

React Task

The idea is to create a single page application, which will allow users to search the Netflix Roulette database.

There will be two pages: main page with search and a page with a particular film.

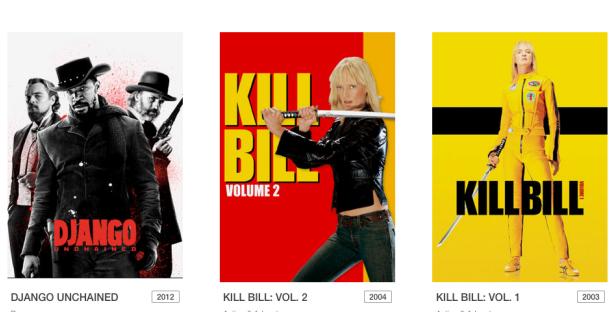
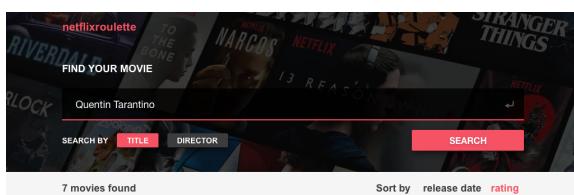
Netflix Roulette API

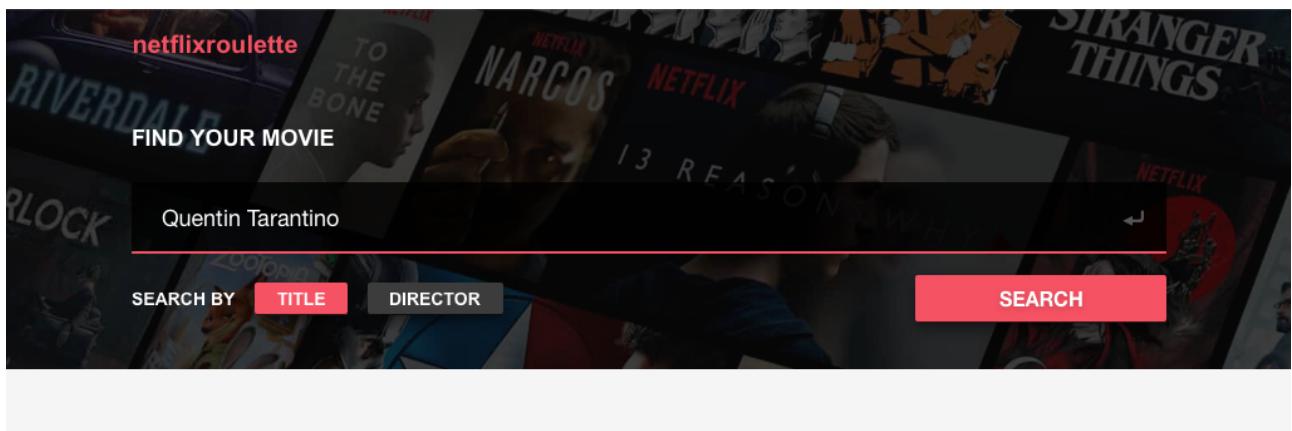
All requests should be performed with AJAX. The API is open, it means that you should not have any problems with cross-origin requests.

You can find an API documentation here: <https://netflixroulette.net/api/>

Interface

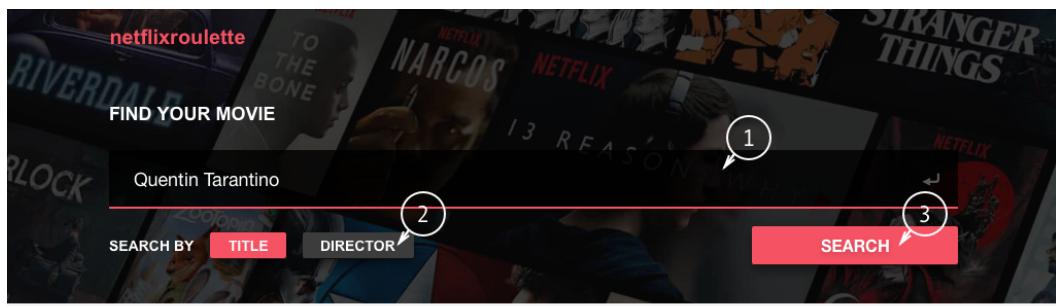
Here are two screens presented with break down. There is also an additional screen with empty state of the main page. Design and captions are available below.





No films found





7 movies found

Sort by release date rating

Red numbered circles (4 through 10) highlight specific elements: (4) points to the "7 movies found" message; (5) points to the "Sort by" dropdown; (6) points to the movie poster for "KILL BILL: VOL. 1"; (7) points to the movie poster for "DJANGO UNCHAINED"; (8) points to the year "2012" in the "DJANGO UNCHAINED" poster; (9) points to the genre "Dramas" in the "DJANGO UNCHAINED" poster; (10) points to the number "10" in the "DJANGO UNCHAINED" poster.

Movie	Year	Genre
KILL BILL: VOL. 2	2004	Action & Adventure
KILL BILL: VOL. 1	2003	Action & Adventure
PULP FICTION	1994	Oscar-winning Movies
JACKIE BROWN	1997	Dramas
RESERVOIR DOGS	1992	Independent Movies
FOUR ROOMS	1995	Comedies
DJANGO UNCHAINED	2012	Dramas





Films by Quentin Tarantino



JACKIE BROWN

1997

Dramas



RESERVOIR DOGS

1992

Independent Movies



FOUR ROOMS

1995

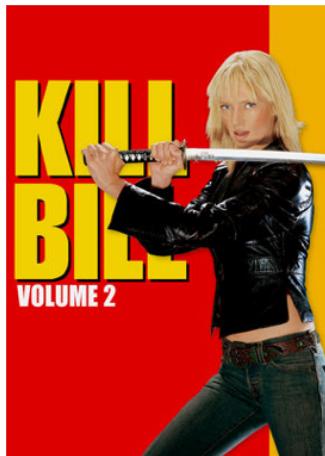
Comedies



DJANGO UNCHAINED

2012

Dramas



KILL BILL: VOL. 2

2004

Action & Adventure



KILL BILL: VOL. 1

2003

Action & Adventure

Captions

0. Empty results state

1. Search field

Enter button should work as well

2. Search filter

By default “Title” is selected. Click to switch option.

3. Search button

4. Results count

5. Results filter

“release date” is selected by default. Click to switch option.

6. Results body

All found items, without pagination

7. Item image

URL for the image you will get from the server

8. Item release date

9. Item genre

10. Item title

11. Film cover

12. Film title

13. Film rating

14. Search button

Returns user to the main page with search

15. Film duration and release year

16. Description

17. Director and cast list

18. Films by the same director

Task 1

Setting up the environment

For this task you will need to setup a simple NodeJS server to implement server-side rendering. We recommend using the Express.

Please install **Webpack** and setup **Babel** for your application.
Try to make your configuration simple, as for the small SPA.

You will also need to install React, Redux, React-Redux, React-Router, Jest and Redux Form.

Task 2

Create the HTML layout with React components for both pages

You should only layout these pages using JSX. They will be used in future tasks.

Try to use all of the React composition power when performing the task:
decompose components into small reusable parts.

Task 3

Create routing for your application.

Link app states between each other with React router.

By default user lands on a new page with empty results state (*caption 0*).

When user performs a new search, you should redirect him to:

localhost/search/**Search%20Query**

When a new user lands on the page with such URL, you should perform search and display results.

When user clicks on a film item, redirect him to:

localhost/film/**Attack%20on%20titan**

On switching search type or sorting type you shouldn't switch any routes.

Task 4

Fetching and managing data with Redux

Read netflixroulette API docs:

<https://netflixroulette.net/api/>

Make your search component perform real AJAX requests.

Connect redux and store all the data in your Redux store.

Task 5

Testing

Please create unit tests for some edge cases for each of your components.

Covering redux reducers with tests is **a plus**.

Task 6

Server side rendering

Implement server rendering in your NodeJS application.