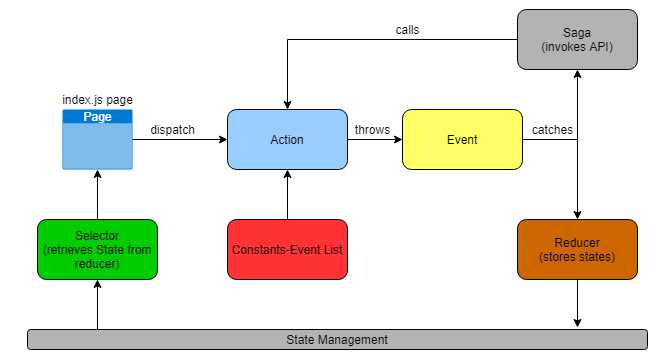
# **REACT JS CONCEPTS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Task 1:**

* [State and Props](https://www.youtube.com/watch?v=DJtI3Pogd88)
* [Arrow *Function*](https://www.youtube.com/watch?v=h33Srr5J9nY&t=159s)
* [Pure Component](https://www.youtube.com/watch?v=PXXjkq4A-OU)
* [Node Package Manager vs yarn](https://www.youtube.com/watch?v=M8nlgBQBxxo)
* [React DOM knowledge](https://www.youtube.com/watch?v=Iw2BLUjQo1E)
* [Class based component and its lifecycle](https://www.youtube.com/watch?v=lnV34uLEzis)
* [Class and ‘this’ keyword](https://www.youtube.com/watch?v=gvicrj31JOM)
* **Functional component and its react hooks -**
* <https://www.youtube.com/watch?v=Cla1WwguArA>
* [https://www.youtube.com/watch?v=-9MvABBFdlw&list=PL7pEw9n3GkoWKbs-](https://www.youtube.com/watch?v=-9MvABBFdlw&list=PL7pEw9n3GkoWKbs-9UGRyne-NlrMTQuzQ) 9 UGRyne-NlrMTQuzQ
* <https://www.youtube.com/watch?v=uGgPINlKqBs>
* <https://www.youtube.com/watch?v=kVyrzn29QPk>
* [Mutable and Immutable](https://www.youtube.com/watch?v=-hi-QQHWlHg)
* [React-redux](https://www.youtube.com/watch?v=7Erbf5NXQQw&list=PL7pEw9n3GkoWgIc-Ambc-QZGcTKEei2O3)
* [Redux-saga](https://www.youtube.com/watch?v=eUMbH6X_Adc)
* [React-loadable and lazy loading](https://www.youtube.com/watch?v=00gQyXRL7KI)
* [Error-Boudrie](https://www.youtube.com/watch?v=DNYXgtZBRPE)
* **Functionality of Redux Saga -**



#### Index.js Page

* This is the main file. Our component code execution starts from this file.
* It dispatches the action and fetch the state from reducer with help of selector.
* Re-render the component on state change.

#### Action

* Action works on a unique constant event string which is declared in constant.js.
* It throws an event, when called.

#### Saga

* It catches the event and invokes an API. And calls an action on the basis of API response example: SUCCESS, ERROR.

#### Reducer

* It catches the event. Create or update the state.
* Update loading to true or false depending upon the action.

#### Selector

* Retrieves the state from reducer.

#### Scenario:

Suppose we need to load user page in UI. A GET\_USER action will be dispatched from index.js file.

Saga catches the GET\_USER event and in turn it will invoke the get users API and returns the response. Reducer will also catch the GET\_USER event and creates a state for user in its state machine. And set loading to true.

Now, Saga will again call an SUCCESS or ERROR action corresponding to success or error response of API. Reducer will catch this event as well and updates the state with result or error depending upon the event and set the loading of data to false.

[Material ui lib](https://material-ui.com/)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Task 2: Udemy training**

Udemy My Learning: <https://www.udemy.com/home/my-courses/learning/>

Username: [**complereinfosystem@gmail.com**](mailto:complereinfosystem@gmail.com)

Password:MyLifeisAwesome

**Courses to complete:**

* [React - The Complete Guide (incl Hooks, React Router, Redux)](https://www.udemy.com/course/react-the-complete-guide-incl-redux/)