

Backend Methods Description:

1. `getPopularMovies` (`MovieController`):

- This method handles GET requests to fetch popular movies.
- It calls the `getPopularMovies` method in the `MovieService` to fetch data from the TMDB API.
- Returns the fetched movie data as a response.

2. `getMovieDetails` (`MovieController`):

- This method handles GET requests to fetch details of a specific movie.
- It calls the `getMovieDetails` method in the `MovieService` to fetch details from the TMDB API.
- Returns the fetched movie details as a response.

3. `createComment` (`MovieController`):

- This method handles POST requests to create a new comment for a movie.
- It calls the `createComment` method in the `CommentService` to create a new comment entity.
- Returns the created comment data as a response.

4. `getPopularMovies` (`MovieService`):

- This method interacts with the TMDB API to fetch data for popular movies.
- Returns the fetched movie data.

5. `getMovieDetails` (`MovieService`):

- This method interacts with the TMDB API to fetch details of a specific movie.
- Returns the fetched movie details.

6. `createComment` (`CommentService`):

- This method creates a new comment entity and associates it with a movie and author.
- Saves the created comment entity in the database.
- Returns the created comment entity.

Frontend Methods Description:

1. `getPopularMovies` (`MovieService`):

- This method makes an HTTP GET request to the backend endpoint for fetching popular movies.
- Returns an observable containing the fetched movie data.

2. `getMovieDetails` (`MovieService`):

- This method makes an HTTP GET request to the backend endpoint for fetching details of a specific movie.
- Returns an observable containing the fetched movie details.

3. `ngOnInit` (`MovieListComponent`):

- This lifecycle hook method is called when the movie list component is initialized.
- Calls the `getPopularMovies` method in the `MovieService` to fetch popular movies.
- Subscribes to the observable returned by `getPopularMovies` to update the UI with the fetched data.

4. `ngOnInit` (`MovieDetailsComponent`):

- This lifecycle hook method is called when the movie details component is initialized.
- Calls the `getMovieDetails` method in the `MovieService` to fetch details of the selected movie.
- Subscribes to the observable returned by `getMovieDetails` to update the UI with the fetched data.

Interactions:

- In the frontend, components trigger service methods to make HTTP requests to the backend.
- Backend controllers handle these requests by calling corresponding service methods.
- Service methods in the backend interact with external API (TMDB API) or perform database operations.
- The response data from the backend is then returned to the frontend, where it is used to update the UI.