

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

$\nexists$  F1P  $\Rightarrow$  showAllCommands  $\Rightarrow$  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

$\diamond$  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 Structure

src

app

layout

App.css

App.jsx (renamed)

logo.svg

features

## React APIs

render()

constructor()

componentDidMount()

componentWillUnmount()

componentDidUpdate()

setState()

JSX

React Hooks

Raster VS. Vector

$\diamond$  install node version manager: <http://bit.ly/2Hn8HjG>

$\diamond$  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

$\diamond$  Create Read Update Delete

$\diamond$  extensions:

Auto Rename Tag

Bracket Pair Colorizer

Debugger for Chrome

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

change Localhost:8000 to 3000

"webRoot": "`{workspaceFolder}/src`"

VS Code ES7 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

$\diamond$  if not working  $\Rightarrow$  try: npm start

$\diamond$  React Developer tools  $\Rightarrow$  add to chrome

## // index.js

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

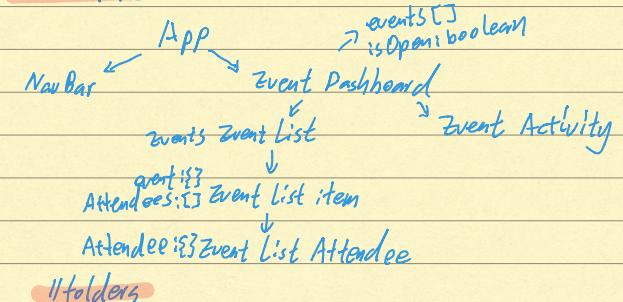
\* if it is passed in from a parent via props ∈ Not state

\* 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

#### /EventDetailed

#### EventDetailedPage.jsx

#### /EventList

#### EventList.jsx

#### EventListItem.jsx

#### EventListAttendee.jsx

#### /nav

#### /NavBar

#### NavBar.jsx

#### index.css

#### index.js

#### serviceWorker.js

## /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(){

<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;

# Layout

App

Home  
Page

Read Update Delete

Master / Detail view

updating and deleting events

Component re-use

React Lifecycle Events

ComponentDidMount

ComponentWillReceiveProps

event  
Dashboard

Event  
Detailed

EventForm

people  
Dashboard

user  
Detailed

settings  
Dashboard

Event  
List

settings  
Nav

EventList  
Item

Account  
Page

About  
Page

Basics  
Page

Photos  
Page

EventList  
Attendee

## Stateless Functional Components

simple components

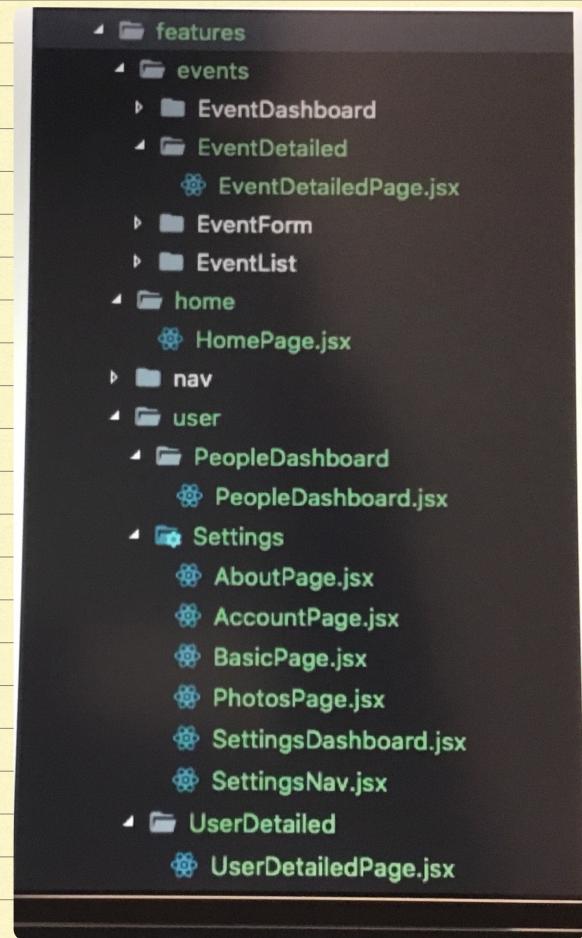
Do not have state\*

Do not have access to React Lifecycle methods\*

Offer no performance improvements over class components

Designed to simplify the code

\*React hooks changes this



## || App.jsx (app / layout)

NavBar is Stateful

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

→ auto generated after type Event Dashboard

```
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)

Events [] → stateful

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

isOpen → Stateful

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

class EventDashboard extends Component {

state = {

events: eventsFromDashboard,

isOpen: false,

selectedEvent: null

}

\* arrow function auto bind to the class it belongs

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

isOpen: !prevState.isOpen

}

handleCreateFormOpen = () => {

this.setState({

isOpen: true,

selectedEvent: null

)

}

//handleIsOpenToggle = () => {

this.setState({isOpen}) => {

isOpen: !isOpen

)

can guarantee run sync

handleFormCancel = () => {

this.setState({

isOpen: false

)

}

handleCreateEvent = (newEvent) => {

newEvent.id = cuid(); collision resistant ids

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

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

events: [...events, newEvent],

isOpen: false

)

}

ctrl + space to auto import or manual import

\* ... → spread operator

handleSelectEvent = (event) => {

this.setState({

selectedEvent: event,

isOpen: true

)

}

handleUpdateEvent = (updateEvent) => {

this.setState({events}) => ({

events: events.map(event => {

if (event.id === updateEvent.id) {

return {...event, ...updateEvent};

} else {

return event;

)

),

isOpen: false,

selectedEvent: null

)

}

```
handleDeleteEvent = (id) => {
  this.setState({events}) => {
    events: events.filter(e => e.id !== id)
  })
}
```

render() {

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

```
return
```

```
<Grid>
```

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

```
<EventList
```

```
events = {events}
```

```
selectEvent = {this.handleSelectEvent}
```

```
deleteEvent = {this.handleDeleteEvent}
```

```
/>
```

⌘ + P = search and open the file

can be initials

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

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

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

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

```
{isOpen &&
```

```
<EventForm
```

```
key = {selectedEvent ? selectedEvent.id : 0}
```

```
updateEvent = {this.handleUpdateEvent}
```

```
selectedEvent = {selectedEvent}
```

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

```
cancelFormOpen = {this.handleFormCancel}
```

```
/>}
```

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

```
</Grid>
```

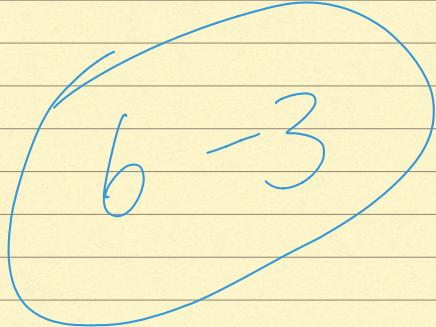
```
)
```

```
}
```

```
export default EventDashboard;
```

//EventDetailedPage.jsx (features / event / EventDetailed)

rfce  
rate & arrow function & preferred



// EventForm.jsx (features/event/EventForm)  
import React, { Component } from 'react'

Forms manage state → Stateful

4.2 E  
II Uncontrolled forms (old)

controlled form

No direct access to the DOM  
so  
No access to the input

only concentrated with altering state  
rely on React to manipulate the DOM

```
class EventForm extends Component {  
  state = {  
    title: '',  
    date: '',  
    city: '',  
    venue: '',  
    hostedBy: ''  
}
```

3;

```
componentDidMount() {  
  if (this.props.selectedEvent !== null) {  
    this.setState({  
      ...this.props.selectedEvent  
    })  
  }  
}
```

3

```
handleFormSubmit = evt => {  
  evt.preventDefault();  
  if (this.state.id) {  
    this.props.updateEvent(this.state);  
  } else {  
    this.props.createEvent(this.state);  
  }  
}
```

3;

```
handleInputChange = ({ target: { name, value } }) => {  
  this.setState({  
    [name]: value  
  });  
}
```

←  
deconstruct

```
handleInputChange = (evt) => {  
  this.setState({  
    [evt.target.name]: evt.target.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}
            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() {
    const { events, selectEvent, deleteEvent } = this.props
    return (
      <Fragment>
        {events.map(event =>
          <EventList Item
            key={event.id} // must have unique ids
            event={event}
            selectEvent={selectEvent}
            deleteEvent={deleteEvent}
          />
        )}
      </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, selectEvent, deleteEvent } = 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 onClick={() => deleteEvent(event.id)}> as="a" color="red" floated="right" content="Delete" />
            <Button onClick={() => selectEvent(event)}> as="a" color="teal" floated="right" content="View" />
          </Segment> ^ arrow function prevents the selectEvent run immediately
        </Segment.Group>
      )
    )
}

```

export default EventList;

// 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;  
}
```