

## User stories

Each of these headers will typically represent a single endpoint with a relevant name, though some functions might require the front-end/client/user to use multiple endpoints for one single transaction.

### Get Movies by Name

Client sends a search string to the server (e.g. "Arrival")

Server calls the TMDB API with the search parameters, and gets first page of results

Returns these results in to the client in JSON format

### Get Movies By Genre

Upon program start, get the list of genres from the API and store in the database

Client sends request for available genre list with genre IDs

Display list of genres from the database to client

Client sends the genre ID to the server (e.g. 12345)

Server calls the TMDB API with search parameter where genre ID matches

Returns these the first page of results

### Get Movies By Rating

Client sends JSON object with minimum vote average, maximum vote average, and order

Server finds movies in that range using the API, sorting by the given order

Returns these movies to the client

## Intermediate

### Create User

Client sends proposed username and password to the server

Server checks if the password is long enough, and if the username is taken

Password is hashed, and stored in a user table alongside a username and ID

HTTP code is returned accordingly

### Login User

Client sends username and password to server

Server hashes the password to find a match in the database

If authenticated, server creates a session ID and passes it to the frontend, along with user preferences and favourites (if they are implemented)

### User Authentication

Hash password

Compare hash of password to one stored in database

## User Profile:

### Add Genre Preferences

Client requests list of genres, server sends list of genres with IDs

Client sends session ID, and list of genres to be added

Server updates the database with the new genres for the user

### Remove Genre Preferences

Client sends list of genre IDs to be removed from the given session (corresponding to user)

Server removes the genres from the user in the database

## Get Movies By Genre Preferences:

Client sends a request for movies according to the given session's user

Server uses API to find movies matching the genres stored within the user's preferences

These movies are returned to the client

## Advanced (2.0)

## User Profile:

### Add Favourite Movies

Client will send a list of movies to the server, with a session ID

Server will add these movies to the session's user's favourites in the database

Server returns a success response to the client

### Remove Favourite Movies

Client will send a list of movies to the server, with a session ID

Server will remove these movies from the user's favourites in database

Server returns a success response to the client

### Edit username

Client sends valid session ID along with new username

Server checks to see if the new username is taken

Server sets the user's username in the database to the new one

### Edit password

Client sends a valid session ID along with new password

Server checks if password is valid, and updates the stored password hash accordingly.

## Get Favourite Movies

Client sends request alongside session ID

Server gets the favourite movies for that user stored in the database

Server returns this list of movies