

course github \Rightarrow <http://bit.ly/reventsapp>

facebook / create-react-app

\rightarrow npm will be available

install node.js & Javascript execution engine \Rightarrow 10.15.3 LTS

in the project folder \Rightarrow project name

npx create-react-app revents --use-npm

cd revents / \Leftarrow go into the folder

code . \Leftarrow open in VSCode \supset it's not working

\nexists $\text{F1P} \Rightarrow$ show All Commands \Rightarrow search path \Rightarrow shell command: Install "code" command in PATH

\wedge \Rightarrow Toggle Terminal

npm start \Leftarrow open the project in chrome

file \Rightarrow auto save

∇ web pack injects JS into HTML

∇ Hot Module Replacement

index.js \supset

```
const rootEl = document.getElementById('root');
let render = () => {
  ReactDOM.render(<App />, rootEl)
```

}

if(module.hot){

```
  module.hot.accept('./App', () => {
    setTimeout(render);
```

)

)

render()

∇ folder struct

SCR

app

layout

App.css

App.jsx (renamed)

logo.svg

features

∇ React APIs

render()

constructor()

componentDidMount()

componentWillUnmount()

componentDidUpdate()

setState()

JSX

React Hooks

Raster VS. Vector

∇ install node version manager: <http://bit.ly/2Hn8HjG>

∇ Free images \Rightarrow pixabay.com

check version \Rightarrow node --version

nvm -h & node version manager helper

nvm ls & different versions

nvm use v8.11.2 & switch version

∇ extensions:

Auto Rename Tag

Bracket Pair Colorizer

Debugger for Chrome

click F1 \Rightarrow change No configurations to new config and choose chrome

change Localhost:8000 to 3000

"webRoot": " $\{\text{workspaceFolder}\}/src$ "

VS Code 257 React/Redux/React-Native/JS snippets

ESLint & check against es rules

Material Icon Theme

Prettier - code formatter shift + Alt + F

command + , (comma) \Rightarrow open settings search Prettier

React Redux Firestore Course Snippets

S-

Alt + shift + F \Rightarrow reformat

// App.jsx \Rightarrow named import
import React, { Component } from 'react';

class App extends Component{

render(){

<div className='app'>

<h1> Re-vents </h1>

</div>

);

}

export default App;

DOM: Document Object Model

∇ if not working \Rightarrow try: npm start

∇ React Developer tools \Rightarrow add to chrome

// index.js

```
import React
import ReactDOM
import App from './app/Layout/App'
import * as serviceWorker from './serviceWorker';
```

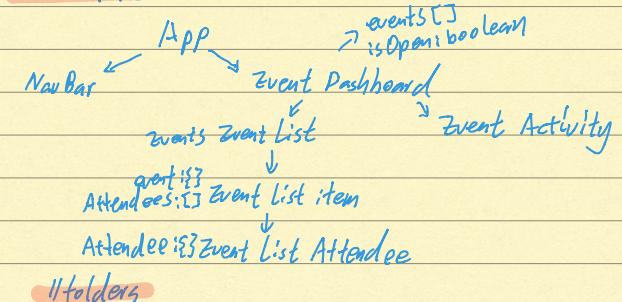
* if it is passed in from a parent via props ∈ Not state

* if not change over time ∈ Not state

* if can be computed based on other state or prop ∈ Not state

```
const rootEl = document.getElementById('root');
let render = () => {
  ReactDOM.render(<App />, rootEl)
}
if(module.hot){
  module.hot.accept('./App', () => {
    setTimeout(render);
  })
}
render()
```

// structure



// folders

/src

App

Layout

App.jsx

Features

Event

EventDashboard

EventDashboard.jsx

EventList

EventList.jsx

EventListItem.jsx

EventListAttendee.jsx

nav

NavBar

NavBar.jsx

/public

assets

favicon.ico

index.html

manifest.json

// Semantic UI intro

Semantic UI React ∈ jQuery tree

↳ named import

```
import React, {Component} from 'react';
import {Button} from 'semantic-ui-react';
```

class App extends Component{

render(){

return(

<div>

<h1>Re-vents</h1>

<button className="ui icon button">
 <i className="smile icon"></i>

CSS Button

</button>

<Button icon="smile" content="React Button"/>

</div>

)

}

export default App;

|| App.jsx (app / layout)

NavBar is Stateful

```
import React, { Component, Fragment } from 'react';
import EventDashboard from '../.. / features / event / EventDashboard / EventDashboard';
↓ similar ones NavBar, {Container}
```

```
class App extends Component {
  render() {
    return (
      <Fragment>
        <NavBar />           ↗ className is needed
        <Container className="main">
          <EventDashboard />
        </Container>
      </Fragment>
    );
  }
}

export default App;
```

//EventDashboard.jsx (features / event / EventDashboard)

```
import React, {Component} from 'react'  
import {Grid} from 'semantic-ui-react'
```

```
class EventDashboard extends Component {
```

```
state = {
```

```
  events: eventsFromDashboard,  
  isOpen: false
```

```
}
```

```
handleIsOpenToggle = () => {
```

```
  this.setState({isOpen}) => {
```

```
    isOpen: !isOpen
```

```
}
```

can guarantee run sync

```
}
```

```
handleCreateEvent = (newEvent) => {
```

```
  newEvent.id = coid(); // collision resistant ids
```

```
  newEvent.hostPhotoURL = "/assets/user.png";
```

```
  this.setState({events}) => {// from previous state
```

```
    events: [...events, newEvent],
```

```
    isOpen: false
```

```
}
```

```
}
```

```
render() {
```

```
  const {events, isOpen} = this.state
```

```
  return (
```

```
    <Grid>
```

```
      <Grid.Column width={10}>
```

```
        <EventList events={events}/>
```

```
      </Grid.Column>
```

```
      <Grid.Column width={6}>
```

```
        <Button onClick={this.handleIsOpenToggle} positive content='Create Event' />
```

```
{isOpen &&
```

```
<EventForm
```

```
  createEvent = {this.handleCreateEvent}
```

```
  cancelFormOpen = {this.handleIsOpenToggle}
```

```
/>}
```

```
</Grid.Column>
```

```
</Grid>
```

```
)
```

```
export default EventDashboard;
```

Events [] → stateful

isOpen → Stateful

const eventsFromDashboard = [{...}, {...}]

* arrow function auto bind to the class it belongs

//this.setState({prevState}) => {

isOpen: !prevState.isOpen

```
}
```

ctrl + space to auto import or manual import

* ... → spread operator

Do not need "()", otherwise will open as soon as load up

⌘ + P = search and open the file

can be initials

// EventForm.jsx (features/event/EventForm)

```
import React, { Component } from 'react'
```

```
class EventForm extends Component {
```

```
state = {
```

```
  title: '',
```

```
  date: '',
```

```
  city: '',
```

```
  venue: '',
```

```
  hostedBy: ''
```

```
}
```

```
handleFormSubmit = evt => {
```

```
  evt.preventDefault();
```

```
  this.props.createEvent(this.state);
```

```
}
```

```
handleInputChange = ({target: {name, value}}) => {
```

```
  this.setState({
```

```
    [name]: value
```

```
  });
```

```
}
```

```
render() {
```

```
  const {cancelFromOpen} = this.props;
```

```
  const {title, date, city, venue, hostedBy} = this.state
```

```
  return (
```

```
    <Segment>
```

```
    <Form onSubmit={this.handleFormSubmit} autoComplete='off'>
```

```
      <Form.Field>
```

```
        <Label> Event Title </Label>
```

```
        <input name='title' onChange={this.handleInputChange} value={title} placeholder="Event Title" />
```

```
      </Form.Field>
```

```
      <Form.Field>
```

```
        <Label> Event Date </Label>
```

```
        <input
```

```
          name='date' onChange={this.handleInputChange} value={date}
```

```
          type='date' placeholder="Event Date"
```

```
        />
```

```
      </Form.Field>
```

```
      <Form.Field>
```

```
        <Label> City </Label>
```

```
        <input
```

```
          name='city' onChange={this.handleInputChange} value={city}
```

```
          placeholder="City event is taking place"
```

```
        />
```

```
      </Form.Field>
```

```
      <Form.Field>
```

```
        <Label> Venue </Label>
```

```
        <input
```

```
          name='venue' onChange={this.handleInputChange} value={venue}
```

Forms manage state → Stateful

4.2 E

// Uncontrolled forms (old)

controlled form

No direct access to the DOM

No access to the input

only concentrated with altering state

rely on React to manipulate the DOM

deconstruct

handleInputChange = (evt) => {

this.setState({

[evt.target.name]: evt.target.value

});

placeholder = "Enter the venue of the event"

/ >

</Form.Field>

<Form.Field>

<Label> Hosted By < Label

<input

name='hostedBy' onChange={this.handleInputChange} value={hostedBy}

placeholder = "Enter the name of person hosting"

/ >

</Form.Field>

<Button positive type='submit'>

Submit

</Button>

<Button onClick={cancelFromOpen} type='button'> Cancel </Button>

</Form>

</Segment>

)

}

export default EventForm;

// EventList.jsx (features/event/EventList) props passed down by parents → No state

```
import React, { Component, Fragment } from 'react'
```

```
class EventList extends Component {
```

```
  render() {
```

```
    return (
```

```
      <Fragment>
```

```
        {this.props.events.map(event => ( → ⚡ must have unique ids
```

```
          <EventListItem key={event.id} event={event}/>
```

```
        ))}
```

```
      ) </Fragment>
```

```
    )
```

```
  export default EventList;
```

// EventList.js
 (features/event/EventList) props passed down by parents → No state
 import React, { Component } from 'react'

```

class EventListItem extends Component {
  const { event } = this.props * get event out of this.props
  render() {
    return (
      <Segment.Group>
        <Segment>
          <Item.Group>
            <Item>
              <Item.Image size="tiny" circular src={event.hostPhotoURL} />
              <Item.Content>
                <Item.Header>{event.title}</Item.Header>
                <Item.Description>
                  Hosted by {event.hostedBy}
                </Item.Description>
                <Item.Content>
                  <Icon name="clock" /> {event.date}
                  <Icon name="marker" /> {event.venue}
                </Item.Content>
              </Item>
            </Item.Group>
          </Segment>
          <Segment>
            <Span>
              <Icon name="clock" /> {event.date}
              <Icon name="marker" /> {event.venue}
            </Span>
          </Segment>
          <Segment secondary> → only run the right side if the left side is true
            <List horizontal>
              {event.attendees && event.attendees.map((attendee) =>
                <EventListAttendee key={attendee.id} attendee={attendee} />
              ))}
            </List>
          </Segment>
          <Segment clearing>
            <Span>{event.description}</Span>
            <Button as="a" color="teal" floated="right" content="View" />
          </Segment>
        </Segment.Group>
    )
  }
}

export default EventListItem;
  
```

// EventListAttendee.jsx (features/event/EventList)

props passed down by parents → No state

```
import React, { Component } from 'react';
import { List, Image } from 'semantic-ui-react';

class EventListAttendee extends Component {
  render() {
    const { attendee } = this.props;
    return (
      <List.Item>
        <Image as='a' size='mini' circular src={attendee.photoURL} />
      </List.Item>
    );
  }
}

export default EventListAttendee;
```

//NavBar.jsx (features/nav/NavBar)

rely on user interaction \Rightarrow stateful

```
import React, {Component} from 'react'
```

```
class NavBar extends Component {
  render() {
    return (
      <Menu inverted fixed='top'>
        <Container>
          <Menu.Item header>
            <Img src='/assets/logo.png' alt='logo' />
            Re-vents
          </Menu.Item>
          <Menu.Item name='Events' />
          <Menu.Item>
            <Button floated='right' positive inverted content='Create Event' />
          </Menu.Item>
          <Menu.Item position='right'>
            <Button basic inverted content='Login' />
            <Button
              basic
              inverted
              content='Sign Out'
              style={{marginLeft: '0.5em'}}/>
          </Menu.Item>
        </Container>
      </Menu>
    )
  }
}

export default NavBar;
```

11 index.css (src)

```
body {  
    background-color: #rgb(234,234,234) !important;  
}  
  
.ui.menu .item img.logo {  
    margin-right: 1.5em;  
}  
  
.ui.fixed.menu {  
    background-image: linear-gradient(  
        135deg,  
        #rgb(24,42,115) 0%,  
        #rgb(33,138,174) 69%,  
        #rgb(32,167,172) 89%,  
    ) !important;  
}  
  
.ui.main.container,  
.main.segment {  
    margin-top: 7em;  
}
```