MoviePick User Guide

(Team 5)

This service is provided for the objectives that are about the relations between movies that are currently in shows in theaters. Specifically, the initial data that has been provided for the service is for the movies that are currently being shown in the theaters of Athens.

A simple client application has been written to populate the service with a few movies and theaters.

The following resources and functions on the resources are available in this service:

Resource/Action	GET	PUT	POST	DELETE
/movie	A	N/A	A	N/A
/theater	A	N/A	A	N/A
/movie/{id}	A	A	N/A	A
/theater/{id}	A	A	N/A	A
/movie/rate/{rate}	A	N/A	N/A	N/A
/movie/genre/{genre}	A	N/A	N/A	N/A
/movie/{id}/theater	A	N/A	A	N/A
/theater/{id}/movie	A	N/A	A	N/A
/movie/{mid}/theater/{tid}	A	N/A	N/A	A
/theater/{tid}/movie/{mid}	A	N/A	N/A	A

JSON objects for resource-action pairs and the expected results are as follows:

1. /movie-GET
    a. Returns all available movies
    b. No JSON object is required as an input
2. /movie-POST
    a. Creates a new movie and adds it to the list of movies in the service
    b. Input JSON object:
    `{\"movieTitle\":\"Black or White\",\"genre\":\"Drama\",\"rate\":3}`
3. /theater-GET
    a. Returns all existing theaters
    b. No JSON object is required as an input
4. /theater-POST
    a. Creates a new theater and adds it to the list of theaters in the service

   b. Input JSON object:
      `{\"theaterName\":\"University 16\",\"theaterAddress\":\"Oconee Connector\"}`

5. /movie/{id}-GET
   a. Returns available description of a movie with the given id
   b. {id} is an integer identifier of a movie
   c. No JSON object is re required as an input
6. /movie/{id}-PUT
   a. Updates the description of a movie with the given id, such as genre and rate
   b. {id} is an integer identifier of a movie
   c. Input JSON object:
      `{\"movieTitle\":\"Black or White\",\"genre\":\"Drama\",\"rate\":2}`
7. /movie/{id}-DELETE
   a. Deletes a movie with the given id from the list of movies in the service
   b. {id} is an integer identifier of a movie
   c. No JSON object is required as an input
   d. If a movie gets deleted, it will be removed from all theaters, it means that the showtimes of that movie in all theaters will be removed
8. /theater/{id}-GET
   a. Returns available description of a theater with the given id
   b. {id} is an integer identifier of a theater
   c. No JSON object is re required as an input
9. /theater/{id}-PUT
   a. Updates the description of a theater with the given id, such as name and address
   b. {id} is an integer identifier of a theater
   c. Input JSON object:
      `{\"theaterName\":\"University 16\",\"theaterAddress\":\"Oconee Connector\"}`
10. /theater/{id}-DELETE
    a. Deletes a theater with the given id from the list of theaters in the service
    b. {id} is an integer identifier of a theater
    c. No JSON object is required as an input
    d. If a theater get deleted, all movies that were being shown in that theater will be removed from the list of showtimes
11. /movie/rate/{rate}-GET
    a. Returns all movies with the description by specific rate
    b. {rate} is an integer between 1 and 5 (inclusive)
    c. No JSON object is required as an input
12. /movie/genre/{genre}-GET
    a. Returns all movies with the description by specific genre
    b. {genre} is a name of genre (String)
    c. No JSON object is required as an input
13. /movie/{id}/theater-GET
    a. Returns all theaters that are showing a movie with a given id and the available showtimes of that movie in each theater

    b. {id} is an integer identifier of a movie
    c. No JSON object is required as an input

14. /movie/{id}/theater-POST
    a. Registers a pre-defined theater with the given id (through POST arguments) to show a movie with a defined id (in URI) with some specific showtimes
    b. {id} is an integer identifier of a movie
    c. Input JSON object:
    ```
    {\"integer\":1,\"stringArray\":[\"Tuesdays 2:30\",\"Thursdays 5:30\"]}
    ```
    First element in the JSON object defines theater id
15. /theater/{id}/movie-GET
    a. Returns all movies that are being shown in a theater with a given id and the available showtimes of that theater for each movie
    b. {id} is an integer identifier of a theater
    c. No JSON object is required as an input
16. /theater/{id}/movie-POST
    a. Adds a pre-defined movie with the given id (through POST arguments) to a theater with the given id (in URI) with its available showtimes
    b. {id} is an integer identifier of a theater
    c. Input JSON object:
    ```
    {\"integer\":1,\"stringArray\":[\"Tuesdays 2:30\",\"Thursdays 5:30\"]}
    ```
    First element in the JSON object defines movie id
17. /movie/{mid}/theater/{tid}-GET
    a. Returns all showtimes of a movie with the given id that is under show in a theater with the defined id plus the information of the movie and the theater
    b. {mid} is an integer identifier of a movie
    c. {tid} is an integer identifier of a theater
    d. No JSON object is required as an input
18. /movie/{mid}/theater/{tid}-DELETE
    a. Removes a theater with the given id from theaters that are showing the given movie which means that theater is not going to show the movie anymore
    b. {mid} is an integer identifier of a movie
    c. {tid} is an integer identifier of a theater
    d. No JSON object is required as an input
19. /theater/{tid}/movie/{mid}-GET
    a. Returns all showtimes of a movie with the given id that is under show in a theater with the defined id plus the information of the movie and the theater
    b. {mid} is an integer identifier of a movie
    c. {tid} is an integer identifier of a theater
    d. No JSON object is required as an input
20. /theater/{tid}movie{mid}-DELETE
    a. Removes a movie with the given id from the movies that the theater shows which means that theater is not going to show the movie anymore
    b. {mid} is an integer identifier of a movie

c. {tid} is an integer identifier of a theater
d. No JSON object is required as an input

The client application does the following sequence of actions on the provided resources and shows the response returned by the service:

- Creating 3 theaters using JSON object (Action: POST _ URI: /theater)
- Retrieving created theaters and showing them in JSON representation (Action: GET _ URI: /theater)
- Creating 4 movies using JSON object (Action: POST _ URI: /movie)
- Updating first movie using JSON object (Action: PUT _ URI: /movie/1)
- Retrieving the updated movie and showing it in JSON representation (Action: GET _ URI: /movie/1)
- Deleting the 4th movie (Action: DELETE _ URI: /movie/4)
- Attempting to retrieve the deleted movie and failing (Action: GET _ URI: /movie/4)
- Creating 3 show times by
    - Adding movie1 to theater1, theater2 and theater3 using JSON object (Action: POST _ URI: /theater/1/movie/), (Action: POST _ URI: /theater/2/movie/), (Action: POST _ URI: /theater/3/movie/)
- Creating 4 show times by
    - Adding theater1 to theaters that are showing movie2 using JSON object (Action: POST _ URI: /movie/2/theater/)
    - Adding theater1 to theaters that are showing movie3 using JSON object (Action: POST _ URI: /movie/3/theater/)
    - Adding theater2 to theaters that are showing movie2 using JSON object (Action: POST _ URI: /movie/2/theater/)
    - Adding theater3 to theaters that are showing movie3 using JSON object (Action: POST _ URI: /movie/3/theater/)
- Retrieving movies and their showtimes of theater1, showing in JSON representation (Action: GET _ URI: /theater/1/movie)
- Retrieving theater1 and showtime(s) of movie1, showing in JSON representation (Action: GET _ URL: /movie/1/theater/1)
- Retrieving movies by a specific genre (Action: GET _ URI: /movie/genre/drama)
- Retrieving movies by a specific rate (Action: GET _ URI: /movie/rate/3)