

URL / Pattern: /

HTTP Method: GET

Description:

Retrieves the home page

URL Parameters:

None

Expected Request Body Data:

None

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status code of 200.
- An HTML page including a welcome and a “slideshow” of trending movies

URL / Pattern: /movies/{mID}

HTTP Method: GET

Description:

Retrieves the movie of mID, if it exists

URL Parameters:

- mID(int) - the unique movie ID to retrieve

Expected Request Body Data:

None

Supported Response Data Types:

- text/html
- application/json

Expected Response (Success):

- Status code of 200.
- If text/html is requested: An HTML page including (but not limited to) title, runtime, date, avgRating, reviews, etc
- If application/json is requested: A movie object with the given id.

URL / Pattern: /movies

HTTP Method: POST

Description:

Creates a new movie

Expected Request Body Data:

JSON string containing the data for a new movie (title, awards, poster link(optional), writers, actors, directors, date, runtime, genres, plot)

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status code of 201
- JSON string containing the new movie object

URL / Pattern: /movie/{mld}

HTTP Method: PUT

Description:

Updates the movie with the given user ID, if the movie exists and the request data is valid. For each property in the JSON string contained in the request that is valid, the movie's data associated with that property will be updated. Properties that do not exist will be ignored.

URL Parameters:

- {mld}(int) - the unique movie to retrieve

Expected Request Body Data:

A JSON string containing 1+ properties that match properties of movie objects on the server. Values associated with properties are expected to contain valid data (server will verify before making changes).

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status code of 200
- JSON string containing the modified movie object

URL / Pattern: /people/{name}

HTTP Method: GET

Description:

Retrieves the person of {name}, if it exists

URL Parameters:

- {name}(string) - the unique name of the person to retrieve

Expected Request Body Data:

None

Supported Response Data Types:

- text/html
- application/json

Expected Response (Success):

- Status code of 200.
- If text/html is requested: An HTML page including (but not limited to) name, work history, etc
- If application/json is requested: A person object with the given name.

URL / Pattern: /people

HTTP Method: POST

Description:

Creates a new person

Expected Request Body Data:

JSON string containing the data for a new person (work history (acting/directing/writing), name)

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status code of 201

- JSON string containing the new person object

URL / Pattern: /people/{name}

HTTP Method: PUT

Description:

Updates the person with the given name, if they exist and the request data is valid. For each property in the JSON string contained in the request that is valid, the person's data associated with that property will be updated. Properties that do not exist will be ignored.

URL Parameters:

- {name}(string) - the unique person to retrieve

Expected Request Body Data:

A JSON string containing 1+ properties that match properties of person objects on the server. Values associated with properties are expected to contain valid data (server will verify before making changes).

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status code of 200
- JSON string containing the modified person object

URL / Pattern: /users/{username}

HTTP Method: GET

Description:

Retrieves the user with {username}, if it exists

URL Parameters:

- username(string) - the unique user with {username} to retrieve

Expected Request Body Data:

None

Supported Response Data Types:

- text/html
- application/json

Expected Response (Success):

- Status code of 200.
- If text/html is requested: An HTML page including (but not limited to) username, reviews made, recommended movies, etc
- If application/json is requested: A user object with the given username.

URL / Pattern: /users

HTTP Method: POST

Description:

Creates a new user

Expected Request Body Data:

JSON string containing the data for a new user (username, password)

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status code of 201
- JSON string containing the new user object

URL / Pattern: /users/{username}

HTTP Method: PUT

Description:

Updates the user with the given username, if they exist and the request data is valid. For each property in the JSON string contained in the request that is valid, the user's data associated with that property will be updated. Properties that do not exist will be ignored.

URL Parameters:

- {username}(string) - the unique user to retrieve

Expected Request Body Data:

A JSON string containing 1+ properties that match properties of user objects on the server. Values associated with properties are expected to contain valid data (server will verify before making changes).

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status code of 200
- JSON string containing the modified user object

URL / Pattern: /reviews/{rID}

HTTP Method: GET

Description:

Retrieves the review of rID, if it exists

URL Parameters:

- rID - the unique review ID to retrieve

Expected Request Body Data:

None

Supported Response Data Types:

- text/html
- application/json

Expected Response (Success):

- Status code of 200.
- If text/html is requested: An HTML page including (but not limited to) date, the movie reviewed, the user author of the review, etc
- If application/json is requested: A review object with the given id.

URL / Pattern: /reviews

HTTP Method: POST

Description:

Creates a new review

Expected Request Body Data:

JSON string containing the data for a new review (date, movie name & id, username, score, title, text)

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status code of 201
- JSON string containing the new review object

URL / Pattern: /review/{rld}

HTTP Method: POST

Description:

Updates the user with the given review ID, if the review exists and the request data is valid. For each property in the JSON string contained in the request that is valid, the review's data associated with that property will be updated. Properties that do not exist will be ignored.

URL Parameters:

- {rld}(int) - the unique review to retrieve

Expected Request Body Data:

A JSON string containing 1+ properties that match properties of review objects on the server. Values associated with properties are expected to contain valid data (server will verify before making changes).

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status code of 200
- JSON string containing the modified review object

URL / Pattern: /login

HTTP Method: GET

Description:

Retrieves the login/register page

URL Parameters:

None

Expected Request Body Data:

None

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status code of 200.
- An HTML page including the username/password inputs and a login / register button

URL / Pattern: /search

HTTP Method: GET

Description:

Retrieves the search page

URL Parameters:

None

Expected Request Body Data:

None

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status code of 200.
- An HTML page including text inputs for searching a movie by title, genre, or actor. Should include a search button as well.

URL / Pattern: /searchresults

HTTP Method: GET

Description:

Retrieves the dynamic search results page

Supported Query Parameters:

- title(string) - filters search by title of the movie (results should be movies that CONTAIN the given title, NOT movies that MATCH the given title)
- genre(string) - filters search by genre of movie (again, contains is the operator, not matches)
- actor(string) - filters search by actor of movie (contains not matches)
- page(int) - not specified by user. Signifies the current page of the results. Max 10 movies / page.

Expected Request Body Data:

None

Supported Response Data Types:

- text/html
- application/json

Expected Response (Success):

- Status code of 200.
- An HTML page containing the search results of the filtered movies. Should not contain more than 10 movies / page. If so, then the user should have a link that navigates to the next (and previous) page
- If application/json is requested: An array of the filtered movie objects.

URL / Pattern: /contribute

HTTP Method: GET

Description:

Retrieves the contribute page

URL Parameters:

None

Expected Request Body Data:

None

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status code of 200.
- An HTML page including links to the addmovie page and addperson page

URL / Pattern: /addmovie

HTTP Method: GET

Description:

Retrieves the addmovie page

URL Parameters:

None

Expected Request Body Data:

None

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status code of 200.
- An HTML page including all the input fields required to create a movie (title, release year, actors, etc) and a submit button.

URL / Pattern: /addperson

HTTP Method: GET

Description:

Retrieves the addperson page

URL Parameters:

None

Expected Request Body Data:

None

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status code of 200.
- An HTML page including all the input fields required to create a person (name, work history, etc) and a submit button.