

Table of Contents

Url Pattern	Pages
/	Pg 2
/movies.....	Pg 2
/movies/{mid}	Pg 3
/newMovie.....	Pg 3
/newMovie.....	Pg 3
/people.....	Pg 4
/people/{name}.....	Pg 5
/newPerson.....	Pg 5
/newPerson.....	Pg 6
/followerPerson.....	Pg 6
/newAccount.....	Pg 6
/newAccount.....	Pg 7
/users.....	Pg 7
/users/{uid}.....	Pg 8
/followerUser.....	Pg 8
/AccountType.....	Pg 9
/WatchedMoviesIn.....	Pg 9
/WatchedMoviesOut.....	Pg 9
/review.....	Pg10
/profile.....	Pg 10
/login.....	Pg 11
/login.....	Pg11
/logout.....	Pg 11

URL/PATTERN: /

HTTP Method: GET

Description:

- Retrieves the home page of the website

Expected Request Body Data:

- None

Support Response Data Types:

- Text/HTML

Expected Response:

- Status code 200
 - An HTML page containing all the components home page
-

URL/Pattern: /movies

HTTP Method: GET

Description:

- Searches movies on the server

Supported Query Parameters:

- title (string) a movie will match if it contains the name value
- actor (string) a movie will match if this actor is in the movie
- director (string) a movie will match if this director directed this movie
- writer (string) a movie will match if this writer contributed to writing the movie
- genre (string) a movie will match if this movie contains that genre keyword
- limit (int) the number of results to return (default 10, minimum 1, maximum 50)
- page (int) the page of the search results

Expected Request Body Data:

- None

Supported Response Data Types

- text/html
- application/JSON

Expected Response(Success)

- Status code of 200
- If HTML is requested
 - An HTML page containing a list of matching movies from the server up to the specific limit. Each entry will be a link to a specific movie on the server
- If application/JSON is requested
 - An array containing a list of matching movie objects will be returned.

Expected Response(Failure)

- Status code of 404
 - If HTML is requested
 - Return error 404 page
 - If application/JSON is requested
 - An empty list is returned
-

Url/Pattern: /movies/{mid}

HTTP Method: GET

Description:

- Searches for a specific movie on the server

URL Parameters

- mid - Unique movie ID to retrieve

Expected Request Body Data:

- None

Supported Response Data Types:

- text/html
- application/json

Expected Response (Success):

- Status Code 200
- If HTML is requested
 - An HTML page containing a title, release year, average review rating, runtime, plot and Movie Poster
- If application/json is requested
 - An object of the movie information

Expected Response (Failure):

- Status code 404
 - If HTML is requested
 - Return error 404 Page
 - If JSON requested
 - Return a NULL object
-

Url/Pattern: /newMovie

HTTP Method: GET

Description:

- Retrieves the movie creating page

Expected Request Body Data:

- None

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status Code 200
- An HTML page containing all the components for creating a new movie on the server

Expected Response (Failure):

- Status Code 404
 - Return error 404 page
-

Url/Pattern: /newMovie

HTTP Method: POST

Description:

- Creates a new movie based on request data

Expected Request Body Data:

- JSON string containing title, release year, runtime, plot, genre keywords, and at least one writer, director, and actor
- Example: {"title":"this title", "year":"2021", "runtime":"1 hr 20 min", "plot":"sharks in a tornado", "genre":["thriller", "action"], "director":["Neil Breen"], "writers":["Neil Breen"], "actors":["Neil Breen"]}

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status Code 201
- A JSON string representing the created movie object
- A string notification to the users that follow a person that is in the movie

Expected Response (Failure):

- Status Code 409
- If JSON requested
 - Return a NULL object

URL/Pattern: /people

HTTP Method: GET

Description:

- Searches for people on the server based on query parameters

Supported Query Parameters:

- name (string) a person will match if it contains the name value
- acted (bool) a person will match if this person has acting experience
- directed (bool) a person will match if this person has directing experience
- wrote (bool) a person will match if this person has writing experience
- movie (string) a person will match if this person has directed, wrote, or acted in this movie
- limit (int) the number of results to return (default 10, minimum 1, maximum 50)
- page (int) the page of the search results

Expected Request Body Data:

- None

Supported Response Data Types

- text/html
- application/JSON

Expected Response(Success)

- Status code of 200
- If HTML is requested
 - An HTML page containing a list of matching people from the server. Each entry will be a link to a specific person on the server
- If application/JSON is requested
 - An array containing a list of matching people objects will be returned.

Expected Response(Failure)

- Status code of 404
 - If HTML is requested
 - Return error 404 page
 - If application/JSON is requested
 - An empty list is returned
-

Url/Pattern: /people/{*name*}

HTTP Method: GET

Description:

- Searches for a specific person on the server

URL Parameters

- name - The name of the person to retrieve

Expected Request Body Data:

- None

Supported Response Data Types:

- text/html
- application/json

Expected Response (Success):

- Status Code 200
- If HTML is requested
 - An HTML page containing an actors name, movies they've acted in, movies they've written, movies they've directed, and frequent collaborators
 - The page will also have the function of following this specific person
- If application/json is requested
 - An object of the director information

Expected Response (Failure):

- If HTML is requested
 - Status code 404
 - Return error 404 Page
 - If JSON requested
 - Return a NULL object
-

Url/Pattern: /newPerson

HTTP Method: GET

Description:

- Retrieves the person creating page

Expected Request Body Data:

- None

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status Code 200
- An HTML page containing all the components for creating a new person on the server

Expected Response (Failure):

- Status Code 404

Return error 404 page

Url/Pattern: /newPerson

HTTP Method: POST

Description:

- Creates a new person based on request data

Expected Request Body Data:

- JSON string containing their name
- Example: {"name": "Neil Breen"}

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status Code 201
- A JSON string representing the created person object

Expected Response (Failure):

- Status Code 409
 - If JSON requested
 - Return a NULL object
-

Url/Pattern: /followerPerson

HTTP Method: PUT

Description:

- Adds the user to the followers of a selected person

Expected Request Body Data:

- none

Supported Response Data Types:

- Text

Expected Response (Success):

- Status Code 204
- A Text saying the user successfully followed a person

Expected Response (Failure):

- Status Code 400
 - Returns "Follower/Followee" does not exist
-

Url/Pattern: /newAccount

HTTP Method: GET

Description:

- Retrieves the account creation page

Expected Request Body Data:

- None

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status Code 200
- An HTML page containing all the components for creating a new account on the server

Expected Response (Failure):

- Status Code 404
 - Return error 404 page
-

Url/Pattern: /newAccount

HTTP Method: POST

Description:

- Creates a new user account based on request data

Expected Request Body Data:

- JSON string containing their username and password
- Example: {"username": "ThePants", "password": "computer4321"}

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status Code 201
- A JSON string representing the created user object

Expected Response (Failure):

- Status Code 403
 - If JSON requested
 - Return a NULL object
-

URL/Pattern: /users

HTTP Method: GET

Description:

- Searches for users on the server based on query parameters

Supported Query Parameters:

- username (string) a user will match if it contains the name value
- limit (int) the number of results to return (default 10, minimum 1, maximum 50)
- page (int) the page of the search results

Expected Request Body Data:

- None

Supported Response Data Types

- text/html
- application/JSON

Expected Response(Success)

- Status code of 200
- If HTML is requested
 - An HTML page containing a list of matching people from the server. Each entry will be a link to a specific user on the server
- If application/JSON is requested
 - An array containing a list of matching user objects will be returned.

Expected Response(Failure)

- Status code of 404
 - If HTML is requested
 - Return error 404 page
 - If application/JSON is requested
 - An empty list is returned
-

Url/Pattern: /users/{uID}

HTTP Method: GET

Description:

- Searches for a specific user on the server

URL Parameters

- uID- The id of the user to retrieve

Expected Request Body Data:

- None

Supported Response Data Types:

- text/html
- application/json

Expected Response (Success):

- Status Code 200
- If HTML is requested
 - An HTML page containing an actors name, movies they've acted in, movies they've written, movies they've directed, and frequent collaborators
 - The page will also have the function of following this specific person
- If application/json is requested
 - An object of the director information

Expected Response (Failure):

- If HTML is requested
 - Status code 404
 - Return error 404 Page
 - If JSON requested
 - Return a NULL object
-

Url/Pattern: /followerUser

HTTP Method: PUT

Description:

- Adds the user to the followers of a selected user

Expected Request Body Data:

- none

Supported Response Data Types:

- Text

Expected Response (Success):

- Status Code 204
- A Text saying the user successfully followed a user

Expected Response (Failure):

- Status Code 400
 - Returns “Follower/Followee” does not exist
-

Url/Pattern: /AccountType

HTTP Method: PUT

Description:

- Toggles the account type of the user

Expected Request Body Data:

- none

Supported Response Data Types:

- Text

Expected Response (Success):

- Status Code 204
- A Text saying the user successfully toggled an account

Expected Response (Failure):

- Status Code 400
 - Returns error message
-

Url/Pattern: /WatchedMoviesIn

HTTP Method: PUT

Description:

- Adds a movie to users watchedMovies list

Expected Request Body Data:

- The name of the movie to be added (string)
- Example: “Double down”

Supported Response Data Types:

- Application/JSON

Expected Response (Success):

- Status Code 204
- A JSON object of the added movie

Expected Response (Failure):

- Status Code 400
 - Returns a NULL
-

Url/Pattern: /WatchedMoviesOut

HTTP Method: PUT

Description:

- Removes a movie to users watchedMovies list

Expected Request Body Data:

- The name of the movie to be removed from the list (String)

Supported Response Data Types:

- application/json of the removed movie

Expected Response (Success):

- Status Code 204

Expected Response (Failure):

- Status Code 400
 - Returns a NULL
-

Url/Pattern: /review

HTTP Method: PUT

Description:

- Creates a review on the movie that user is viewing

Expected Request Body Data:

- JSON string containing their username, the rating, and the review body
- Example: {"username":"ThePants", "rating":10, "reviewSummary":"very nice", "reviewText":"I had a lot of fun"}

Supported Response Data Types:

- application/json

Expected Response (Success):

- Status Code 204
- A JSON string representing the created review object

Expected Response (Failure):

- Status Code 400
 - If JSON requested
 - Return a NULL object
-

Url/Pattern: /profile

HTTP Method: GET

Description:

- Retrieve profile of user that is logged in

Expected Request Body Data:

- None

Supported Response Data Types:

- text/html
- application/json

Expected Response (Success):

- Status Code 200
- If HTML is requested
 - An HTML page containing the users information will be displayed
- If application/json is requested
 - A user object containing all of the necessary information

Expected Response (Failure):

- Status Code 404
 - If HTML is requested
 - An error 404 page
 - If JSON requested
 - Return a NULL object
-

Url/Pattern: /login

HTTP Method: GET

Description:

- Retrieves the login page

Expected Request Body Data:

- None

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status Code 200
- An HTML page with all the components a user would need to login will be displayed

Expected Response (Failure):

- Status Code 404
 - An error 404 page
-

Url/Pattern: /login

HTTP Method: PUT

Description:

- Will log the user into a new session if credentials are satisfied

Expected Request Body Data:

- A JSON object containing a username and password
- Example: {"username": "ThePants", "password": "computer4321"}

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status Code 204
- A "login successful message"

Expected Response (Failure):

- Status Code 400
 - A message to the user saying "unsuccessful login"
-

Url/Pattern: /logout

HTTP Method: PUT

Description:

- Will log the user out of the session

Expected Request Body Data:

- None

Supported Response Data Types:

- text/html

Expected Response (Success):

- Status Code 204
- A "logout successful message"

Expected Response (Failure):

- Status Code 400

- A message to the user saying “unsuccessful logout”

Each Source Type

GET(read), POST(create), PUT(update),DELETE

*Important what each request does or describe what they are going to do

Put Request

For request that will contain a body, outline what information will be expected within the body

Query Parameters

What parameters will be supported and what they will mean