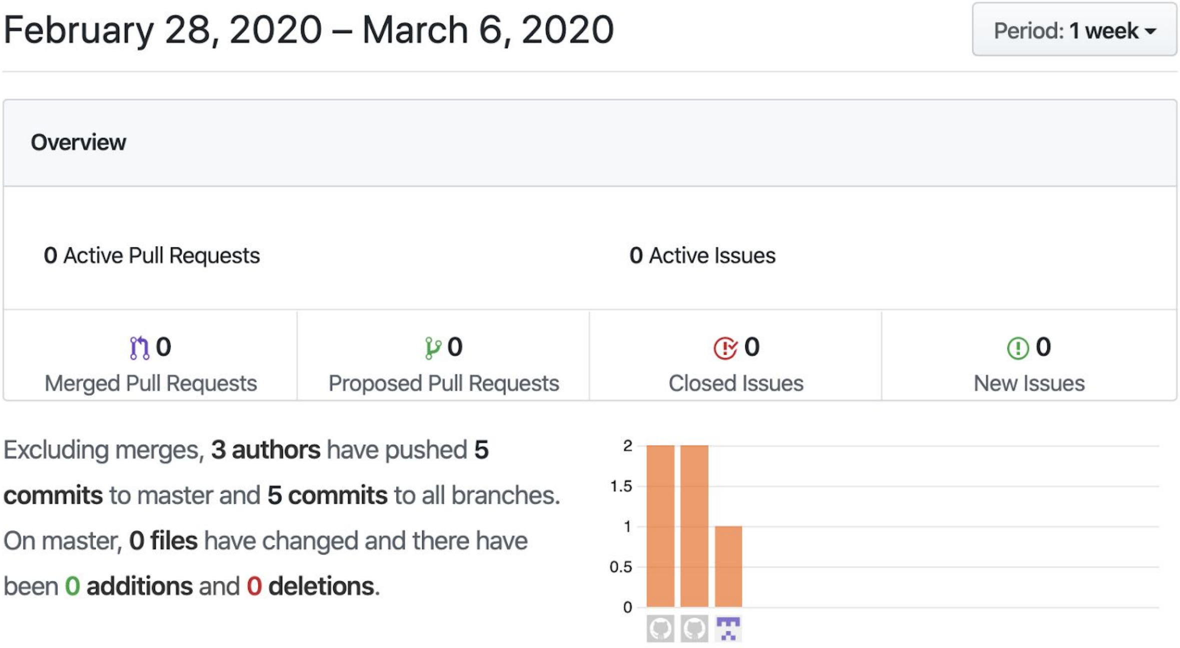
Technical Report for Phase One

Website link

https://master.d3tqvxnaovxzov.amplifyapp.com/showtime

Github link

https://github.com/Lena-Liwen-Tian/PumpkinMovie1



February 28, 2020 – March 6, 2020

Period: 1 week

Overview

0 Active Pull Requests

0 Active Issues

0 Merged Pull Requests

0 Proposed Pull Requests

0 Closed Issues

0 New Issues

Excluding merges, **3 authors** have pushed **5 commits** to master and **5 commits** to all branches. On master, **0 files** have changed and there have been **0 additions** and **0 deletions**.

Team Information

Name:Liwen Tian
Email:liwentian1103@gmail.com
Github username: Lena-Liwen-Tian
Canvas group: group 8
Project name: Pumpkin Movie

Project Description

My project name is Pumpkin Movie, it provides users a better way to buy their ideal movie tickets online. On my website, users can find movie information in details, eg. movie time, movie genres, movie ratings. Users can also get the relevant information about all theatres in the United States, eg.location, phone numbers. Besides, the show time of different movies at different theatres are also listed on the website for users to choose from. Pumpkin Movie allows users to combine information about movies, theatres and show time ahead of time, so that they can buy an ideal movie ticket in the end. The target users are young people who are looking for an easier way to buy an ideal ticket in order to get a better viewing experience in the theatre.

User Stories

In Phase 1, my customer team gave me 5 user stories as follows.
1. As a user, I want to know the phone number of the theatre so that I can contact for help.

I estimated that I would spend about 1hour doing it. Actually, it just cost me half an hour since I used some templates of other pages. The assumption for this user story is that phone number is the most important way for users to contact others and ask for help.

2. As a user, I want to know the address of the theatre so that I can map it.

I estimated that I would spend about 1 hour doing it, but as I mentioned above, I only used half an hour for the theatre category. The assumption for this is that users are not familiar with these theatres so that they need to map it.

3. As a user, I want to know the genre of the movie to make sure I will like it.

I estimated that I would spend 20 minutes completing this, but I spent about 30 minutes on it because I am not familiar with mapping an array in React. The assumption for this is that the users use the genres to figure out if the movie matches their tastes.

4. As a user, I want to know the show time of the movie, so I can be there on time.

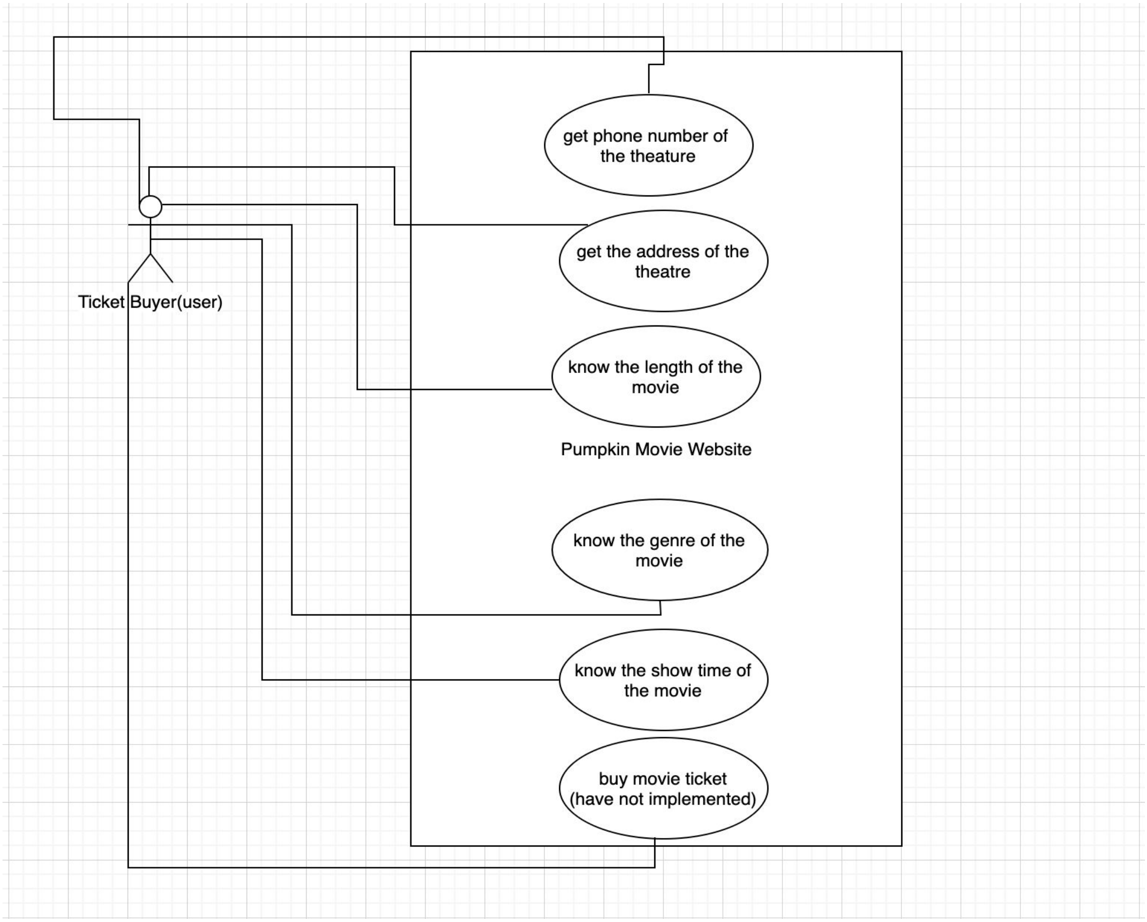
I estimated that I would spend about 1 hour on this, but actually I spent 40 minutes on this. I did not list a lot of instances for the show time category. The assumption is that users do not want to miss any plots in the movie so that they want to be on time.

5. As a user, I want the length of the movie to plan when it lets out.

I estimated that I would spend about 15 minutes for this. I think I really spend about 15 minutes. The assumption is that users are not willing to stay at the theatre after the movie finishes.

Use Case Diagram

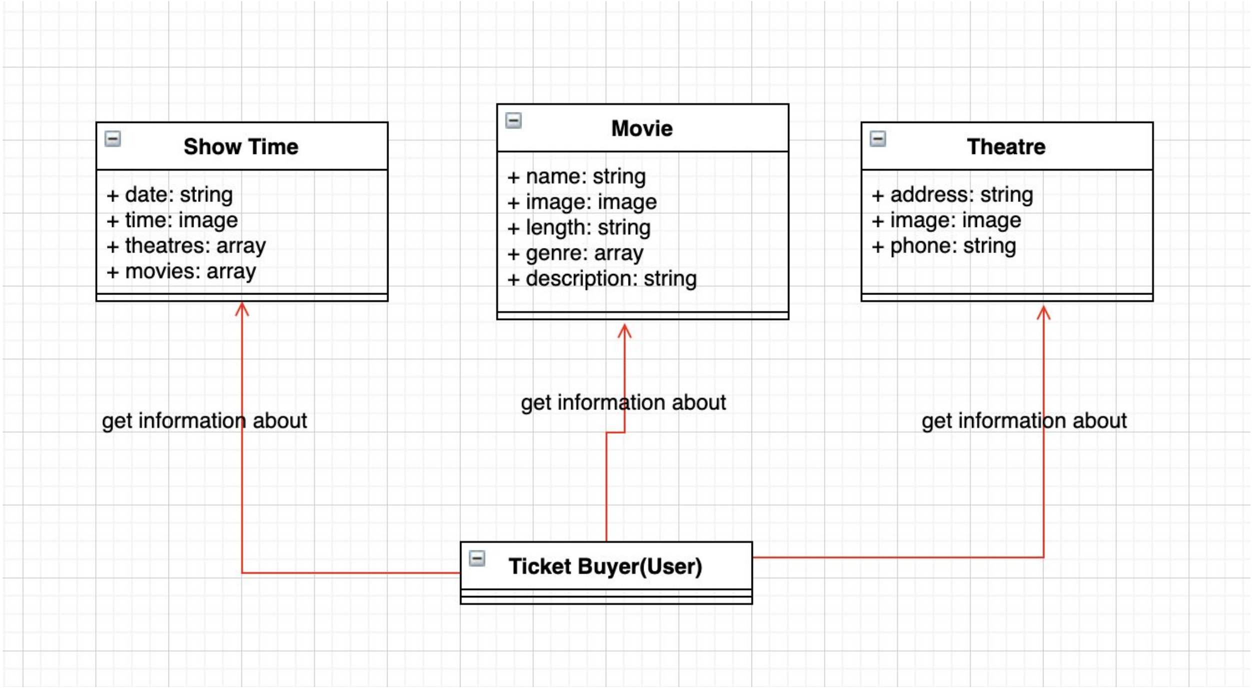
My use case diagram is shown below. Based on the user story above, the User(can also be called TicketBuyer) is an actor, he/she has associations with the Pumpkin Movie Website. He/she has about 5 use cases similar to user stories in Phase1. I added “buy movie tickets” since I included the button of buying a movie ticket in my website though I have not made them dynamically yet.


get phone number of the theature
get the address of the theatre
know the length of the movie
Pumpkin Movie Website
know the genre of the movie
know the show time of the movie
buy movie ticket (have not implemented)
Ticket Buyer(user)

Design and UML Diagrams

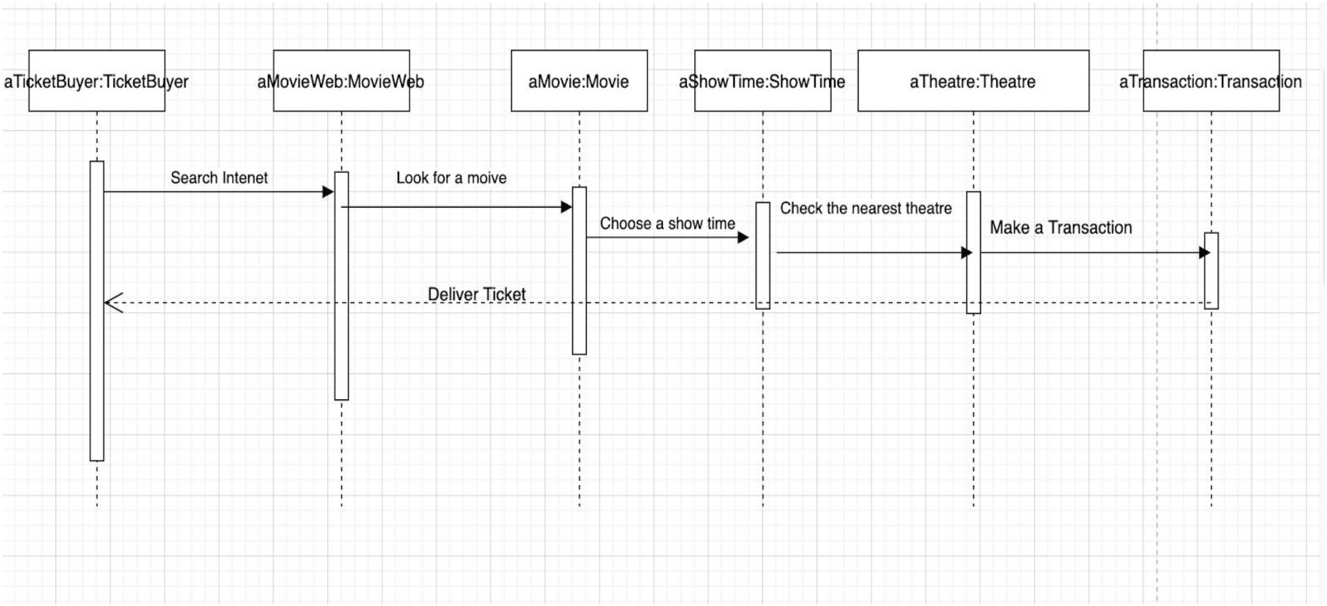
The general idea of Pumpkin Movie is to let users get the information about movies, theatres and show time ahead of time, then they can buy their ideal tickets in the end. The first UML diagram shows three models I created in Phase 1.

The Movies model has 5 fields: name, image, length, genre and description. Genre is an array because one movie may have two genres. The Show Time model has 4 fields: date, time, theatres and movies. Both theatres and movies are arrays since one show time can have more than one theatres and more than one movies. The Theatre model has three fields: address, image and phone. The Ticket Buyer(User) shown at the bottom can get the information from all three models above.



Since we only rendered static pages in Phase1, I have not included a lot of sequential actions in the website now. But I have some general thoughts about the whole ticket purchase process that I will probably implement later.

In order to buy an ideal movie ticket, the first step for users is to search the Internet and open our website. Then they can click the movies button and look through the movies page to see which movie they want to watch. After picking up the movie they want, they click “buy a ticket” at the bottom of the movie page. This button can direct them to the show time page which shows users the time and locations for a certain movie. Suppose the use is not familiar with the location of different theatres, they need to check the location of the different theatres after picking up the right time. After all things are done, the user has chosen the time, location and movie for their ticket, they will be redirected to the transaction page, complete the ticket purchase online. In the end, the ticket will be delivered to the user online. Below is the Sequential Diagram. I have not realized it in this phase, but it shows the transaction process I will complete later.

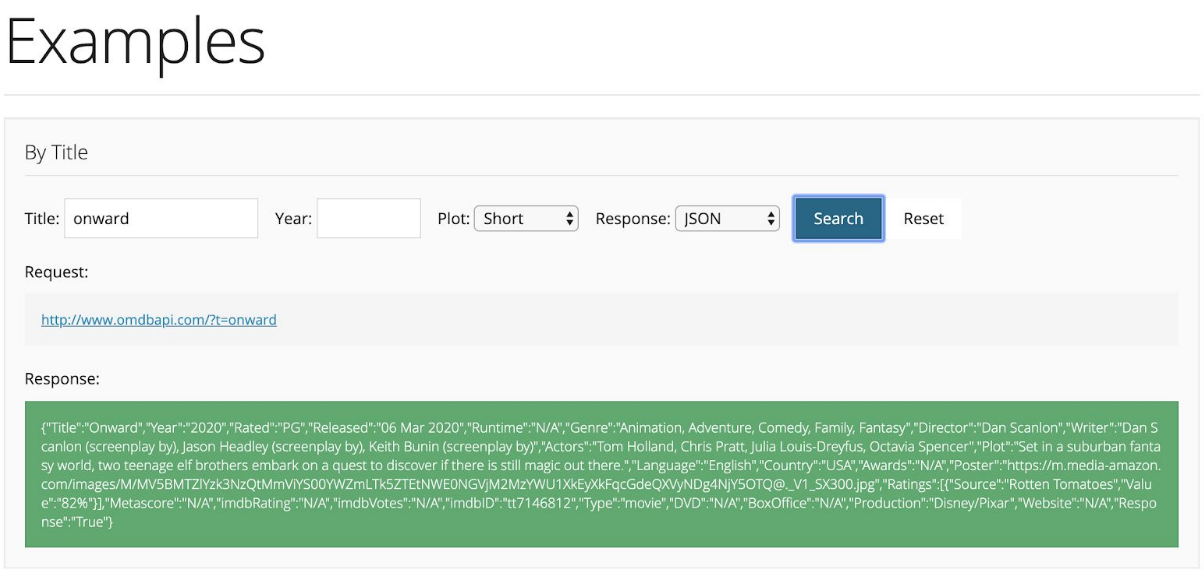


Models

I do not have any testing in phase1. I have three models from three sources. The first model is movies, it includes a movie image, movie name, movie length, movie genre and movie descriptions. My source for this is IMDB database which has lots of popular movies and relevant information. The second model is theatres. This model has theatre names, locations and phone numbers in it. I got this from MovieGlue, an online movie database gathering the information about all movie theatres in the United States. The third model is Show Time. It includes show dates and time for different movies. Different show time also matches theatre names and movie names which are related to the other databases. These come from another online database called GraceNote Developer.

I attached a screenshot here which proves that all these databases can be scraped through their APIs. The screenshot also shows the format of these data.

IMDB -- get the information about movies



Examples

By Title

Title: onward Year: Plot: Short Response: JSON

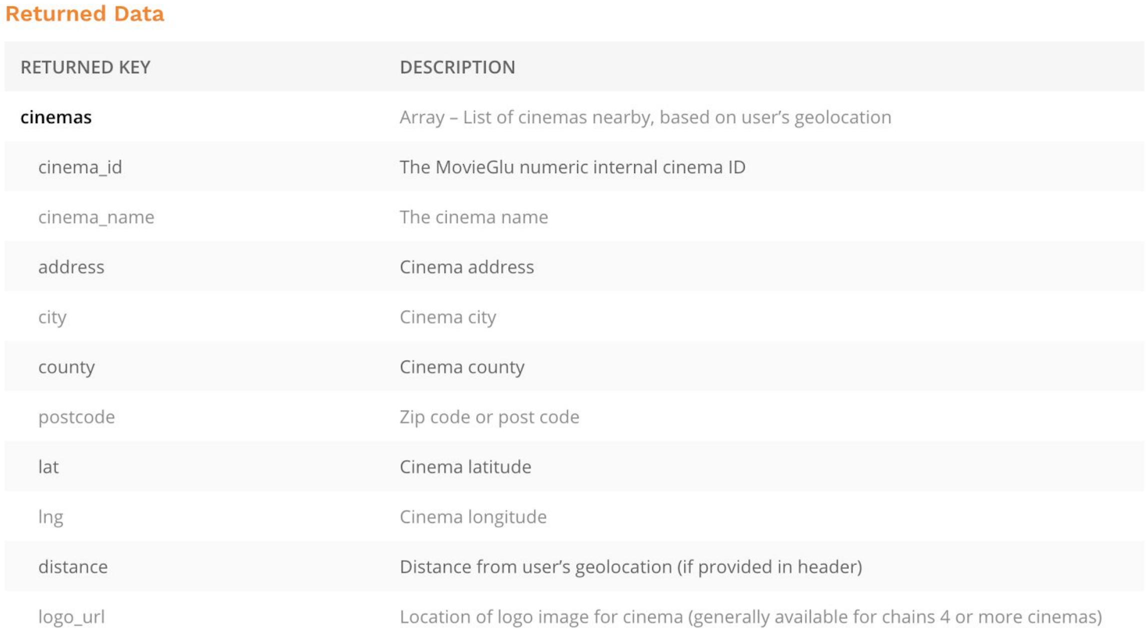
Request:

http://www.omdbapi.com/?t=onward

Response:

```
{"Title":"Onward","Year":"2020","Rated":"PG","Released":"06 Mar 2020","Runtime":"N/A","Genre":"Animation, Adventure, Comedy, Family, Fantasy","Director":"Dan S
canlon","Writer":"Dan S
canlon (screenplay by), Jason Headley (screenplay by), Keith Bunin (screenplay by)","Actors":"Tom Holland, Chris Pratt, Julia Louis-Dreyfus, Octavia Spencer","Plot":"Set in a suburban fanta
sy world, two teenage elf brothers embark on a quest to discover if there is still magic out there.","Language":"English","Country":"USA","Awards":"N/A","Poster":"https://m.media-amazon.
com/images/M/MV5BMTZlYzk3NzQtMmViYS00YWZmLTk5ZTEtNWE0NGVjM2MzYWU1XkEyXkFqcGdeQXVyNDg4NjY5OTQ@._V1_SX300.jpg","Ratings":[{"Source":"Rotten Tomatoes","Valu
e":"82%"}],"Metascore":"N/A","imdbRating":"N/A","imdbVotes":"N/A","imdbID":"tt7146812","Type":"movie","DVD":"N/A","BoxOffice":"N/A","Production":"Disney/Pixar","Website":"N/A","Respo
nse":"True"}
```

MovieGlueDatabase -- get the information about cinema

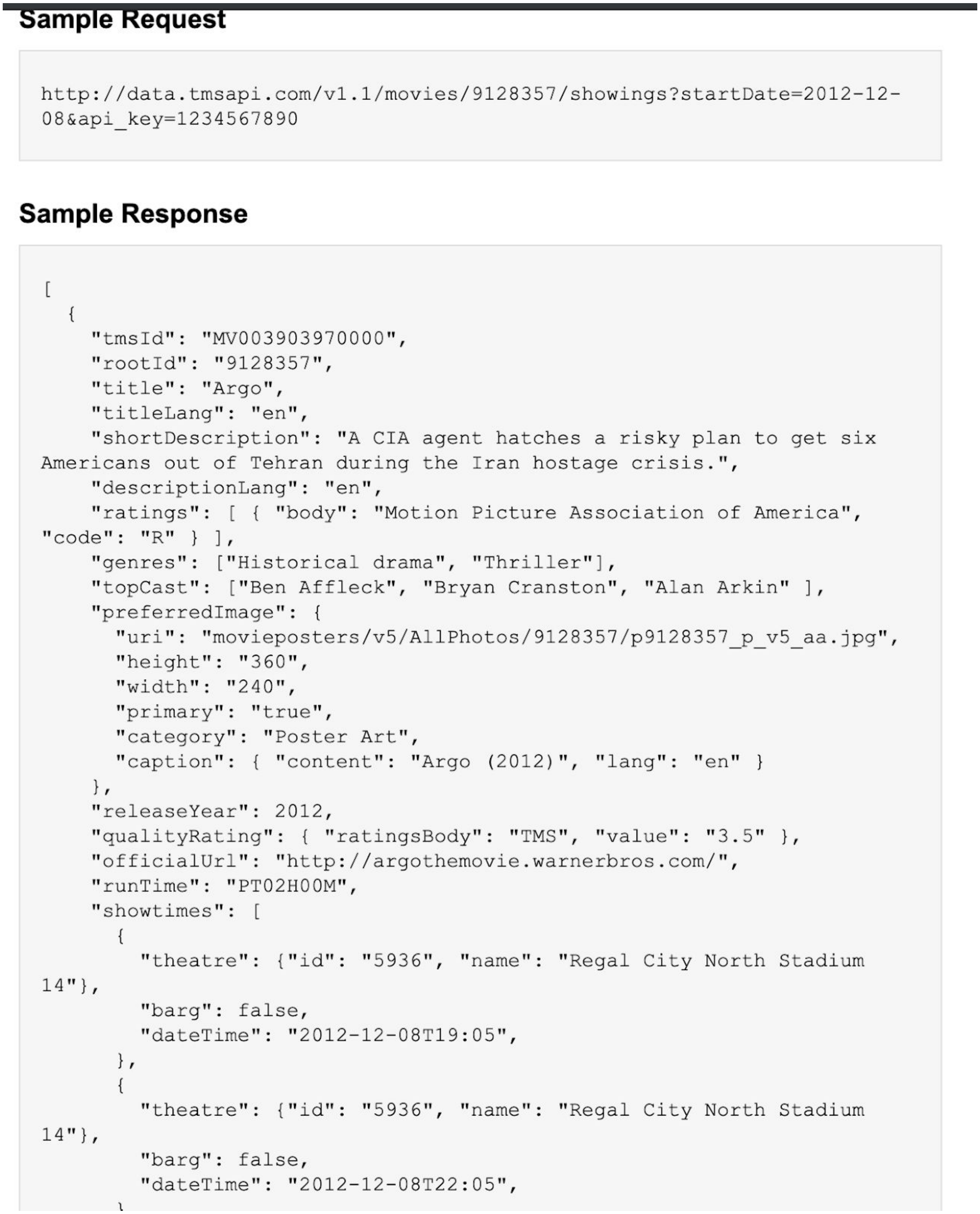


Returned Data

RETURNED KEY	DESCRIPTION
cinemas	Array – List of cinemas nearby, based on user's geolocation
cinema_id	The MovieGlu numeric internal cinema ID
cinema_name	The cinema name
address	Cinema address
city	Cinema city
county	Cinema county
postcode	Zip code or post code
lat	Cinema latitude
lng	Cinema longitude
distance	Distance from user's geolocation (if provided in header)
logo_url	Location of logo image for cinema (generally available for chains 4 or more cinemas)

```
  {
    "cinema_id": 8893,
    "cinema_name": "Cineworld Leicester Square",
    "address": "Leicester Square",
    "city": "London",
    "county": "Greater London",
    "postcode": "WC2H 7NA",
    "lat": 51.51078,
    "lng": -0.1305,
    "distance": 0.031255415281335,
    "logo_url": "https://d2z9fe5yu2p0av.cloudfront.net/chainlogos/uk/UK-868-sq.jpg"
  },
  {
    "cinema_id": 8923,
    "cinema_name": "Vue Cinemas - West End",
    "address": "3 Cranbourn Street",
    "city": "London",
    "county": "Greater London",
    "postcode": "WC2H 7AL",
    "lat": 51.511181,
    "lng": -0.12934,
    "distance": 0.064288623403495,
    "logo_url": "https://d2z9fe5yu2p0av.cloudfront.net/chainlogos/uk/UK-712-sq.jpg"
```

GraceNote Developer -- get the information about show time



Sample Request

```
http://data.tmsapi.com/v1.1/movies/9128357/showings?startDate=2012-12-
08&api_key=1234567890
```

Sample Response

```
[
  {
    "tmsId": "MV003903970000",
    "rootId": "9128357",
    "title": "Argo",
    "titleLang": "en",
    "shortDescription": "A CIA agent hatches a risky plan to get six
Americans out of Tehran during the Iran hostage crisis.",
    "descriptionLang": "en",
    "ratings": [ { "body": "Motion Picture Association of America",
"code": "R" } ],
    "genres": ["Historical drama", "Thriller"],
    "topCast": ["Ben Affleck", "Bryan Cranston", "Alan Arkin" ],
    "preferredImage": {
      "uri": "movieposters/v5/AllPhotos/9128357/p9128357_p_v5_aa.jpg",
      "height": "360",
      "width": "240",
      "primary": "true",
      "category": "Poster Art",
      "caption": { "content": "Argo (2012)", "lang": "en" }
    },
    "releaseYear": 2012,
    "qualityRating": { "ratingsBody": "TMS", "value": "3.5" },
    "officialUrl": "http://argothemovie.warnerbros.com/",
    "runTime": "PT02H00M",
    "showtimes": [
      {
        "theatre": {"id": "5936", "name": "Regal City North Stadium
14"},
        "barg": false,
        "dateTime": "2012-12-08T19:05",
      },
      {
        "theatre": {"id": "5936", "name": "Regal City North Stadium
14"},
        "barg": false,
        "dateTime": "2012-12-08T22:05",
      }
```

Tools

I use Reactjs to build the front end in phase1. Since we are only required to show 14 static web pages, I have not built the back end at this time. I also included BootStrap in my scripts and incorporated some of their Reactjs templates to make my website look better. I also used Python to connect to the three online databases and scraped some of their data. The software I used in this phase is Visual Studios. This is my first time using it to create a Project. It is not very hard. I used AWS Amplify to deploy my website. I tried other AWS services like S3 but failed. I found Amplify is good for deploying static pages.

Reflections

Though I have some experiences of developing websites, I still feel a little stressful for phase 1. The main reason is because I want to try a different Framework that I have not used before. I used Python Flask before, but I changed to MERN for this project. I feel a little proud that I still complete it on time by myself though I am still learning how to do it. I stayed up late for a few days, now I feel I am so lucky to get all things done by myself. However, there are still a lot of things I can improve. I should try to add some new features to my website. I am thinking about connecting to GoogleMap API to show the location of different theatres on the map. I also tried to do user authentication but failed. I think I can complete user authentication in the next phase. I learned a lot of things about React in this phase, I hope I could keep learning later.