

# REACT.JS

## Lecture 3

# AGENDA

- Recap last lecture.
- Routing
- Axios
- API Integration
- Questions!

# ROUTING

Routing is the capacity to show different pages to the user. That means the user can move between different parts of an application by entering a URL or clicking on an element.

React at its core is a very simple library, and it does not dictate anything about routing so you need to install it first :

```
npm install react-router-dom@5
```

<https://v5.reactrouter.com/web/example/basic>

---

# ROUTING

## **<BrowserRouter>**

A `<Router>` that uses the HTML5 history API (`pushState`, `replaceState` and the `popstate` event) to keep your UI in sync with the URL.

## **<Route>**

The `Route` component is perhaps the most important component in React Router to understand and learn to use well. Its most basic responsibility is to render some UI when its path matches the current URL.

# ROUTING

```
<Link to="/">Home</Link>
```

Provides declarative, accessible navigation around your application.

## Routing hooks :

**useHistory** : The useHistory hook gives you access to the history instance that you may use to navigate.

**useParams** : useParams returns an object of key/value pairs of URL parameters.

# AXIOS

Axios is basically an external library, which is used to make promise based HTTP calls

**Creating instance for shared config :**

```
import axios from 'axios';  
  
export const instance =  
  axios.create({  
    baseUrl: 'API BASE URL',  
  });
```

---



After initialize axios instance you can use it for API calling to integrate with backend server :

```
instance.post('/endpoint_name')  
  .then(function (response) {  
    console.log(response);  
  })  
  .catch(function (error) {  
    console.log(error);  
  });
```

LET'S CHECK AXIOS DOCS



# INTERCEPTORS

Interceptors are functions that Axios calls for every request to transform the request before Axios sends it, or transform the response before Axios returns the response to your code.

<https://github.com/axios/axios>

**There are two types of interceptors:**

- **request interceptor:** this is called before the actual call to the endpoint is made.
  - **response interceptor:** this is called before the promise is completed and the data is received by the then callback.
-

# INTERCEPTORS

To use interceptors you need his two methods for request and another for response :

```
axios.interceptors.request.use(function (config) {  
    // Do something before request is sent  
  
    return config;  
  
}, function (error) {  
    // Do something with request error  
  
    return Promise.reject(error);  
  
});
```

# INTERCEPTORS

To use interceptors you need his two methods for request and another for response :

```
// Add a response interceptor
axios.interceptors.response.use(function (response) {
    // Do something with response data
    return response;
}, function (error) {
    // Do something with response error
    return Promise.reject(error);
});
```

# INTERCEPTORS

Interceptors can be used for example :

- Show loader in request method and hide it in response one
- Set authorization header
- Set Accept-language key based on user selected language
- Handling general errors returned from API

THANK YOU

# LECTURE 3

## LAP

# TASK : MOVIES APP

1 - Add navbar with links for different pages as shown in task

Create account on the movie database website :

<https://www.themoviedb.org/login>

2- get Movies list using API :

[https://api.themoviedb.org/3/movie/popular?api\\_key={apiKey}](https://api.themoviedb.org/3/movie/popular?api_key={apiKey})

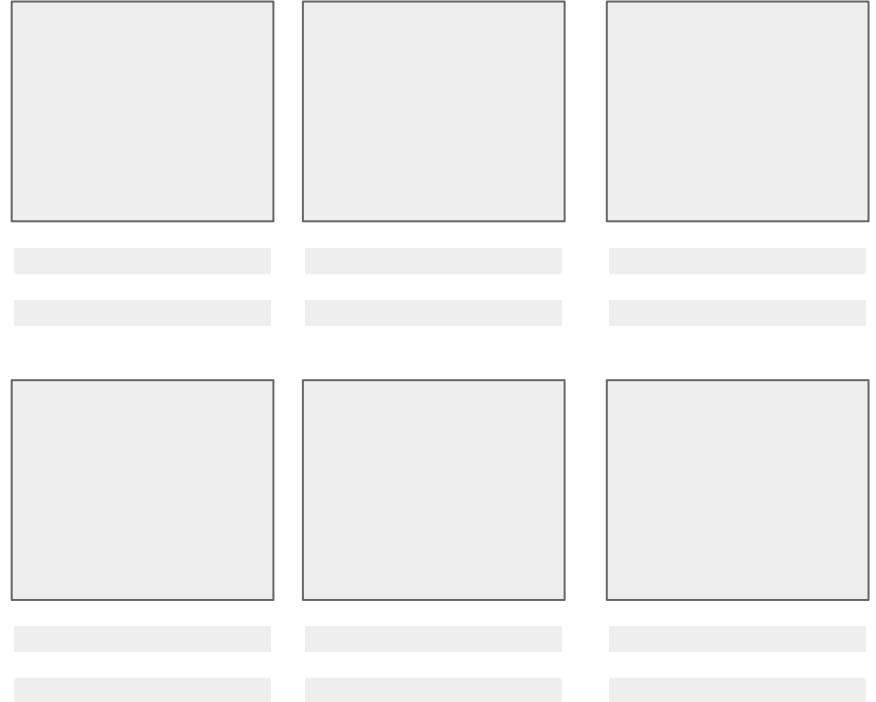
3- using dynamic routes get movie details data using API :

[https://api.themoviedb.org/3/movie/880009?api\\_key={apiKey}](https://api.themoviedb.org/3/movie/880009?api_key={apiKey})

Where 880009 is the movie ID

Movies favorites

login register



# TASK : MOVIES APP

Movies favorites


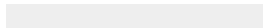
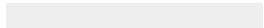
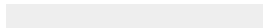
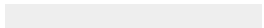
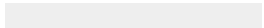
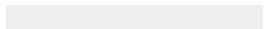
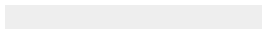
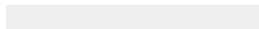
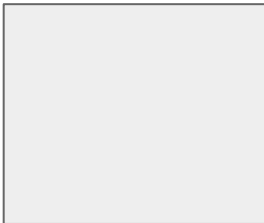
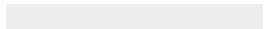
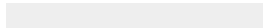
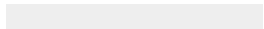
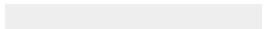
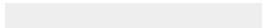
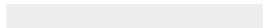
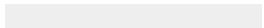
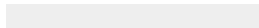
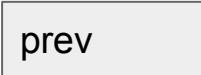

login

4-Handle pagination by send query params to url called **page** [Bonus]

[https://api.themoviedb.org/3/movie/popular?api\\_key=7a1c19ea3c361a4d3cc53eb70ef8298c&page=4](https://api.themoviedb.org/3/movie/popular?api_key=7a1c19ea3c361a4d3cc53eb70ef8298c&page=4)

5- Apply search for movies using the following api

[https://api.themoviedb.org/3/search/movie?api\\_key=7a1c19ea3c361a4d3cc53eb70ef8298c&query={](https://api.themoviedb.org/3/search/movie?api_key=7a1c19ea3c361a4d3cc53eb70ef8298c&query={MovieName})  
[MovieName}](https://api.themoviedb.org/3/search/movie?api_key=7a1c19ea3c361a4d3cc53eb70ef8298c&query={MovieName}) [Bonus]



# TASK : MOVIES APP

For images please add static url  
part to img src before  
poster\_path key

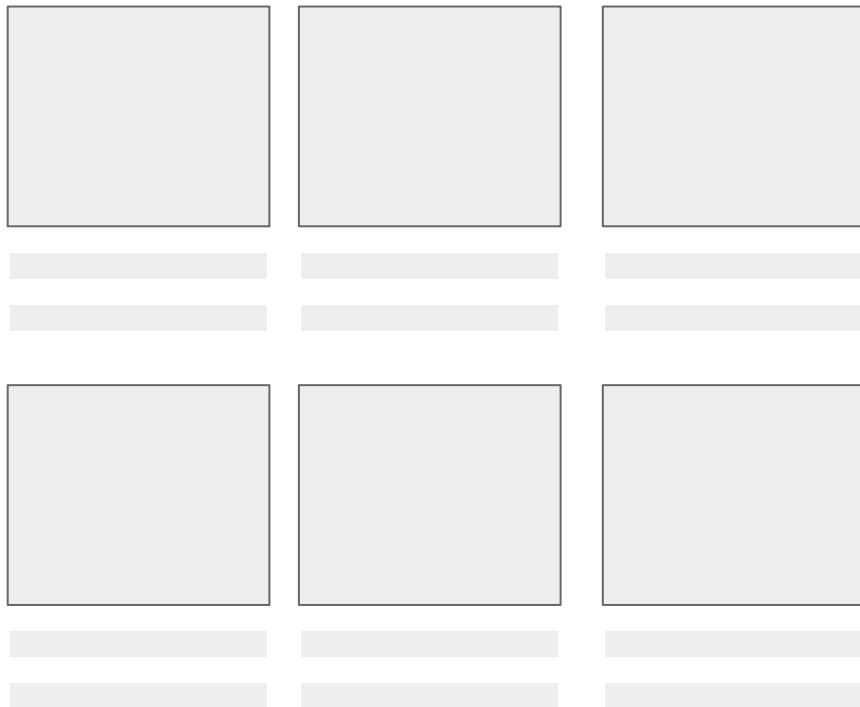
[https://image.tmdb.org/t/p/w500/{poster\\_path}](https://image.tmdb.org/t/p/w500/{poster_path})

Movies DB link:

<https://developers.themoviedb.org/3/movies/get-popular-movies>

Movies favorites

login



Release Date ▾

Filters ▾



**Rogue One: A Star Wars Story**  
Science fiction film/Action



**Moana**  
Fantasy/Adventure



**Arrival**  
Mystery/Science fiction film



**Fist Fight**  
Comedy



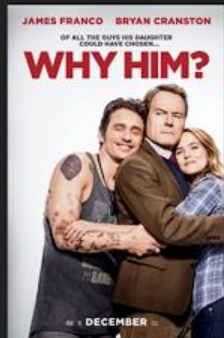
**Get Out**  
Thriller/Horror



**Assassin's Creed**  
Fantasy/Science fiction film

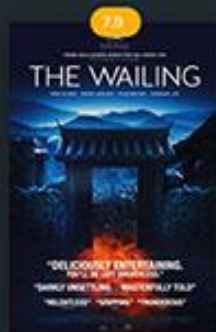


**Doctor Strange**  
Fantasy/Science fiction film





## New Releases



The Wailing

Mystery, Thriller



Warcraft

Action, Adventure, Fantasy



The Space Between Us

Adventure, Drama, Sci-Fi



The Witch

Horror, Mystery



The Shallows

Drama, Horror, Thriller



X-Men: Apocalypse

Action, Adventure, Sci-Fi



Me Before You

Drama, Romance



Now You See Me 2

Action, Comedy, Thriller



The Fits

Drama



Money Monster

Crime, Drama, Thriller