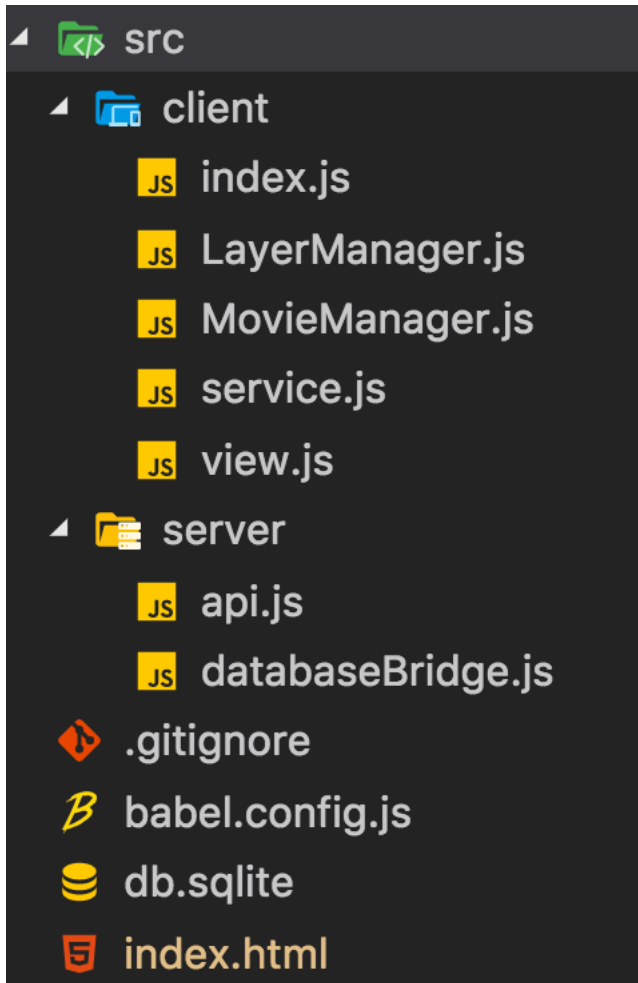
참고URL

이미지

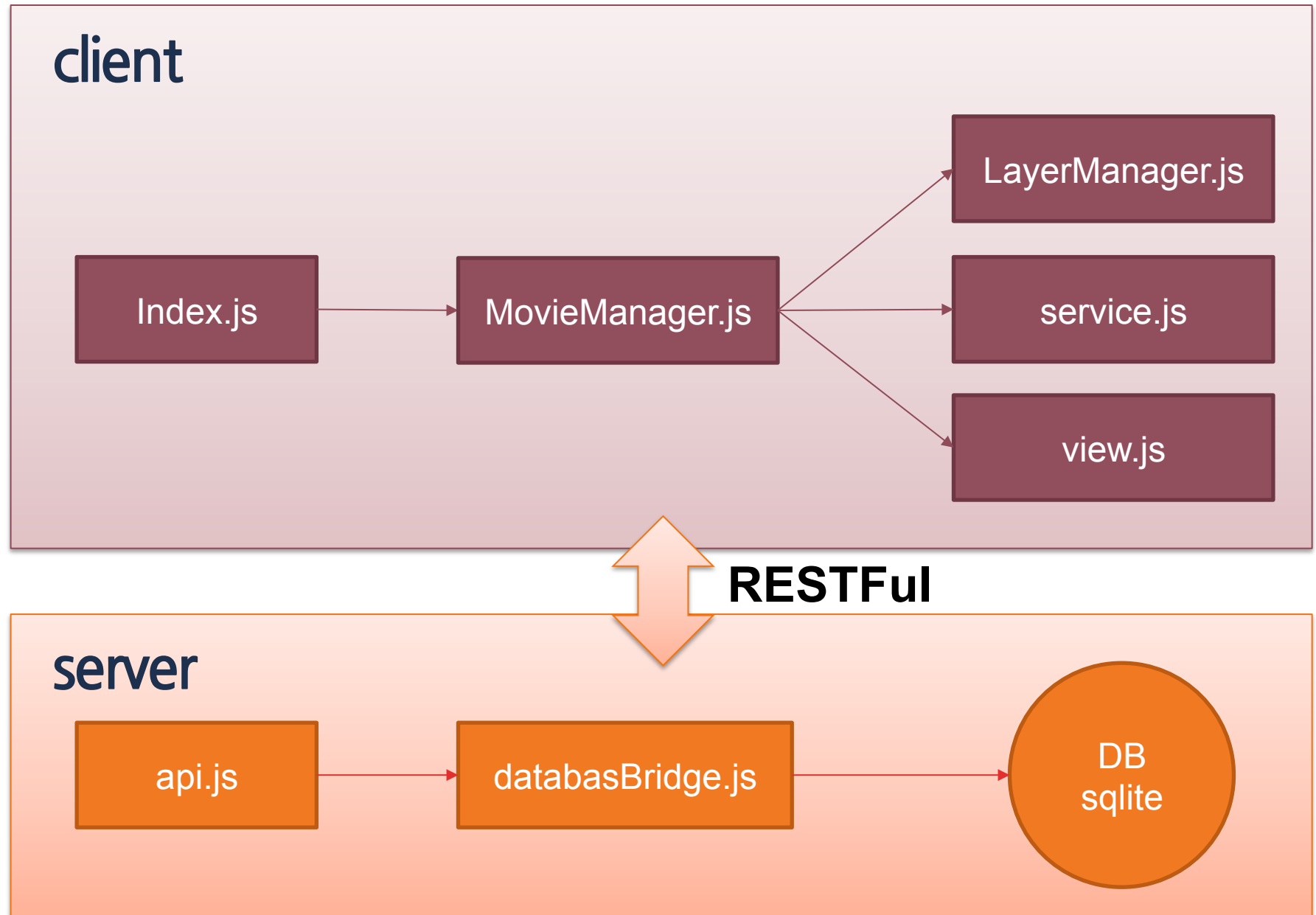
[파일 선택](#) [선택된 파일 없음](#)

전송하기

실습 예제 설명



- client
 - Index.js : 진입점
 - LayerManager.js : 레이어 관리
 - MovieManager.js : 영화 후기 관리
 - service.js : 서버 API 호출
 - view.js : 화면에 표현되는 HTML 구성
- server
 - api.js : API router, databaseBridge 를 사용하여 데이터 관리
 - databaseBridge.js : DB 처리
- db.sqlite : sqlite 데이터 파일
- Index.html : 진입 페이지



설계하기 (실습)

데이터 베이스 작업

DB Browser for SQLite로 생성하기

- DB 생성 (db.sqlite)
- Table 설계하기 (MOVIE 테이블)
 - 등록시간을 존재해야한다.
 - 모든 영화후기 정보는 고유한 no를 가져야한다. No는 (입력시 자동증가된다)

영화관람 후기

등록



완벽한 타인

★★★★★

등록일: 2018-10-28 15:27:42

[관련 링크](#)

[수정](#) [삭제](#)



벽 속에 숨은 마법시계

★★★★★

등록일: 2018-10-28 15:09:21

[수정](#) [삭제](#)

영화관람 후기

등록

제목

평점

0

후기

참고URL

이미지

[파일 선택](#) [선택된 파일 없음](#)

전송하기

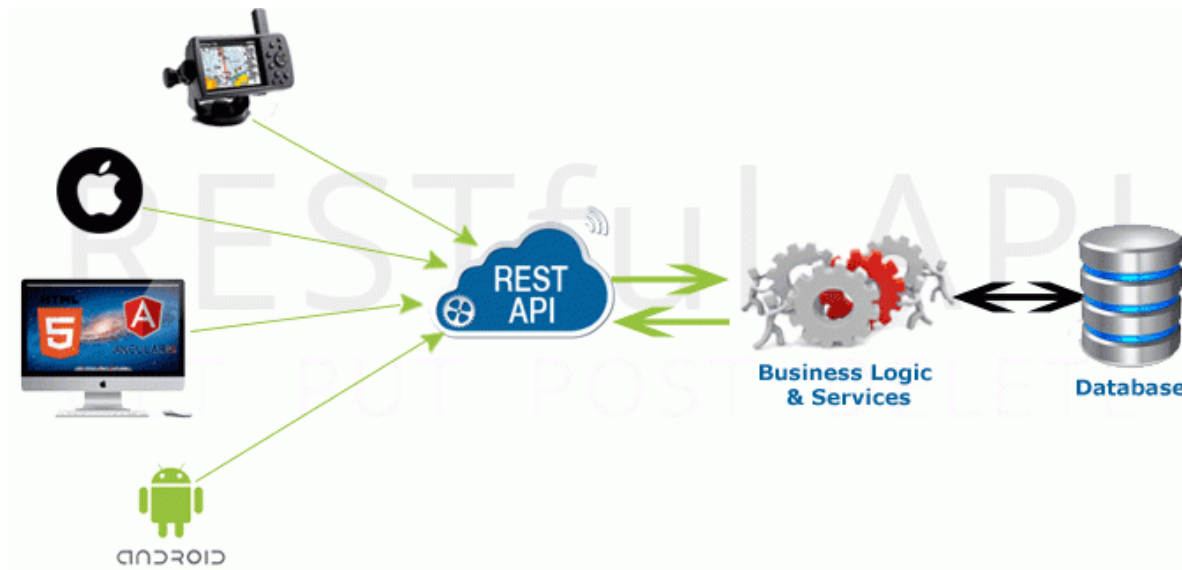
서버 사이드 작업

RESTful API

REST는 **R**epresentational **S**tate **T**ransfer 의 약어.

웹의 장점을 최대한 활용할 수 있는 아키텍처

REST API를 읽는 것 만으로도 메시지가 의도하는 바를 명확



RESTful API

URI - 자원(RESOURCE)

/members/1

/houses/apartments

HTTP Method - 행위(Verb)

- **POST:** 리소스를 생성
- **PUT:** 리소스를 수정
- **GET:** 리소스를 조회
- **DELETE:** 리소스를 제거

GET	/movies	Get list of movies
GET	/movies/:id	Find a movie by its ID
POST	/movies	Create a new movie
PUT	/movies	Update an existing movie
DELETE	/movies	Delete an existing movie

RESTFul API 설계 주의 사항

1. URI에 행위를 포함하지 않는다.
(리소스명은 동사 보다는 명사)

GET /members/**add** (X)

POST /members (O)

2. URI의 슬래시(/)는 계층 관계를 표현한다.

GET /houses/apartments

GET /animals/mammals/whales

서버 사이드 작업을 위한 툴

Postman

<https://www.getpostman.com/>

API 개발을 빠르고 쉽게, 개발된 API를 테스트할 수 있고, 팀원들간 공유를 할 수 있게 해주는 플랫폼



Chrome Extensions : [Postman Interceptor](#)

크롬 브라우저의 Request를 Postman 에서 바로 확인할 수 있다.

도움되는 크롬 익스텐션 툴

JSON View

브라우저에서 JSON 잘 보이도록 하는 extention

<https://chrome.google.com/webstore/detail/jsonview/chklaanhfefbnpoihckbnefhakgolnmc>

HTTP Archive Viewer

<https://chrome.google.com/webstore/detail/http-archive-viewer/ebdbdmhegaooopfnjikefdpeoaidml>

기타

Css peeper

<https://chrome.google.com/webstore/detail/css-peeper/mbnbehikldihnfefhnaidhihoofhpehk>

Wappalyzer

<https://chrome.google.com/webstore/detail/wappalyzer/gppongmhjkpfnbhagpmjfkannfbllamg>

The QR Code Extension

<https://chrome.google.com/webstore/detail/the-qr-code-extension/ojdcdmnjggnhgljmhkjlablejfeeb>

Github 관련

Octotree, GitHub Hovercard, Gitmoji

서버 사이드 작업 (실습)

API 작업 (server/api.js, databaseBridge.js)

- 영화후기 작성 API

POST /api/movies → DatabaseBridge의 **insertItem** 개발

- 성공일 경우 { no: 저장된 후기 id } 데이터 형태 반환.
- 실패일 경우 {} 데이터 형태 반환

- 영화후기 목록 조회 API

GET /api/movies → DatabaseBridge 의 **Items** 개발

- 성공일 경우 { { 영화정보 객체 } } 데이터 형태 반환.
- 실패일 경우 {} 데이터 형태 반환

서버 사이드 작업 (실습)

API 작업 (server/api.js, databaseBridge.js)

- 영화후기 내용 조회 API

GET /api/movies/:no → DatabaseBridge 의 **Item** 개발

- 성공일 경우 { no: 후기 id } 데이터 형태 반환.
- 실패일 경우 {} 데이터 형태 반환

- 영화후기 삭제 API

DELETE /api/movies/:no → DatabaseBridge 의 **deleteItems** 개발

- 성공일 경우 { no: 삭제된 후기 id } 데이터 형태 반환.
- 실패일 경우 {} 데이터 형태 반환

프론트 엔드 작업

영화 후기 작성하기

Client/service.js

- 입력 폼에 대해 각각 유효성 검사(validate)를 추가한다.
- 유효하지 않는 폼의 경우 다음과 같은 메시지를 출력한다.
 - 영화 제목 (필수) → 영화 제목을 입력해 주세요.
 - 영화 평점 (필수, 숫자) → 1~5의 평점을 입력해 주세요.
 - 영화 후기 (필수) → 영화 후기를 입력해 주세요.
 - 영화 사이트 링크 (선택)
 - 이미지 (선택, 로컬 파일 선택)
- 서버에 값을 저장하는 insert 함수를 작성한다.

Client/MovieManager.js

- 전송하기 버튼을 클릭 시(attachHandlers)에 입력 폼의 내용이 서버로 전송된다.
- 전송 후 입력 폼값을 초기화한다. (this.resetForm 함수 사용)

영화관람 후기

등록	
제목	<input type="text"/>
평점	<input type="text" value="0"/>
후기	<div><div></div></div>
참고URL	<input type="text"/>
이미지	<input type="button" value="파일 선택"/> 선택된 파일 없음
전송하기	

프론트 엔드 작업

File API 적용하기(client/MovieManager.js)

attachHandlers 함수에서

- 입력폼의 파일첨부 버튼을 선택 시 이벤트 핸들러 를 등록한다.

checkFiles 함수에서

- File API를 사용해 이미지 타입만 첨부 할 수 있도록 한다.
- 첨부된 이미지를 FileReader API로 읽어 들여 DataURI로 값을 설정한다.
- DataURI로 얻은 데이터를 setImage 함수의 파라미터로 넘긴다.
- 이미 구현된 setImage 함수는 form의 image 필드에 값을 설정한다.

프론트 엔드 작업

CANVAS 적용하기

client/view.js의 drawCanvas

- 입력 폼에 첨부된 이미지 데이터(DataURI)를 이용하여 target(canvas)의 width와 height 크기에 맞는 이미지를 노출한다

이미지

파일 선택 halloween.jpg



전송하기

프론트 엔드 작업

영화 후기 목록 화면

작성한 영화 후기 목록을 표시한다.

Client/service.js

- 서버에 값을 저장하는 list 함수를 작성한다.

Client/MovieManager.js

refreshList 함수 작성

- Service.js의 list 함수를 호출하여 목록 데이터를 얻고, 얻은 데이터를
- Client/view.js의 drawMovie 함수를 이용하여 화면에 표현한다.

updateList 함수는

1. 페이지가 처음 로딩하였을때
2. 등록이 된 이후에 호출하도록 수정한다.



완벽한 타인



등록일: 2018-10-28 15:27:42

[관련 링크](#)



벽 속에 숨은 마법시계



등록일: 2018-10-28 15:09:21



퍼스트맨



등록일: 2018-10-28 14:49:14

[관련 링크](#)

프론트 엔드 작업

영화 후기 상세 화면

영화 후기 목록에서 특정 영화후기를 선택 시 후기 내용이 레이어로 표시된다.

attachHandlers 함수에서

- 하단의 글 제목을 클릭하면 글 품 영역 위로 새로운 레이어가 노출된다.

Client/service.js

- 서버에 값을 저장하는 get 함수를 작성한다.

Client/MovieManager.js

onClickDetail 함수 수정

- Service.js의 get 함수를 호출하여 영화 후기 데이터를 얻고, 얻은 데이터를
- Client/view.js의 drawMovieDetail 함수를 이용하여 화면에 표현한다.
- 레이어 상단엔 글 제목과 작성 날짜, 등 상세 정보가 노출되고, 하단영역엔 글 내용이 이미지와 같이 출력된다.
- LayerManager의 instance를 이용하여 this.elements["detail"] 를 layer로 띄운다.

프론트 엔드 작업

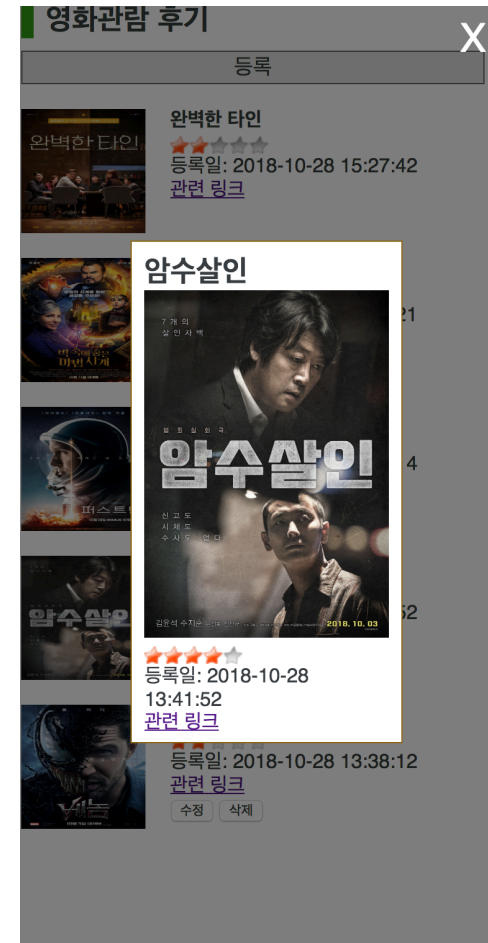
영화 후기 상세 화면

레이어는 client/LayerManager.js를 이용한다

- LayerManager의 마크업 구조

```
<section class="layer">
  <div class="layer-content"></div>
  <div class="close">X</div>
</section>
```

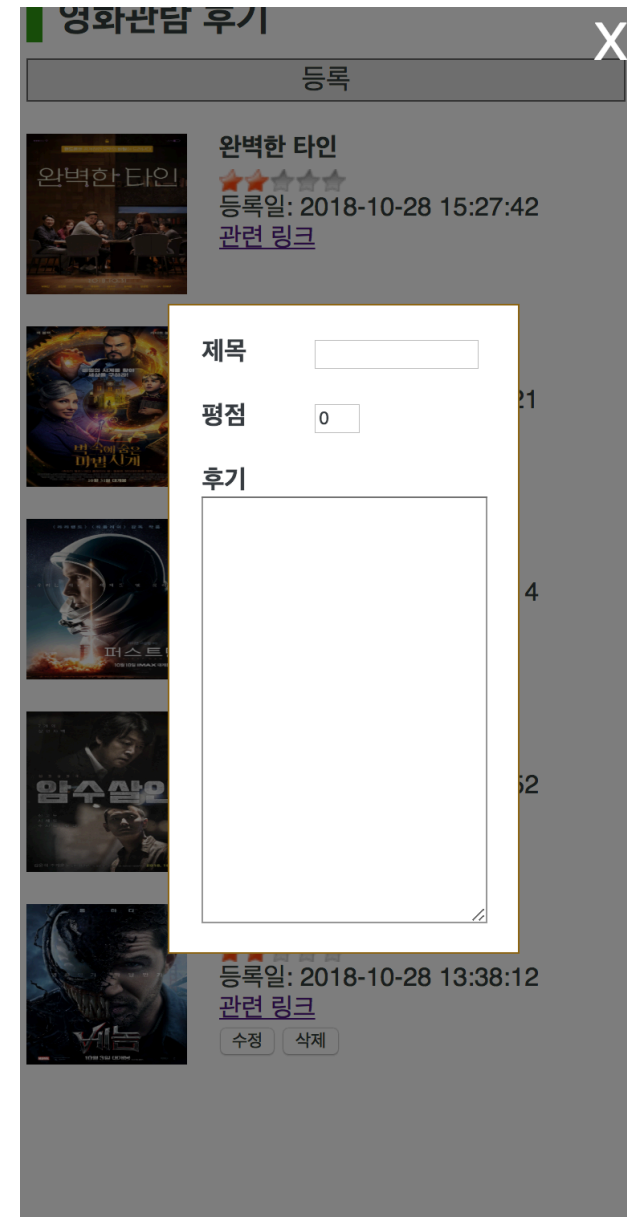
- LayerManager.show(layer의 상단 .layer인 엘리먼트);
- LayerManager.hide(layer의 상단 .layer인 엘리먼트);



추가 작업 #1

UI 개선

- 입력 폼이 목록 상단에 위치해 사용에 불편을 주고 있으므로 이를 '영화 후기 상세'와 같은 별도의 버튼을 사용해 입력 폼이 상세 보기와 같이 레이어로 노출되도록 변경한다.
- Form영역 layer 마크업으로 구성
- readyForm 함수에서 this.elements["form"] 를 LayerManager를 이용하여 보여주기



추가 작업 #2

UI 개선

- 파일 첨부를 드래그&드롭 가능하도록 드롭 영역을 추가한다.

이미지

파일 선택 선택된 파일 없음

이곳에 파일을 드롭하세요.

attachHandlers

- `this.elements["drop"]` 엘리먼트 영역에 `drop`, `dragover` 이벤트를 붙인다.
- `Drop` 이벤트에서 `checkFiles`를 사용하여 파일 첨부했을때와 동일한 동작이 되도록 구성한다.

추가 작업 #3

UI 개선

영화 등록, 영화 추가를 Web Animation API를 이용하여 레이어 노출작업을 진행한다.

Client/LayerManager.js

layer에 투명도(opacity)를 0~1로 0.5초 동안 변경되도록 Web Animation API를 사용하여 구성하시오.

추가 작업 #4

영화 후기 삭제 API 개발

[서버 사이드]

앞에서 설계한 API를 이용하여 리스트에서 삭제 버튼을 누를 경우 확인창과 함께 데이터를 삭제하는 기능을 추가해보자.



완벽한 타인



등록일: 2018

[관련 링크](#)

삭제

글을 삭제 하시겠습니까?

취소

확인

추가 작업 #5

영화 후기 수정 API 개발

[서버 사이드]

URI를 설계하고 DatabaseBridge의 수정 메소드를 정의한다.

Express에 설계한 URI를 등록한다.

API 작업 (server/api.js, databaseBridge.js)

- 영화후기 수정 API
PUT /api/movies/:no → DatabaseBridge 의 **updateItems** 개발
 - 성공일 경우 { no: 수정된 후기 id } 데이터 형태 반환.
 - 실패일 경우 {} 데이터 형태 반환

추가 작업 #5

영화 후기 수정 API 개발

[프론트 엔드 작업]

목록에 수정 버튼을 노출한다

·
수정 버튼을 누르면 등록화면을 수정화면으로 수정한다..



퍼스트맨



등록일: 2018-10-28 14:49:14

[관련 링크](#)

수정

삭제

제목

퍼스트맨

평점

1

후기

"여러분은 지금까지 보지 못한 걸 보게 될 겁니다"
이제껏 누구도 경험하지 못한 세계에 도전한 우주비행사 **닐**(라이언
고슬링)은,
거대한 위험 속에서 극한의 위기를 체험하게 된다.
전세계가 바라보는 가운데, 그는 새로운 세상을 열 첫 발걸음을 내
딛는데...

이제, 세계는 달라질 것이다.

추가 작업 #6

영화 후기 등록/수정시

제목과 후기 내용에 따옴표가 표시될 수 있도록 등록/ 수정될 수 있도록 수정한다.

Server/api.js의 encodeEntities, decodeEntities 를 이용하여
Server/api.js의 등록과 수정 API를 수정한다.

고맙습니다.
