**Redux.**

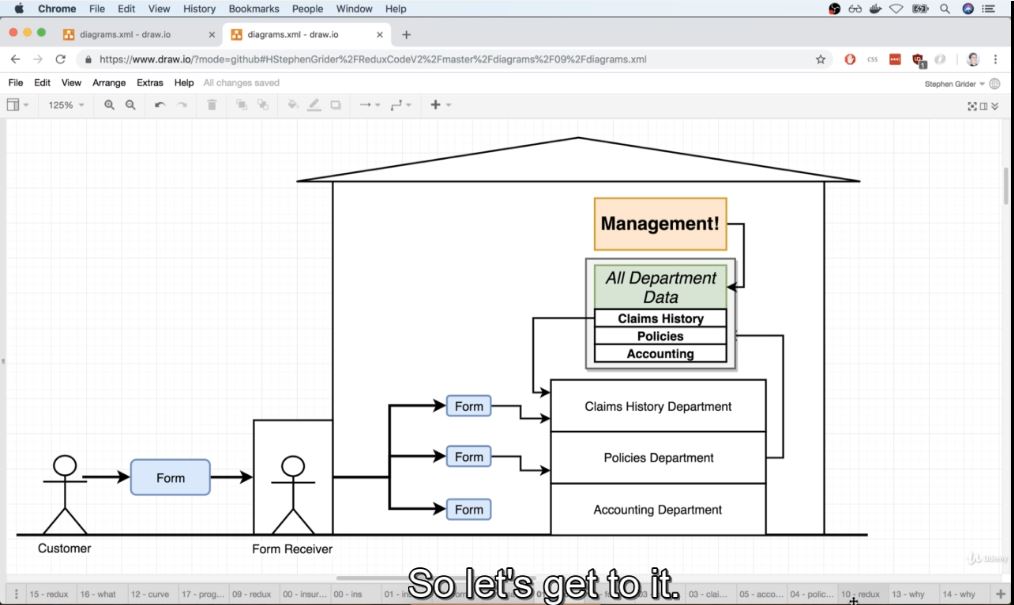
**Installation in React Project**

npm install --save redux react-redux

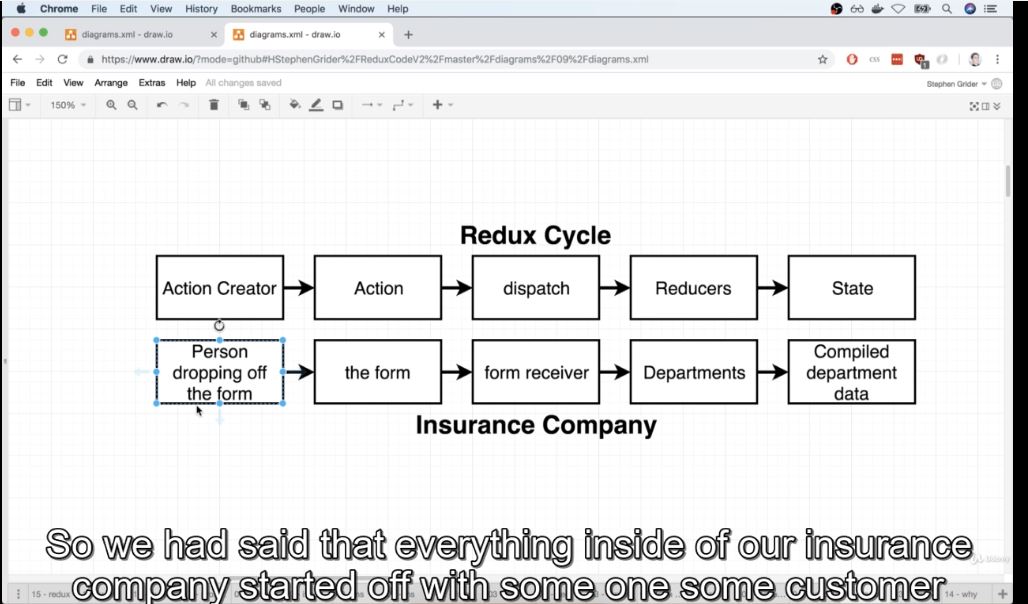
Redux is state management library

With the help of redux creating complex application easier

**Basic scenario**

****

**Redux Cycle**

****

**Action creator** is a function that is going to create or return a plain java script object. We refer to this plain java script object as action

**An action** has a type property and payload property (issurance company ky lehaz sy form ke types hoge 3 like claim wala form ya create policy wala ya delete policy wala or with some amount of money agr new policy bnane)

Type property describe some change we want to make inside of our data

Paload property describe some context around the change that we want to make

The purpose of action is to describe some change that we want to make to the data inside of our application

**Dispatch** form receiver form ke copies bnata ha or 3 department hn claim policy or accounting wala to 3no department ma ak ak copy da deta ha

Dispatch function is going to take in action. It’s going to make copies of that object and then pass it off to a bunch of different places inside of our application

**Reducer** is a function that is responsible for taking in action and some existing amount of data is going to process that action and then make some changes in data and then return is so that it can then be centerailzed in some other location

Same ise trah sy kam krta ha jasy department krta department like claim wala accounting wala or policy wala jb b inky pass form aie ga like new policy wala form aya to policy department policy ko update kry ga

**State** jb departments data ko update krty hn to ak centralized location pa data ko send kr daty hn or isy state kahy gn redux ma

Redux state property is a central repository of all information that has been created by our reducers or by our department

**//actioncreator**

const **createPolicy**=(name,amount)=>{

return { //Action (a form in analogy)

type: 'CREATE\_POLICY',

payload: {

name: name,

amount: amount//agr 2015 ky syntax use kry to name,amount b likh skty srf brack ma

}

}

}

**//Actioncreator**

const **deletePolicy**=(name)=>{

return {

type:'DELETE\_POLICY',

payload:{

name:name

}

}

}

**//action creator**

const **createClaim**=(name,amountOfMoneyToCollect)=>{

return{

type:'CREATE\_CLAIM',

payload:{

name:name,

amountOfMoneyToCollect:amountOfMoneyToCollect

}

}

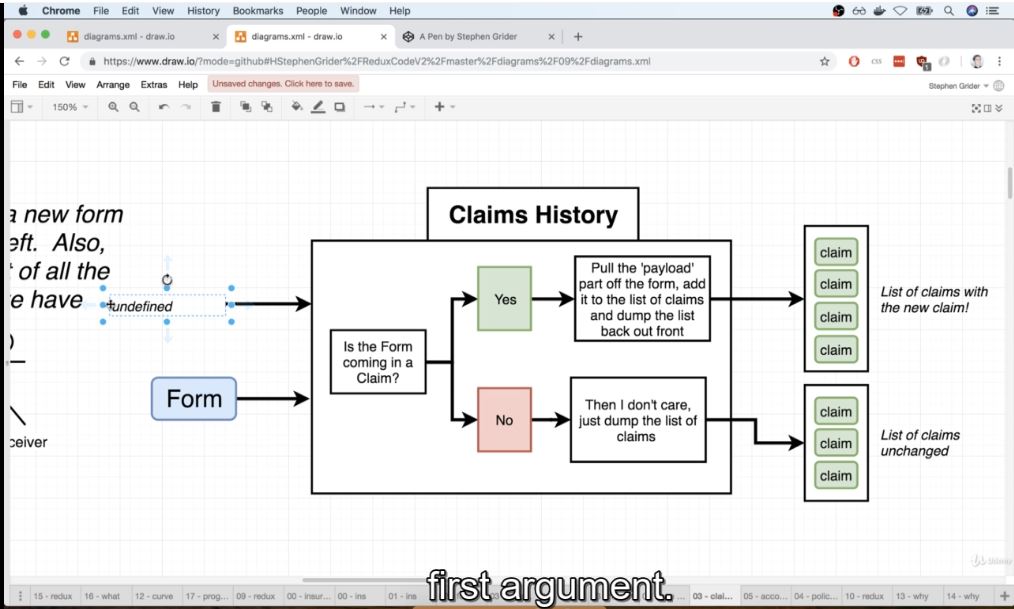
}

Ya oper jo 3 part hn ya action creator ky hn persom 3 types ky form create kr ska ha ak issurance krvany ky leay yaw ala create policy form hoga

Ak issurance remove krvany ky leay to ya delete policy wala form hoga

Ak claim krny ky leay mtleb paisy lany ky leay to ya wala create claims wala form hoga

**Code of Reducer (Departments)**

****

//Reducer (Departments!)

const **claimHistory**=(oldListOfClaims=[],action)=>{ //

if (action.type==='CREATE\_CLAIM')

{

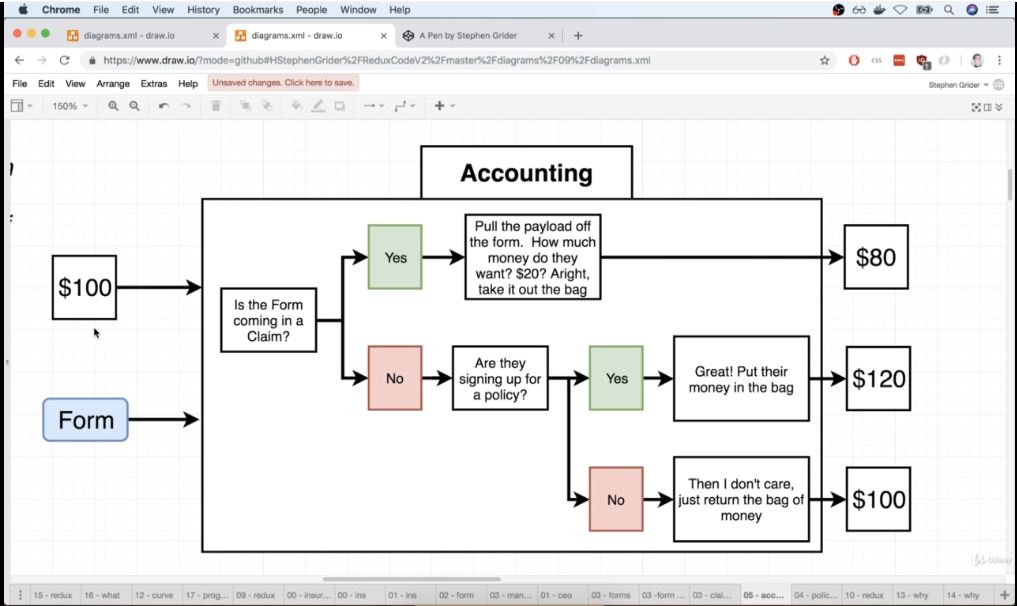
//we care about this action (form)

return[...oldListOfClaims,action.payload];

}

//we dont care about this action

return oldListOfClaims;

}

//Reducer (Department)

const **accounting**=(bagOfMoney=100,action)=>{

if (action.type==='CREATE\_CLAIM')

{

return bagOfMoney-action.payload.amountOfMoneyToCollect

}

else if (action.type==='CREATE\_POLICY')

{

return bagOfMoney + action.payload.amount

}

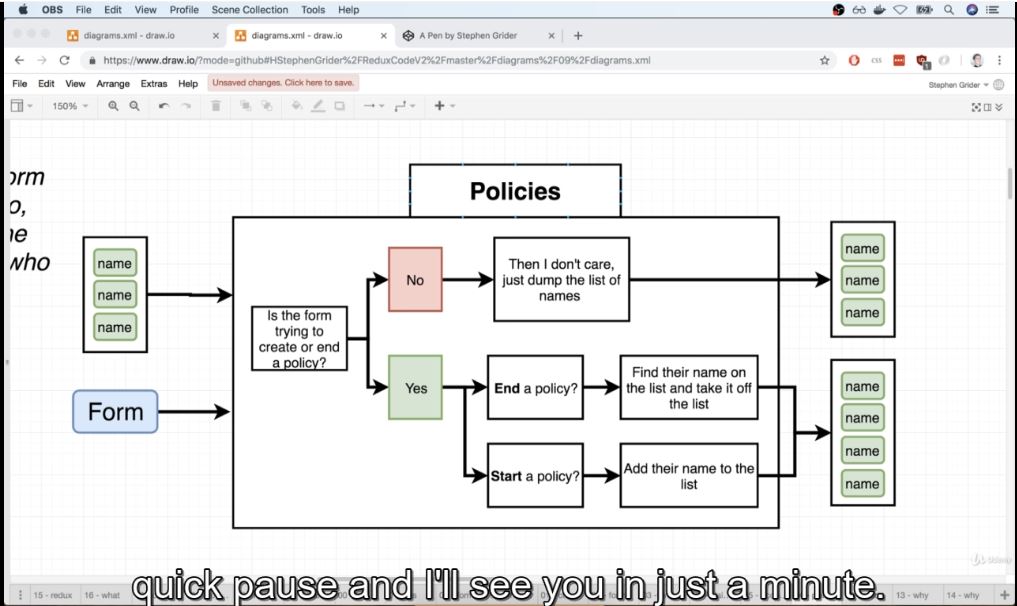
else

{

return bagOfMoney;

}

}



const **policies**=( listOfPolicies = [] , action)=>{

if (action.type==='CREATE\_POLICY')

{

return[...listOfPolicies , action.payload.name]

}

else if (action.type==='DELETE\_POLICY')

{

return listOfPolicies.filter(name=>name!==action.payload.name);

}

else

{

return listOfPolicies;

}

}

Ab ager hm dkhy har reducer ky first parameter ma hmny ak default value de ha vo is case ma ky ager company abe bne e ha to pehly sy data given ni hoga like pehly sy kise na issuarnce ni ke hoge ise leay listOfpolicies=[] ha or ise trah oldListOfClaims=[] empty list ky sth default value de ha or bagOfMoney ko 100 sy default value de ha ky company ka account ma start ma 100 dollars thy

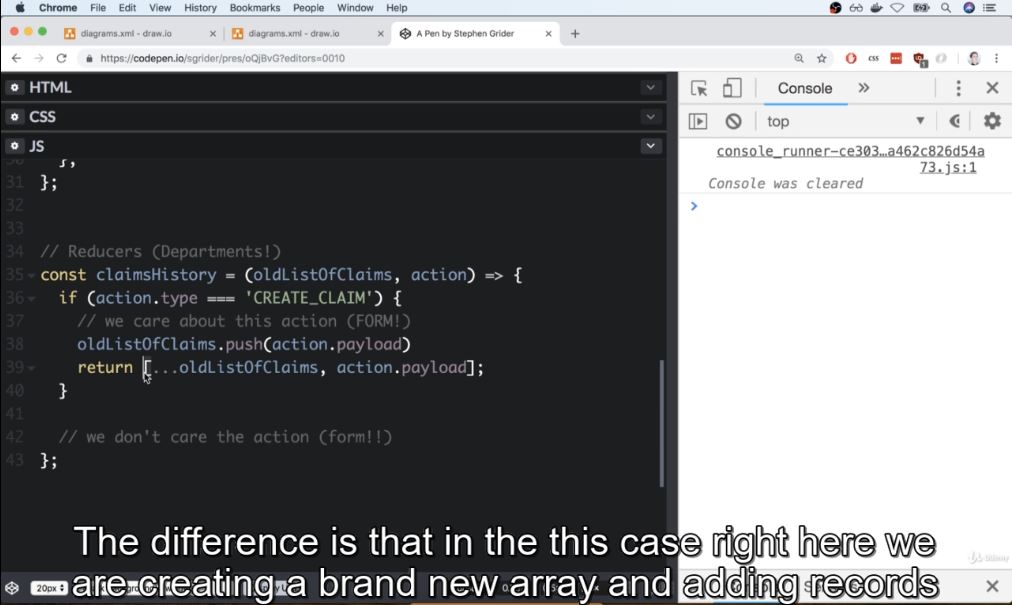
**Filter method and [...listOfPolicies+action.palyload.name]**

listOfPolicies.filter(name=>name!==action.payload.name);

yaha filter function use hoa ha filter function new array return krta ha or existing array ko change ni krta

ya deletepolicy waly department ma use hoa ha jo person issurance remove krvana chata ha to uska name remove kr k yak new list return kry ga jismy usky elva baki sbky name ho gn

or …listOfPolices ka mtleb ha ky



oldListOfClaims.push(action.payload) sy existing array ky last ma ak new number add ho jae ga but […oldListOfClaims, action.payload] sy ak new array bny ge or usmy existing array ke sary element add ho gn or last ma new element be add ho jae ga.

Redux library



createStore or combineReducers 2 function hn redux library ky

combineReducer ma sary reducers ko pass kr dn gn key ky sth

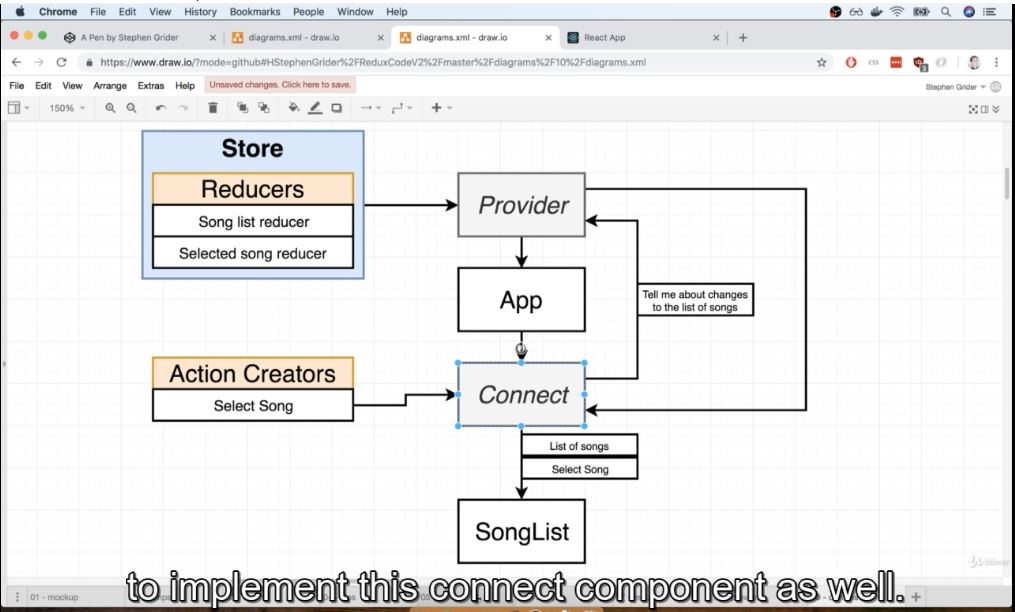
**Redux Cycle summary**

Jb bhe data ko change krna ho to action creator ko call krty hn or phr dispatch ma action ko pass krty hn dispatch actions ke copies bnata kr sb har reducer ko transfer kr deta ha

**Provider**

Ya react-redux library ka ak component ha

**SONGS App (react-redux library)**

****

Store ma sary reducers hoty hn or sare states hote hn application ke is store ko as a prop pass krty hn provider ma.provider sb sy pehly render hota ha application ma. Provider ky pass interntal reference hota ha store ka. Provider basically app ka sb components ko information provide krta ha

Connector function provider sy communicate kr skta ha or ya props system ky through communicate ni krta blky ya context system ky through communicate krta ha. **Context system** allow krta ha any parent component to directly communicate with any child like parent or child ky drmiyan baki or b component hn but context system sy parent direct communicate kr skta ha child sy jasy is digram ma app centre ma ha or parent provider ha vo songList ky sth communicate kr skta ha jo mjy smej ai ha shaid ase e ha

**Named Export and Default Export. And Index.js file in Folder**

Ya nechy example ma named export ha ab isy kise b component ma import krna ho to curly bracket lgae gn but agr ya default export hota to curly bracket na lgtay

export **const** selectSong=(song)**=>**{

    return{

        type:'SONG\_SELECTED',

        payload: song

    };

};

Or agr isko kahe b import krvana ho to

Import {selectSong} from ./folderName jisma ya ha /yaha file ka name

Or agr Default export hota jasy ky nechhy given ha to isko import is trah krty

Import selecSong from ./folderName/fileName

*//Action creator*

**const** selectSong=(song)**=>**{

    return{

        type:'SONG\_SELECTED',

        payload: song

    };

};

export default selectSong;

t

File ka name tbhe ae ga jb index.js ky elava koi or name ho or agr index.js e name ha to ya na b likhy to tb b yahe index.js file import ho ge q ky agr hm folder ka name likhy but file ka name na likhy to index.js file e impot hote ha.

*//Action creator*

export **const** selectSong=(song)**=>**{

    return{

        type:'SONG\_SELECTED',

        payload: song

    };

};

Action creator na selectSong action create kia ha jo ky plain javascript object ha jiski 2 properties hn yaha ak type or dosre payload

Reducer folder ma index.js file

import {combineReducers} from  'redux';

*//Reducers*

**const** songReducer=()**=>**{

    return [

        {title: 'mary pass tum ho',duranton : '5:12'},

        {title: 'tu mera ni',duranton : '5:00'},

        {title: 'koi chand rkh',duranton : '4:20'},

        {title: 'kala rang',duranton : '3:12'},

        {title: 'bhola record',duranton : '2:10'}

    ]

}

**const** selectedSongReducer=(selectedSong=null,action)**=>**{

if (action.type==='SONG\_SELECTED')

  {

    return action.payload;

  }

  return selectedSong;

}

export default combineReducers({

    songs:songReducer,

    selectedSong:selectedSongReducer

});

2 reducer bnae hn songReducer or selectedSongReducer

combineReducer jo ky redux ka ak function ha usy import kia ha or isko call kia ha jiski waja sys b reducers combine ho geay hn

Main index.js file

import React from 'react';

import ReactDOM from 'react-dom';

import {Provider} from 'react-redux';

import {createStore} from  'redux';

import App from './components/App';

import reducers from './reducers';

ReactDOM.render(

    <Provider store={createStore(reducers)}>

    <App/>

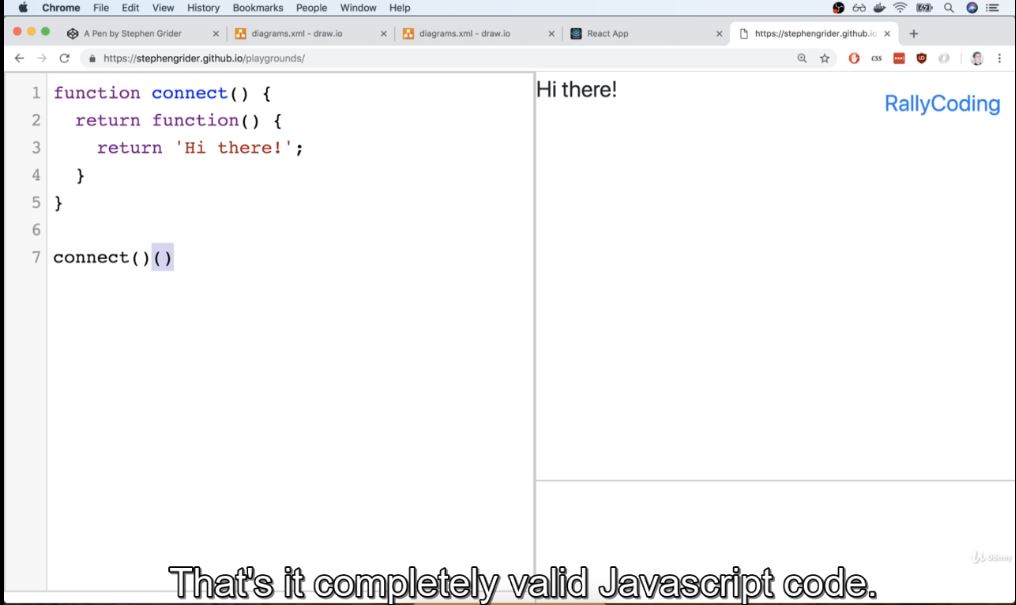
    </Provider>,document.querySelector('#root')

);

Provider ak component ha react-redux ka jo information provide krta ha app component ko

createStore sy ak store create ho jae ga jismy sb reducers hn or isko store ma assign kia ha store ko provider component ma pass kia ha

**Connect Function**

****

**Connect function** ky syntax ma ma pehle jo braces hn vo function ko return krte hn or dosre braces returned function ko invoke/call krte hn.

**Rendering a list with redux data**

import React from 'react';

import {connect} from 'react-redux';

**class** SongList **extends** React.Component {

    renderList(){

        return this.props.songs.map((song)**=>**{

            return (

                  <div className="item" key={song.title}>

                      <div className="right floated content">

                          <button className="ui button primary">

                              SeLect

                          </button>

                      </div>

                      <div className="content">{song.title}</div>

                  </div>

            );

        })

    }

    render(){

        return <div className="ui divided list">{this.renderList()}</div>;

    }

}

**const** mapStateToProps=(state)**=>**{

    console.log(state);

    return state;

}

export default connect(mapStateToProps)(SongList);

**How we make use of react-redux library to get data from redux store to a component..??**

1. Sb sy pehly connect ko import krty hn react redux ma with curly bracket
2. mapStateToProps ko define krty hn
3. connect ko call krty hn mapStateToProps first argument hota ha or 2sra argument vo component hota ha jismy data ly k rana chaty hn jasy yaha SongList ka component ha
4. mapStateToProps jo return krta ha vo as a prop hota ha render ma

**Calling action creator from component.**

1. Sb sy pehly to hm action connect ko import kry gn component ma
2. Phr connect function ma isy pass kry gn call krty wkt
3. Or is trah phr connect function action creator ko as a prop pass kr dy ga render function ma

import React from 'react';

import {connect} from 'react-redux';

import {selectSong} from '../actions';

**class** SongList **extends** React.Component {

    renderList(){

        return this.props.songs.map((song)**=>**{

            return (

                  <div className="item" key={song.title}>

                      <div className="right floated content">

                          <button

                          className="ui button primary"

                          onClick={() **=>** this.props.selectSong(song)}

                          >

                              SeLect

                          </button>

                      </div>

                      <div className="content">{song.title}</div>

                  </div>

            );

        })

    }

    render(){

        return <div className="ui divided list">{this.renderList()}</div>;

    }

}

**const** mapStateToProps=(state)**=>**{

    console.log(state);

    return {songs: state.songs};

}

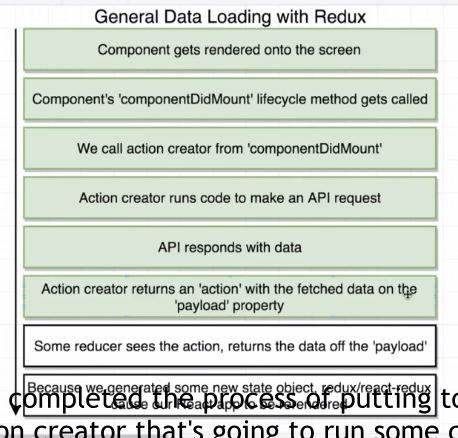
export default connect(mapStateToProps,{selectSong})(SongList);

connect function automatically dispatch ma action creator ko pass kr deta ha

dispatch jasy ky pta e ha action ke copies sb reduecer ma pass kr deta ha or phr jis reduecer ky sth action ke type match ho jate ha usky according state change hote ha.

**Blog Application React Redux.**

**General data loading API**

****

**Main Index file**

import React from 'react';

import ReactDOM from 'react-dom';

import {Provider} from 'react-redux';

import {createStore} from  'redux';

import App from './components/App';

import reducers from './reducers';

ReactDOM.render(

    <Provider store={createStore(reducers)}>

    <App/>

    </Provider>,document.querySelector('#root')

);

**App component**

import React from 'react';

import PostList from './PostList';

**const** App=()**=>**{

return <div className="ui container"><PostList/></div>;

};

export default App;

**Reducer**

import {combineReducers} from 'redux';

export default combineReducers({

    replace:()**=>**'sddasd'

})

**Action**

import jsonplaceholder from '../apis/jasonPlaceholder';

export **const** fetchPosts=**async** ()**=>**{

**const** response = await jsonplaceholder.get('/posts');

    return {

        type: 'Fetch\_POSTS',

        payload: response

    };

}

**Axios command**

npm install –save axios

**Api**

import axios from 'axios';

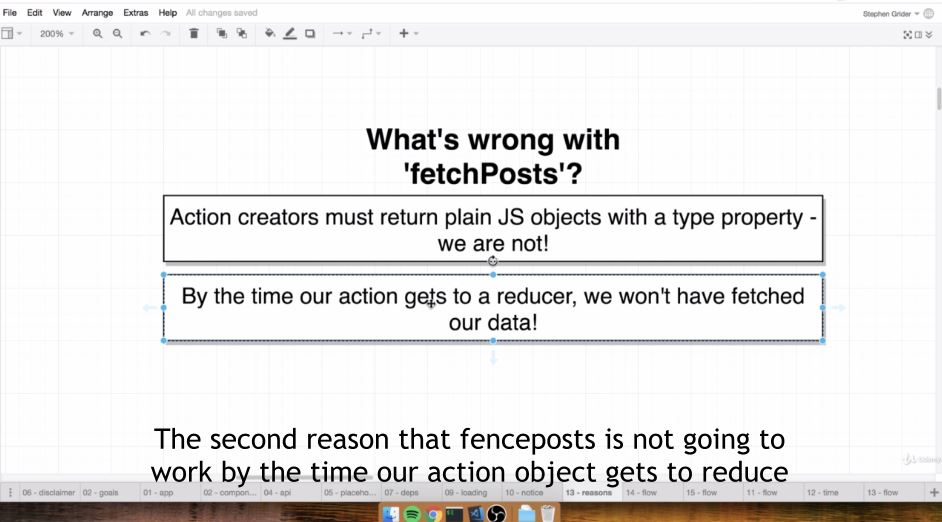
export default axios.create({

baseURL: 'https://jsonplaceholder.typicode.com/'

});

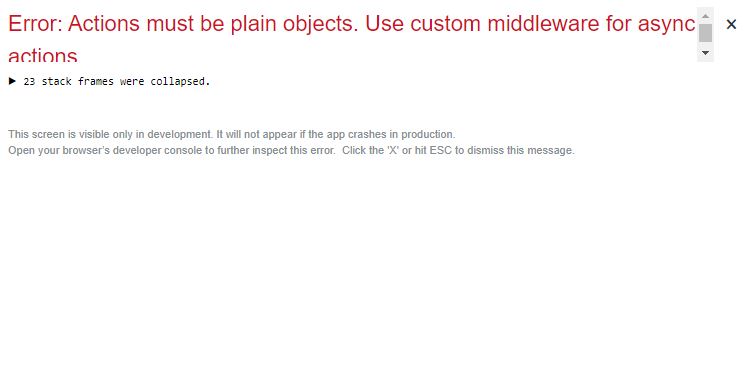
**Explanation.**

Major two issue pic ma btae geay hn jinki waja sy api sy data get ni ho ra simple react waly method sy jasy api call krty thy



Simple sa structure ha program ka ab ak cheez note kry action creator ma hmy request object mil raha ha but jasy ky hmy pta ha action creator java plain object return krta ha or ise leay error aie ga jo ky nechy given ha

Pehla issue ya tha or isky error ke pic nechy given ha



2sra issue

Ab agr hm action creator ma promise wala method use kry to phr request object return ni hoga or error hath jae ga jasy nechy given ha. Agr promise wala method bho0ola hoa ha to react wale file ma explain kia hoa ha

import jsonplaceholder from '../apis/jasonPlaceholder';

export **const** fetchPosts=()**=>**{

**const** promise =  jsonplaceholder.get('/posts');

    return {

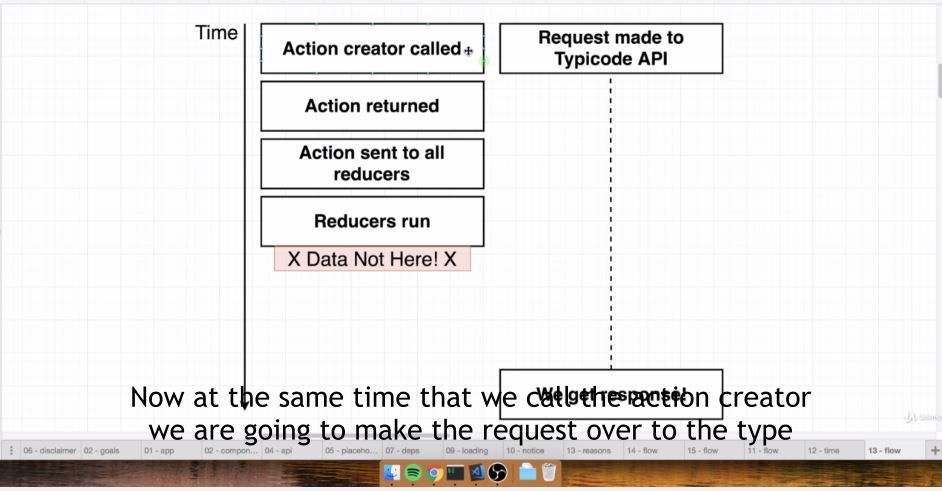
        type: 'Fetch\_POSTS',

        payload: promise

    };

}

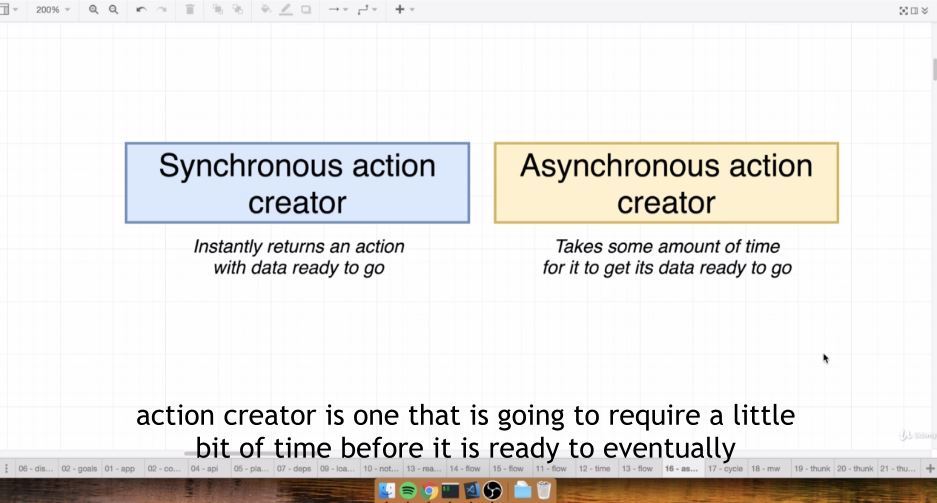
Is sy error to remove ho jae ga but data fetch ni kr skty



Action create hony sy reducer tk pass hony tk bohat he kam time lgta ha but api sy data get krny ma bohat time lgta ha 10 sec ma lg skty or 10 minute tk b ise leay jb

Jb hm action creator ko call kr rhy hn use time api ko b call kr rhy hn ky data get krny ky leay. Jasy ky action create hony sy reducer ma pass hony tk bot km time lgta ha to reducer action ma promise object ko dekhy ga but q ky request abe complete ni hoe or hm reducer ko wait ni krva skty ya esa koi method ni ha ky hm delay krvae or jb request complete ho jae to jo data fetch ho vo reducer ma a jae

**Asynchronous action creator**

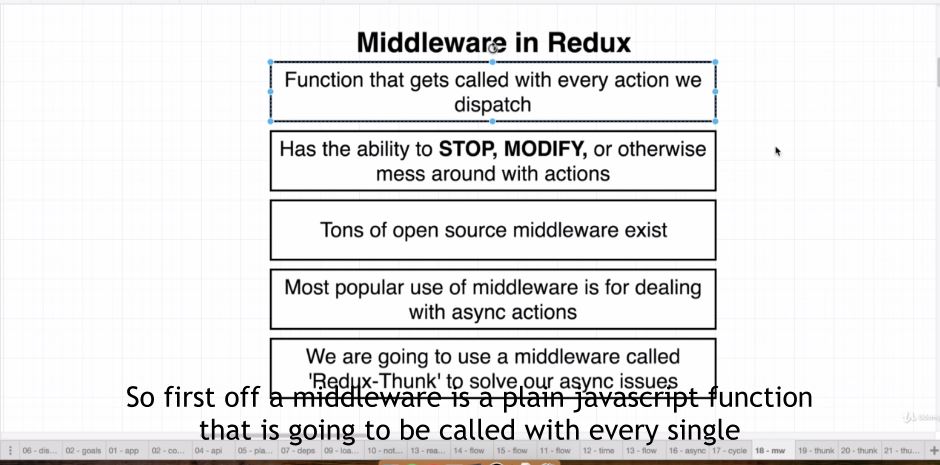
****

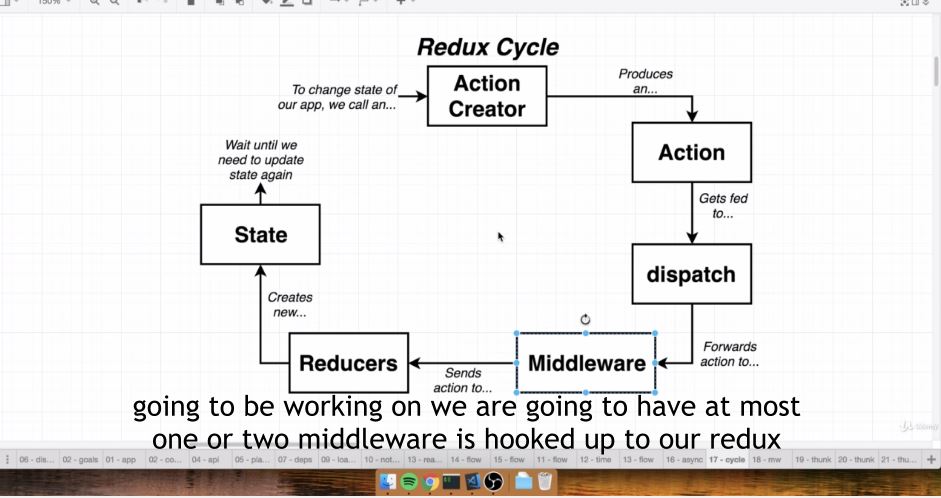
Ab oper wale jo problems the jo vo asynchronouse action creator sy ho solve ho jae ge asynchronous creator thora wait krta ha or jb data ready ho jata ha tb reducer ma pass hota ha

**Redux-Thunk**

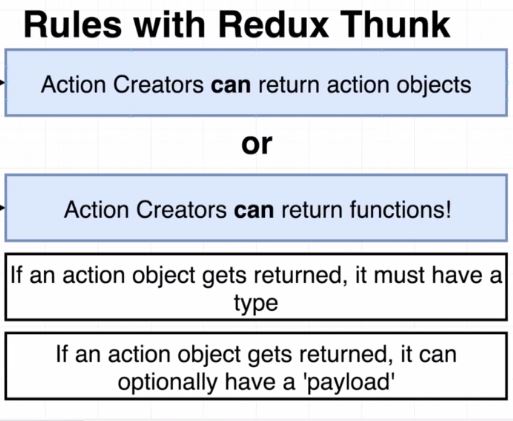
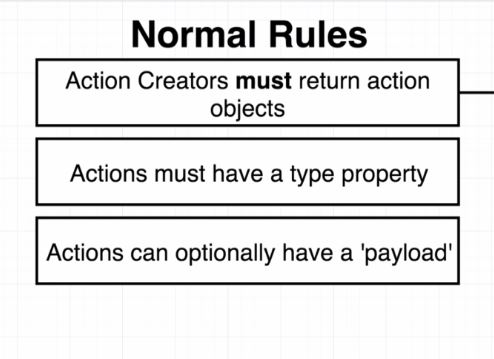
npm install –save redux-thunk

Or asynchronous creator ky leay middle layer ke zaroret prte ha or ya middle layer redux thunk ha

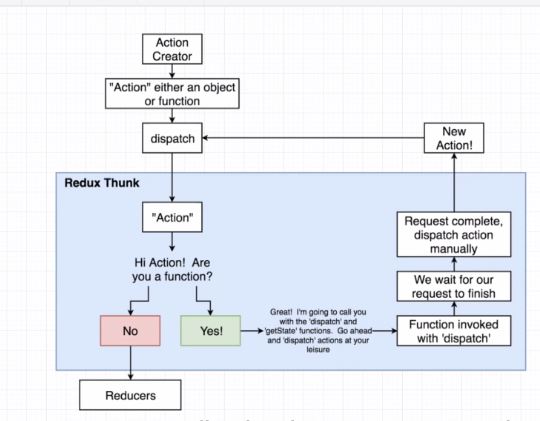




Normal action creator Rule and action creator rule with Redux Thunk



**Flow of action creator with redux thunk.**

****

import jsonplaceholder from '../apis/jasonPlaceholder';

export **const** fetchPosts=**async** ()**=>**{

    return **function**(dispatch,getState){

**const** promise = await jsonplaceholder.get('/posts');

    }

    return {

        type: 'Fetch\_POSTS',

        payload: promise

    };

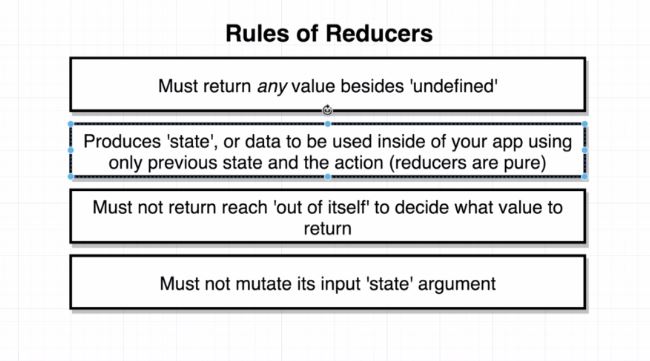
}

We can pass action into the dispatch function .Those actions will be sent through all of our different middle layer and eventually forwarding it to reducers

getState can be called on our redux store that will return all over data inside our store

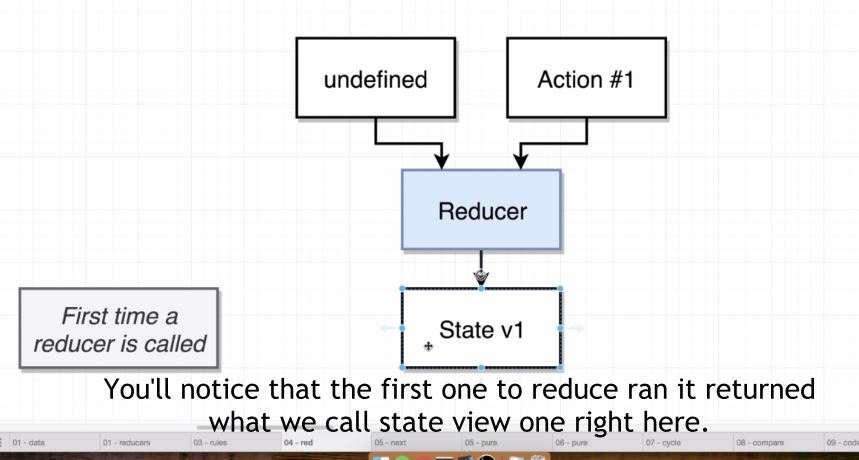
Through dispatch we can change any data we want and through getState we can read or access any data that we want.

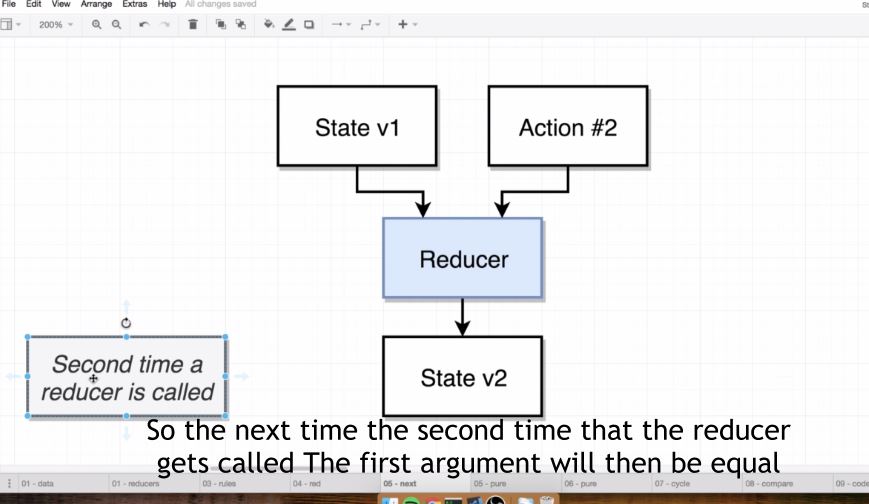
**Rule of Reducers**



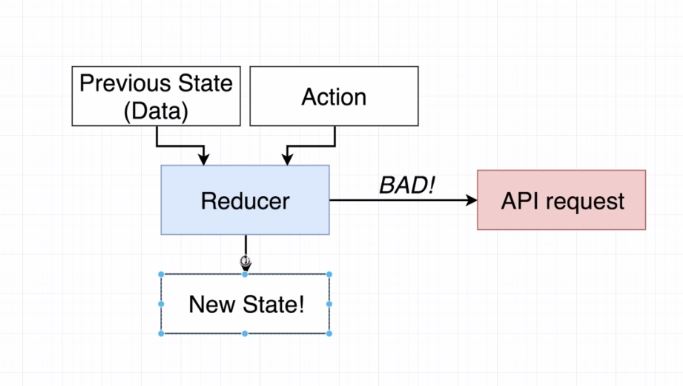
Pehla rule ya ha ky reducer koi value return zaror krta agr undefined kry gn mtleb ku6 b return na kry to error aie ga

2sra rule ya ha ky reducer jo return krta ha next time reducer call hota ha to jo previous value reducer return krta ha vo state ma hote ha



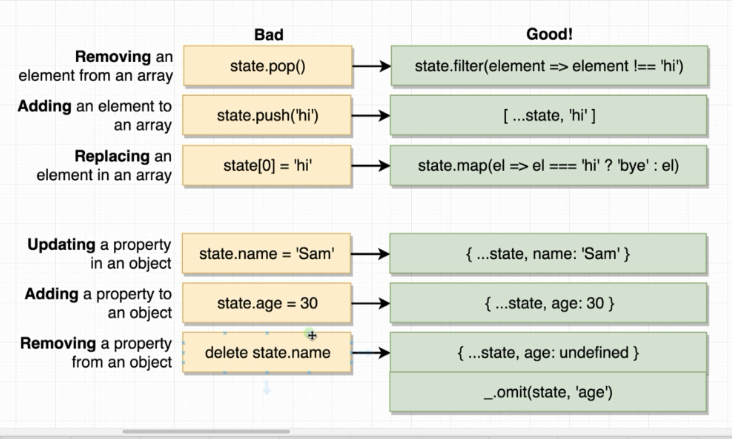


3sra rule ya ha ky reducer sy srf state or action ma e operation perform kr skty hn reducer ma but network request wagera ni kr skty ya isky elava ku6 or ni kr skty. Nechy digram fiven ha



Last rule jo ha vo simple beginner ky leay likhty hn ky mutate (change) ni kr skty but corner cases hoty hn jidher change kr skty hn explanation ky keay **Redux Store Desgin video 7 A Misleading Rule**

**Array and Object Function**

****

**PostList component ma data load kia ha redux way (action creator ma api call ke)**

import React from 'react';

import {connect} from 'react-redux';

import {fetchPosts} from '../actions';

import UserHeader from './UserHeader';

**class** PostList **extends** React.Component{

    componentDidMount(){

        this.props.fetchPosts();

    }

    renderList(){

        return this.props.posts.map(post **=>**{

            return (

                <div className="item" key={post.id}>

                  <i className="large middle aligned icon user"/>

                   <div className="content">

                       <div className="description">

                          <h2>

                              {post.title}

                          </h2>

                          <p>

                              {post.body}

                          </p>

                       </div>

                       <UserHeader userId={post.userId}/>

                   </div>

                </div>

            );

        })

    }

    render(){

        return <div className="ui relaxed divided list">{this.renderList()}</div> ;

    }

}

**const** mapStateToProps=(state)**=>**{

        return {posts:state.posts};

    }

export default connect(mapStateToProps,{fetchPosts})(PostList);

**postReducer**

Post Reducer simple ha bs is dfa switch statement wala syntax use kia ha

**const** postReducer= (state =[], action)**=>**{

switch (action.type){

    case 'FETCH\_POSTS':

    return action.payload;

    default:

        return state;

}

};

export default postReducer;

**action creator for fetching post api data**

ya method ha jis sy action creator ma api ko call krty hn with the help of middle layer called redux-thunk

import jsonplaceholder from '../apis/jasonPlaceholder';

export **const** fetchPosts=()**=>** **async** dispatch**=>**{

**const** response = await jsonplaceholder.get('/posts');

       dispatch( {type: 'FETCH\_POSTS',payload: response.data});

};

**Fetching userData**

**postList Component**

import React from 'react';

import {connect} from 'react-redux';

import {fetchPosts} from '../actions';

import UserHeader from './UserHeader';

**class** PostList **extends** React.Component{

    componentDidMount(){

        this.props.fetchPosts();

    }

    renderList(){

        return this.props.posts.map(post **=>**{

            return (

                <div className="item" key={post.id}>

                  <i className="large middle aligned icon user"/>

                   <div className="content">

                       <div className="description">

                          <h2>

                              {post.title}

                          </h2>

                          <p>

                              {post.body}

                          </p>

                       </div>

                       <UserHeader userId={post.userId}/>

                   </div>

                </div>

            );

        })

    }

    render(){

        return <div className="ui relaxed divided list">{this.renderList()}</div> ;

    }

}

**const** mapStateToProps=(state)**=>**{

        return {posts:state.posts};

    }

export default connect(mapStateToProps,{fetchPosts})(PostList);

**UserHeader component**

import React from 'react';

import {connect} from 'react-redux';

import  {fetchUser} from '../actions';

**class** UserHeader **extends** React.Component{

    componentDidMount(){

        this.props.fetchUser(this.props.userId);

        console.log(this.props.userId);

    }

render(){

**const** user =this.props.users.find((user)**=>**user.id === this.props.userId);

     if (!user){

         return null

     }

     return <div className="header">{user.name}</div>

   }

}

**const** mapStateToProps=(state)**=>**{

    return {users: state.users};

}

export default connect(mapStateToProps,{fetchUser})(UserHeader);

**Action creator for fetching userdata**

export **const** fetchUser=(id)**=>async** dispatch**=>**{

**const** response = await jsonplaceholder.get(`/users/${id}`);

       dispatch( {type: 'FETCH\_USER',payload: response.data});

}

**userReducer**

**const** userReducer= (state =[], action)**=>**{

    switch (action.type){

        case 'FETCH\_USER':

            return [...state,action.payload];

        default:

            return state;

    }

    };

    export default userReducer;

postList component ma userHeader component ko use kia ha userHeard Component ma postList sy data jo get ho raha vaha api ma ak property userid ha ya as a prop pass kia ha userHeader component ma.

userHeader ma componentdidMount method ma action creator fetchUser ko call kia ha or us ma userid ko as a prop pass kia ha ab data vohe fetch ho ga jo userid sy match hoga

actioncreator sy api call hoe or data userreducer ma aya or data reducer ma […state,action.payload] is sy state ma data add hoga or new array return hog easy postList component ma map function lgaya hoa ha to jitney list ma data hoga itne dfa userHeader component chly ga or hard fa id jo hoge vohe data get hoga or phr action creator userReducer ma […state,action.payload] is sy state ka previous data ky sth new data push hoga or new array return hoge.

Or phr hmny userHeader ma find wala function use kia ha

**const** user =this.props.users.find((user)**=>**user.id === this.props.userId);

     if (!user){

         return null

     }

     return <div className="header">{user.name}</div>

   }

Find wala function



Specific value find krta ha find function

Ab condition yah ha ky jo userId postList sy a rai ha vo agr equal hoe userReducer wale list ky id sy to uska name e render hoga

**Little issue with the above code in UserHeader component**

Dekha jae to 2 props use ho rhy hn is component ma render method ma ak componentDidMount ma postList wala prop or 2sra mapToState sy user ke list ya zeada thek ni ha agr complex application ho to hmy phr is ma issue ho skta ha

Ise leay ab mapToState ma hmny vo condition lga de ha

**const** mapStateToProps=(state,ownProps)**=>**{

    return {user: state.users.find((user)**=>**user.id === ownProps.userId)};

}

**ownProps** is leay use kia ha ky render method ka prop use krna ho agr mapToState method ma to hmy 2nd parameter dena prta ha state ky sth, 2nd parameter ownProps ha ya refferenc dy ra actual prop jo ha render ma pass hoa ha.

Or ak chez note kry to is fetchUser action creator itne dfa call hoga jite postList ma userid a raii ha repetation id b ho ge to tb b fetchUser action call hoga or ya thek ni ha ise leay is repetation waly cheez khtem krny ky leay hm action creator ma chaning kry gn

**import \_ from 'lodash'**

import \_ from 'lodash';

import jsonplaceholder from '../apis/jasonPlaceholder';

export **const** fetchPostsAndUsers =()**=>** **async** (dispatch,getState)**=>**{

      await dispatch(fetchPosts());

**const** userIds=\_.uniq(\_.map(getState().posts,'userId'));

       userIds.forEach(id**=>** dispatch(fetchUser(id)));

};

export **const** fetchPosts=()**=>** **async** dispatch**=>**{

**const** response = await jsonplaceholder.get('/posts');

       dispatch( {type: 'FETCH\_POSTS',payload: response.data});

};

export **const** fetchUser= (id)**=>async** dispatch**=>**{

**const** response = await jsonplaceholder.get(`/users/${id}`);

               dispatch( {type: 'FETCH\_USER',payload: response.data});

        };

fetchPostsAndUsers ak new action creator bnaya ha or is ma fetchPost ko call kia ha or getState sy post reducer ka data get kia ha or phr

=\_.uniq(\_.map(getState().posts,'userId')

\_.uniq ke help sy repetation id khtem ho jae ge or userIds ma ak list a gai ha unique id ke

Or phr forEach loop sa dispatch fetchUser ab call hoga userIds ke value pass kr ky userIds ma ab unique id values hn

userIds.forEach(id**=>** dispatch(fetchUser(id)));

or postList component ma jo new action creator bnaya ha isko call kry gn or userHeader component ma koi b action creator call ni kry gn.

userHeader Component

import React from 'react';

import {connect} from 'react-redux';

**class** UserHeader **extends** React.Component{

render(){

**const** {user}=this.props; *//destructring the prop*

   if(!user){

       return  null;

   }

     return <div className="header">{user.name}</div>

   }

}

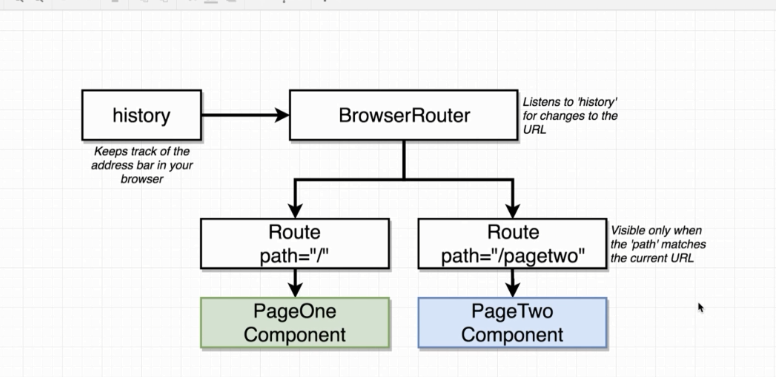
**const** mapStateToProps=(state,ownProps)**=>**{

    return {user: state.users.find((user)**=>**user.id === ownProps.userId)};

}

export default connect(mapStateToProps)(UserHeader);

**Stream Client App**

**ReactRouter work kasy krta ha digram sy chal jae ga pta**

Jb hm application bnaty hn or browser pa load krty hn to hm browserRouter ka instance create krty hn jasy nechy code ma given ha. BrowserRouter internally ak apna object create krta ha “History” ya history object address bar ma jo url hota ha usy dkhta ha (ya address bar ke bat ise lea ke ha ky hm direct address b dy skty hn component ka but nechy code ma link tag ma address given ha) or ya domain ky bad waly url ko extract krta ha or phr history object is path ko communicate krta ha browser router sy or phr browserRouter is path ko communicate krta ha apny sb component sy or phr jisky sth path match hota ha vo wala component show ho jata ha baki sb hide ho jaty hn

import React from 'react';

import {BrowserRouter, Route , Link} from 'react-router-dom';

**const** PageOne=()**=>**{

    return <div>

               PageOne

               <Link to="/pagetwo"> Navigate to page two  </Link>

        </div>

};

**const** PageTwo=()**=>**{

    return     <div>

               PageTwo

               <Link to="/">Navigate to page one</Link>

               </div>

}

**const** App=()**=>**{

    return (

        <div>

            <BrowserRouter>

               <div>

              <Route path="/"  exact component={PageOne}/>

              <Route path="/pagetwo"   component={PageTwo}/>

              </div>

            </BrowserRouter>

        </div>

    )

}

export default App;

Html tag ka use q ni krty



hm html ma < a href=path> click on this </a>

Is trah kr ky navigation krty hn different pages par.

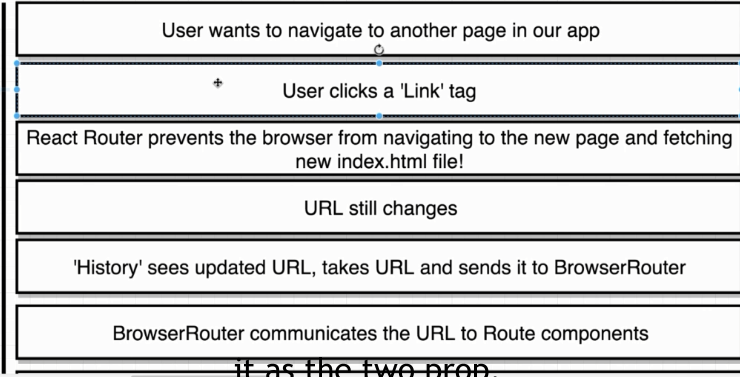
React ma agr is trah html tag ka use kry to har dfa link pa click krny sy hm new html document ke request krty hn or jo data load hoa hota ha screen pa vo sara khtem ho jata ha.

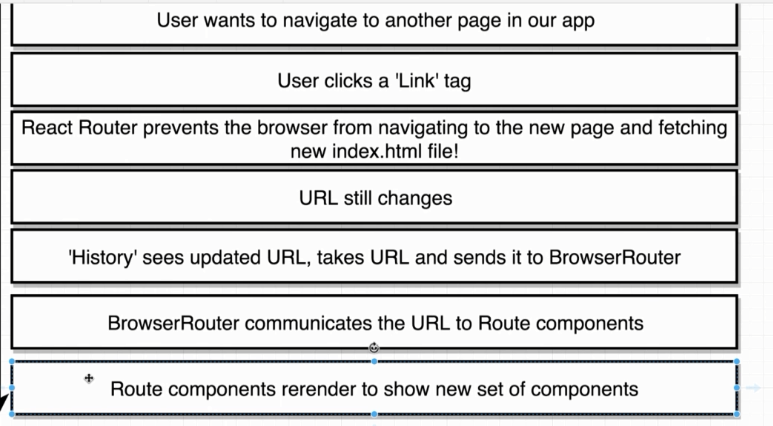
React jo ky single page app ha is ma hm link ka tag use kry gn isko use krny sy new html document ke request ni hoge.

Single Page App ka mtleb ha ky single html document load hota ha

But when the navigate around they still are making use of the same document.

We are just showing and hiding different set of component base upon url. We are essentially tricking the user into thinking that they are really going to different pages. But infact we are just showing and hiding different components.





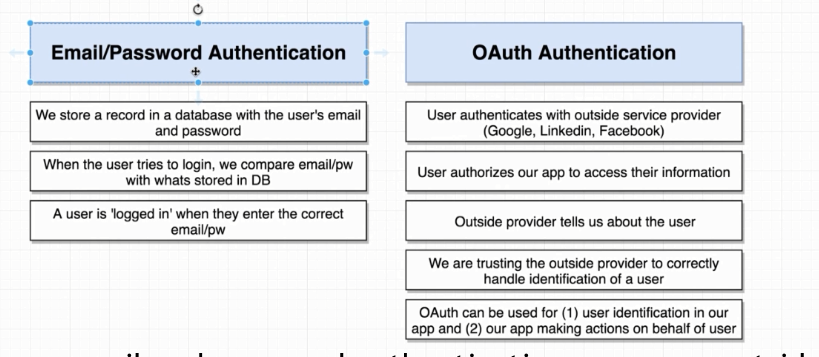
**GoogleAuth.**

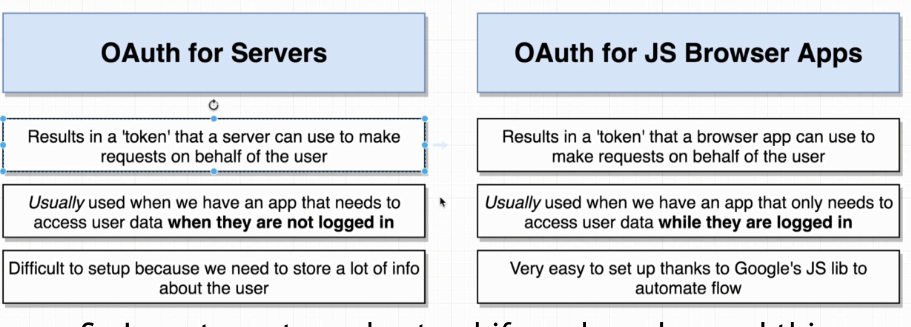
OAuth Authentication simple ya ha ky user ke authentication hote ha some outside provider sy

Outside provider like google ,facebook etc.

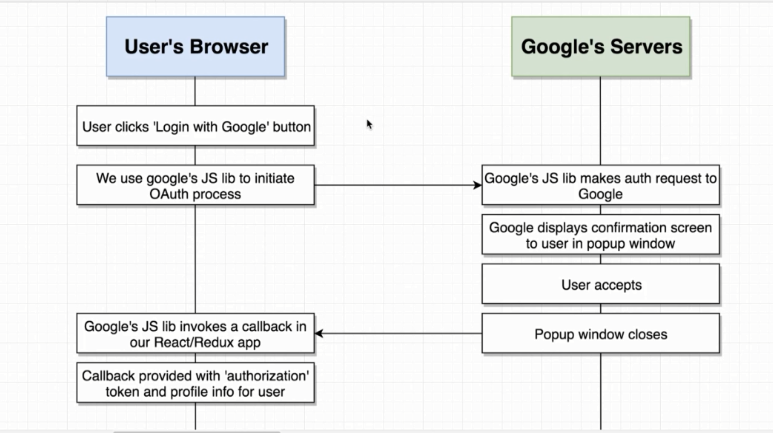
Outside provider user ka bary ma btata ha. Or hm outsider pa trust krty hn ky vo user ke correctly identification kr raha ha.

OAuth Authentication 2 trah sy use hote ha ak to user ke identification ky leay or dosra ya ky hmare app action ly sket ha user ky behalf pa

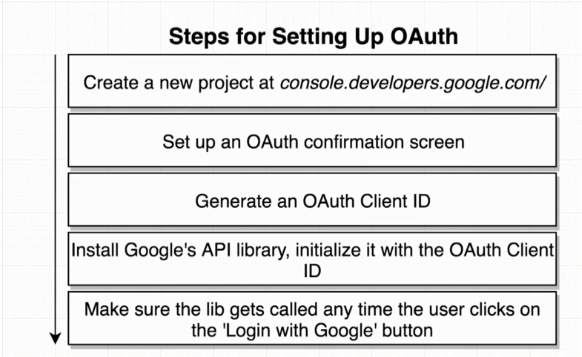
****

**OAuth for JS Browser Apps**

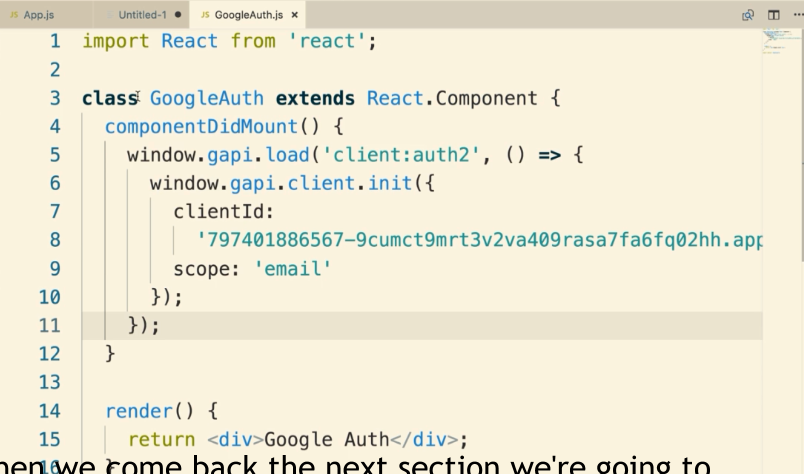
**Flow of OAuth Authentication**

****

**Steps for Google Auth.**

****

1

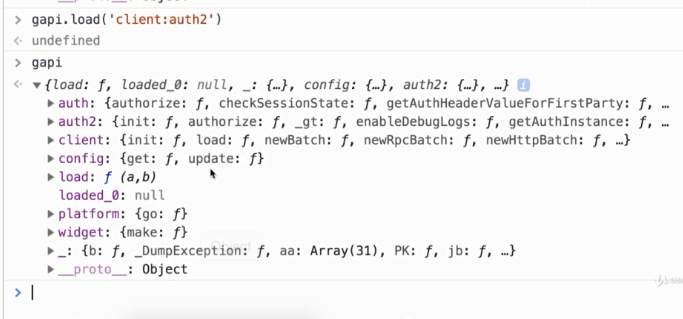


Window.gapi.load(‘client:auth2’)

Is sy google api load ke ha extra code ky sth ‘client:auth2’

Extra code jo ha is sy hmary pas baki or function b a jae gn agr hm srf gapi.load likhty hn to simple ya load ka mtleb ha ky load up some internal library by making a follow up request over to Google server and fetching some additional java script code and then adding to this google library

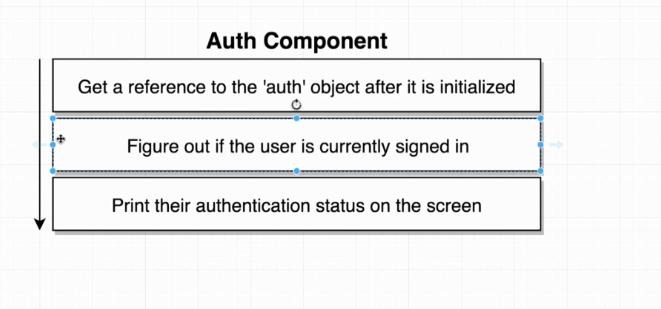


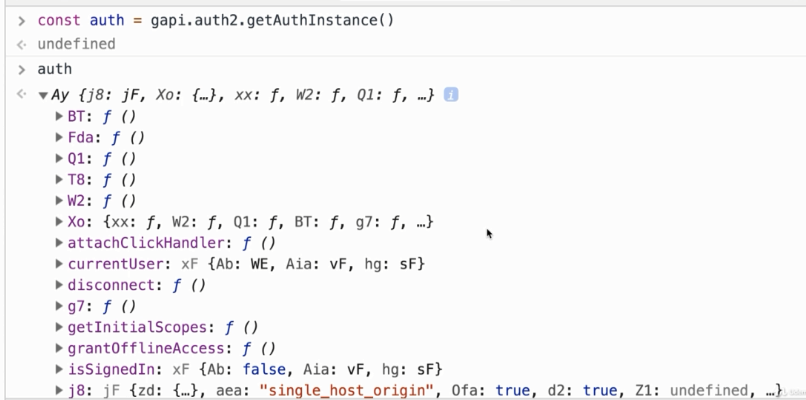


Ab client:auth ky sth load kia to kafi function mil geay hn or jo hmy chae sign

Client:auth2 library jb proper sai trah load ho jae ge tb callback function sy apni app ko client id ky sth initialize kry gn

2.





Ab gapi ko auth2.getAuthInstance ky sth run kry gn to ya google object return kry ga is sy hmy jo chae mil jae ga signed in or out wagera



Ab hmny google object sy isSignedIn ko call kry gn or is sy hmy pta chal jae ga ky signed in ha ky ni

componentDidMount q ky ak dfa e render hota ha to ise leay component level state bnae gn jismy object ke value store krva dn gn



Ab hm manaual console sy sign in krvae gn or phr hmy state sap ta chal jae ga agr sign in ho geay hoe to I am signed in show ho ga agr ni signed out hoe to I am not signed in ka msg show ho ga or agr ab start ma ku6 b ni kea to I don’t know if we are signed in display hoga

Ab hm manually ni kry gn event handler sys y kry gn onCLick sy sigin in or sign out kry gn.

Call back function bnaya onSignIn or onSignOut ka or in dono ma this.auth.signIn or this.authsignOut ko call kia ha or in callback function ka reference dia ha eventhandler ma or trah react component ma Auth authentiacation complete ho jate ha



Ab ya jo listen wala method ma this.AuthChange ko call kia ha is sy text b change hoga like agr hm signout kry gn ya sign in kry gn to listen method ke waja sy ya refresh krna ni pry ga without refresh e change hoga text wrna pehly refresh krty thy to hmy change nazer ata tha



**Redux Form**

**Redux-devtool**-extension is a browser extension that is going to allow you to very easily debug your redux store

<https://github.com/zalmoxisus/redux-devtools-extension>

yaha sy hm iske extension add kry gn chrome ma phr apni application ma exta code add kry gn main file ma. composeEnhancer wala code

is extension sy hmy store ma sare state ka pta chal jata ha or hm kise b state pa asani sy jump b kr skty hn. Redux-devtool sy redux form library learn krna easy ho jate ha

import React from ‘react’;

import ReactDOM from  ‘react-dom’;

import App from ‘./components/App’;

import {Provider} from ‘react-redux’;

import {createStore, applyMiddleware,compose} from ‘redux’;

import reducers from ‘./reducers’;

**const** composeEnhancers=window.\_\_REDUX\_DEVTOOLS\_EXTENSION\_COMPOSE\_\_||compose;

**const** store=createStore(

    reducers,

    composeEnhacers(applyMiddleware())

    );

ReactDOM.render(

    <Provider store={store}>

     <App/>

    </Provider>,

    document.querySelector(‘#root’)

)

**Debug session.**

Jb hm application ko refresh krty hn to all data inside our redux store falls away and we loss it.but jb hm debug session ko enable krty hn to redux dev tool is going to make sure that all the data stays around between the refreshes of the page

Refresh krny ky bad b redux store ka data loss ni hoga

**Redux Form library**

npm install –save redux-from

library ko install kia ha project ma

**Redux Form documentation url**

redux-form.com

**Reducer Index file ka code**

import {combineReducers} from 'redux';

import {reducer as formReducer} from 'redux-form';

import authReducer from './authReducer';

export default combineReducers({

    auth: authReducer,

    form: formReducer

});

as formReducer jo likha ha vo name dia ha reucer ko reducer ak named component ha ise leay bracket sy import kia ha redux-form library sy

import React from 'react';

**createForm Reducer.**

import React from 'react';

import {Field , reduxForm} from 'redux-form';

**class** StreamCreate **extends** React.Component{

    render(){

        return(

            <form>

                <Field name="title"/>

                <Field name="description"/>

            </form>

        )

    }

}

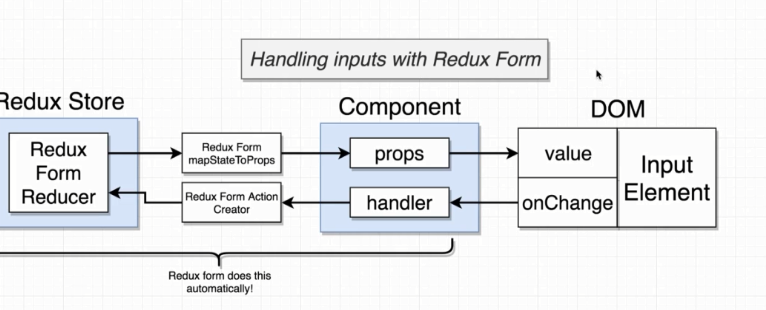
export default reduxForm({

    form: 'streamCreate'

})(StreamCreate);

Field ak component ha or reduForm ak function ha ya similar ha connect function ky reduForm ma ak object pass kry gn.

**Handling input with ReduxForm**

****

Sb kam automatically ho jata ha bs hm na last kam khud krna hota ha ky value or onChange property ko inputElement sth wireup kry or

import React from 'react';

import {Field , reduxForm} from 'redux-form';

**class** StreamCreate **extends** React.Component{

    rednerInput(props){

    return <input onChange={props.input.onChange}

            value={props.input.value}/>

    }

    render(){

        return(

            <form>

                <Field name="title" component={this.rednerInput}/>

                <Field name="description" component={this.rednerInput}/>

            </form>

        )

    }

}

export default reduxForm({

    form: 'streamCreate'

})(StreamCreate);

Ab oper waly component ko or shor kr skty hn hm hmny onChange prop ko assign kia ha onChange property ko or asy he value prop ko assign kia ha value property ko

onChange or value prop hmary passs redux-form library sy mily hn streamCreate form ka name ha ya nechy reduxForm ma as a object use kia ha

**Short Syntax**

import React from 'react';

import {Field , reduxForm} from 'redux-form';

**class** StreamCreate **extends** React.Component{

    rednerInput(formProps){

    return <input {...formProps.input}/>

    }

    render(){

        return(

            <form>

                <Field name="title" component={this.rednerInput}/>

                <Field name="description" component={this.rednerInput}/>

            </form>

        )

    }

}

export default reduxForm({

    form: 'streamCreate'

})(StreamCreate);

Short syntax ma ya ha ky ya formprop ke input property ly ga. Or ya input property ma sary objects hn callback wagera sb onChange b value b asy bohat sy sary hn input ky leay hoty hn or ya sb jo ky input property ma aty hn redux form ma to input property ko he add kr dia ha input element ma.

Inshort yak y input object (jo pehly ma na property kahe the) jis ma sb properties hn isko add kr dia ha as a prop input element ma

Thora or short kr skty hn with the help of destructring

import React from 'react';

import {Field , reduxForm} from 'redux-form';

**class** StreamCreate **extends** React.Component{

    rednerInput({input}){

    return <input {...input}/>

    }

    render(){

        return(

            <form>

                <Field name="title" component={this.rednerInput}/>

                <Field name="description" component={this.rednerInput}/>

            </form>

        )

    }

}

export default reduxForm({

    form: 'streamCreate'

})(StreamCreate);

**Submit Form**

**React submit form**

React ma form ko is trah submit krty hn ak call back bnaty hn or usmy event.preventDefault ko call krty hn is sy enter ke gai value enter krny ky bad gaib ni hote or ab yahe same kam hm redux ma thora sa different trah sy krty hn

state={term:''}

    onFormsubmit(event){

        event.preventDefault();

    }

    render(){

        return (

         <div className="ui segment">

             <form onSubmit={this.onFormsubmit} className="ui form">

**Redux form sumbit**

Redux ma call back function bnaty hn but usmy event.PreventDefault ni likhty redux ka handler function automatically ya kr deta ha

import React from 'react';

import {Field , reduxForm} from 'redux-form';

**class** StreamCreate **extends** React.Component{

    rednerInput({input,label}){

    return (

        <div className="field">

            <label>{label}</label>

            <input {...input}/>

       </div>

    )

    }

    onSubmit(formValues){

    }

    render(){

        console.log(this.props)

        return(

            <form onSubmit={this.props.handleSubmit(this.onSubmit)} className="ui form">

                <Field name="title" component={this.rednerInput}  label="Enter  Title"/>

                <Field name="description" component={this.rednerInput}  label="Enter  Description"/>

                <button className="ui button primary"> Submit</button>

            </form>

        )

    }

}

export default reduxForm({

    form: 'streamCreate'

})(StreamCreate);

Ab onSubmit callback sy hmny jo data enter kia ha get kr skty hn hm