

Web application development

(Introduction to Basic React)

Instructor: Tran Vinh Khiem

September 1st, 2022



Smart Software System Team

A photograph of a railway track receding into the distance under a cloudy sky.

*“We love what we do and we do what our clients
love & work with great clients all over the world
to create thoughtful and purposeful websites.”*

— ProWeb365



Basic React – Exercise 1 – Use Effect

- The useEffect() Hook is used to execute "side effects" inside a component. Another way to think about side-effect code is that functional components have just one duty — delivering JSX content to display. If the component requires additional functionality, such as retrieving API data, this should be performed in a useEffect() Hook.
- Access to this link and do exercise:
<https://docs.google.com/document/d/1hovSiruqRDxactW-glb9v-geOCNzXS3ySyWgta5zk80/>



Basic React – Exercise 1.1 – Fetching data

```
import * as React from "react";

function fetchUser() {
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve({ id: 1, name: "Adam" });
    }, 1000);
  );
}

function App() {
  const [id, setId] = React.useState("loading...");
  const [name, setName] = React.useState("loading...");

  React.useEffect(() => {
    fetchUser().then((user) => {
      setId(user.id);
      setName(user.name);
    });
  );
}

  return (
    <>
      <p>ID: {id}</p>
      <p>Name: {name}</p>
    </>
  );
}

export default App;
```



Basic React – Exercise 1.2 – Create spinner fallbacks

- The Suspense component's simplest fallback is text that alerts the user that something is occurring. The fallback attribute may be any valid React element, which allows us to make the fallback more aesthetically pleasing. For instance, the react-spinners package has a variety of spinner components, all of which are compatible with Suspense.



Basic React – Exercise 1.2 – Create spinner fallbacks

```
1 ✓ import * as React from "react";
2   import { FadeLoader } from "react-spinners";
3   import MyPage from "./MyPage";
4
5 ✓ function App() {
6   return (
7     <React.Suspense fallback={<FadeLoader color="lightblue" size={150} />}>
8       <MyPage />
9     </React.Suspense>
10  );
11}
12
13 export default App;
14
```

```
1 import React from "react";
2
3 export default function MyFeature() {
4   return <p>My Feature</p>;
5 }
6
```

```
1 import * as React from "react";
2
3 const MyFeature = React.lazy(() =>
4   Promise.all([
5     import("./MyFeature"),
6     new Promise((resolve) => {
7       setTimeout(() => {
8         resolve();
9       }, 3000);
10    }),
11  ]).then(([m]) => m)
12);
13
14 function MyPage() {
15   return (
16     <>
17       <h1>My Page</h1>
18       <MyFeature />
19     </>
20   );
21 }
22
23 export default MyPage;
```



Basic React – Exercise 1.3 – React Route

Access to this link and do exercises:

<https://docs.google.com/document/d/1QPzOAOLPqlI4pOeeJH8gdBnRmwmfFDOy8A3UIUhZ1c>



Basic React – Exercise 1.3 – Navigation components

```
import "typeface-roboto";
import React, { useState } from "react";
import Drawer from "@mui/material/Drawer";
import Button from "@mui/material/Button";
import List from "@mui/material/List";
import ListItem from "@mui/material/ListItem";
import ListItemText from "@mui/material/ListItemText";
import { BrowserRouter as Router, Route, Switch, Link } from "react-router-dom";
import First from "./First";
import Second from "./Second";
import Third from "./Third";

export default function App({ links }) {
  const [open, setOpen] = useState(false);

  function toggleDrawer({ type, key: any }) {
    if (type === "keydown" && (key === "Tab" || key === "Shift")) {
      return;
    }

    setOpen(!open);
  }

  return (
    <Router>
      <Button onClick={toggleDrawer}>Open Nav</Button>
      <section>
        <Route path="/first" component={First} />
        <Route path="/second" component={Second} />
        <Route path="/third" component={Third} />
      </section>
      <Drawer open={open} onClose={toggleDrawer}>
```

```
<Drawer open={open} onClose={toggleDrawer}>
  <div
    style={{ width: 250 }}
    role="presentation"
    onClick={toggleDrawer}
    onKeydown={toggleDrawer}
  >
    <List>
      {links.map((link) => (
        <ListItem button key={link.url} component={Link} to={link.url}>
          <Switch>
            <Route
              exact
              path={link.url}
              render={() => (
                <ListItemText
                  primary={link.name}
                  primaryTypographyProps={{ color: "primary" }}
                />
              )}
            />
            <Route
              path="/"
              render={() => <ListItemText primary={link.name} />}
            />
          </Switch>
        </ListItem>
      ))}
    </List>
  </div>
</Drawer>
```



Basic React – Exercise 1.3 – Navigation components

```
        </Drawer>
      </Router>
    );
}

App.defaultProps = {
  links: [
    { url: "/first", name: "First Page" },
    { url: "/second", name: "Second Page" },
    { url: "/third", name: "Third Page" },
  ],
};
```



Basic React – Exercise 1.3.1 – Navigation with tabs

```
import "typeface-roboto";
import React, { useState } from "react";
import Drawer from "@mui/material/Drawer";
import Button from "@mui/material/Button";
import List from "@mui/material/List";
import ListItem from "@mui/material/ListItem";
import ListItemText from "@mui/materialListItemText";
import { BrowserRouter as Router, Route, Switch, Link } from "react-router-dom";
import First from "./First";
import Second from "./Second";
import Third from "./Third";

export default function App({ links }) {
  const [open, setOpen] = useState(false);

  function toggleDrawer({ type, key: any }) {
    if (type === "keydown" && (key === "Tab" || key === "Shift")) {
      return;
    }

    setOpen(!open);
  }

  return (
    <Router>
      <Button onClick={toggleDrawer}>Open Nav</Button>
      <section>
        <Route path="/first" component={First} />
        <Route path="/second" component={Second} />
        <Route path="/third" component={Third} />
      </section>
      <Drawer open={open} onClose={toggleDrawer}>
```

```
<Drawer open={open} onClose={toggleDrawer}>
  <div
    style={{ width: 250 }}
    role="presentation"
    onClick={toggleDrawer}
    onKeydown={toggleDrawer}
  >
    <List>
      {links.map((link) => (
        <ListItem button key={link.url} component={Link} to={link.url}>
          <Switch>
            <Route
              exact
              path={link.url}
              render={() => (
                <ListItemText
                  primary={link.name}
                  primaryTypographyProps={{ color: "primary" }}
                />
              )}
            />
            <Route
              path="/"
              render={() => <ListItemText primary={link.name} />}
            />
          </Switch>
        </ListItem>
      ))}
    </List>
  </div>
</Drawer>
```



Basic React – Exercise 1.3.1 – Navigation with tabs

```
import "typeface-roboto";
import React from "react";
import { BrowserRouter as Router, Route, Link } from "react-router-dom";
import AppBar from "@mui/material/AppBar";
import Tabs from "@mui/material/Tabs";
import Tab from "@mui/material/Tab";
import Typography from "@mui/material/Typography";

const tabContentStyle = {
  padding: 16,
};

function TabContainer({ value }) {
  return (
    <AppBar position="static">
      <Tabs value={value}>
        <Tab label="Item One" component={Link} to="/" />
        <Tab label="Item Two" component={Link} to="/page2" />
        <Tab label="Item Three" component={Link} to="/page3" />
      </Tabs>
    </AppBar>
  );
}

export default function App() {
  return (
    <Router>
      <Route
        exact
        path=""
        render={() => (
          </Router>
        )}
      >
        <TabContainer value={0} />
        <Typography component="div" style={tabContentStyle}>
          Item One
        </Typography>
      </>
    </>
  );
}
```

```
<>
  <TabContainer value={0} />
  <Typography component="div" style={tabContentStyle}>
    Item One
  </Typography>
</>
</>
<Route
  exact
  path="/page2"
  render={() => (
    <>
      <TabContainer value={1} />
      <Typography component="div" style={tabContentStyle}>
        Item Two
      </Typography>
    </>
  </>
)>
<Route
  exact
  path="/page3"
  render={() => (
    <>
      <TabContainer value={2} />
      <Typography component="div" style={tabContentStyle}>
        Item Three
      </Typography>
    </>
  </>
)>
</>
```

Basic React – Exercise 1.3.2 – More navigations



- Create responsive router as below.

Kip Russel
Harrison Swift
Carter Heaney
Evert Conroy
Hoyt Kautzer
Miles Kerluke
Tiara Stoltenberg
Dovie Terry
Madelynn Berge

Carter Heaney

ID: 28977789

Address: 1095 Johnny Ridge
Lynchstad
Iowa
81452-3853

Department: Clothing

Basic React – Exercise 1.3.2 – More navigations

```
import { BrowserRouter, Link, Route, Switch } from 'react-router-dom'
import PeopleContainer from './PeopleContainer'

function App() {
  return (
    <BrowserRouter>
      <Switch>
        <Route path="/people">
          <PeopleContainer />
        </Route>
        <Link to="/people">People</Link>
      </Switch>
    </BrowserRouter>
  )
}

export default App
```

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'
import App from './App'
import reportWebVitals from './reportWebVitals'

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
)

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-v2
reportWebVitals()
```

Basic React – Exercise 1.3.2 – More navigations

```
import { useParams } from 'react-router-dom'
import people from './people'
import './Person.css'

const Person = () => {
  const { id } = useParams()
  const person = people.filter((p) => p.id === id)[0]

  return (
    <main className="Person">
      <h1>{person.name}</h1>
      <label>ID:</label>
      <div className="Person-id">{id}</div>
      <label>Address:</label>
      <div className="Person-address">{person.address}</div>
      <label>Department:</label>
      <div className="Person-department">{person.department}</div>
    </main>
  )
}

export default Person
```

```
import Media from 'react-media'
import { Redirect, Route, Switch } from 'react-router-dom'
import Person from './Person'
import PeopleList from './PeopleList'
import people from './people'

const PeopleContainer = () => {
  return (
    <Media
      queries={{
        small: '(max-width: 700px)',
      }}
    >
      {(size) =>
        size.small ? (
          <Switch>
            <Route path="/people/:id">
              <Person />
            </Route>
            <PeopleList />
          </Switch>
        ) : (
          <div style={{ display: 'flex' }}>
            <PeopleList />
            <Switch>
              <Route path="/people/:id">
                <Person />
              </Route>
              <Redirect to={`/people/${people[0].id}`} />
            </Switch>
          </div>
        )
      }
    </Media>
  )
}
```

Basic React – Exercise 1.3.2 – More navigations

```
import { NavLink } from 'react-router-dom'
import people from './people'
import './PeopleList.css'

const PeopleList = () => {
  return (
    <nav className="PeopleList">
      <ul>
        {people.map((person) => (
          <li key={`person-${person.id}`}>
            <NavLink
              activeClassName="currentPerson"
              to={`/people/${person.id}`}
            >
              {person.name}
            </NavLink>
          </li>
        ))}
      </ul>
    </nav>
  )
}

export default PeopleList
```

```
import { useParams } from 'react-router-dom'
import people from './people'
import './Person.css'

const Person = () => {
  const { id } = useParams()
  const person = people.filter((p) => p.id === id)[0]

  return (
    <main className="Person">
      <h1>{person.name}</h1>
      <label>ID:</label>
      <div className="Person-id">{id}</div>
      <label>Address:</label>
      <div className="Person-address">{person.address}</div>
      <label>Department:</label>
      <div className="Person-department">{person.department}</div>
    </main>
  )
}

export default Person
```

Basic React – Exercise 1.3.2 – More navigations



```
[  
  {  
    "id": "58826508",  
    "name": "Kip Russel",  
    "address": "18627 Sporer Mews\nMaximechester\nSouth Dakota\n04691",  
    "department": "Movies"  
  },  
  {  
    "id": "34033353",  
    "name": "Harrison Swift",  
    "address": "910 Lueilwitz Lake\nLake Troy\nWisconsin\n25072",  
    "department": "Games"  
  },  
  {  
    "id": "28977789",  
    "name": "Carter Heaney",  
    "address": "1095 Johnny Ridge\nLynchstad\nIowa\n81452-3853",  
    "department": "Clothing"  
  },  
  {  
    "id": "62734836",  
    "name": "Evert Conroy",  
    "address": "812 Dario Drive\nMurrayville\nIllinois\n51582",  
    "department": "Games"  
  },  
]
```

It's your turn, please add css file for app.js, index.js, PeopleList and Person.



Basic React – Exercise 1.3.2 – React props

Access to this link and do exercises:

https://docs.google.com/document/d/1ggWY8zkiDPa_cDzGKIrk07ZCEp-VocON9mcw_Okresw



Basic React – Exercise 1.4 – Text input

```
import "typeface-roboto";
import React from "react";
import FormGroup from "@mui/material/FormGroup";
import MyTextInput from "./MyTextInput";
import MySelect from "./MySelect";

export default function App() {
  return (
    <FormGroup style={{ width: 200, margin: 10 }}>
      <MyTextInput />
      <MySelect />
    </FormGroup>
  );
}
```

```
import React, { useState } from "react";
import TextField from "@mui/material/TextField";

export default function MyTextInput() {
  const [value, setValue] = useState("");

  return (
    <TextField
      label="Name"
      value={value}
      onChange={(e) => setValue(e.target.value)}
      margin="normal"
    />
  );
}
```

```
import React, { useState } from "react";
importInputLabel from "@mui/material/InputLabel";
import MenuItem from "@mui/material/MenuItem";
import FormControl from "@mui/material/FormControl";
import Select from "@mui/material/Select";

export default function MySelect() {
  const [value, setValue] = useState("first");

  return (
    <FormControl>
      <InputLabel htmlFor="my-select">My Select</InputLabel>
      <Select
        value={value}
        onChange={(e) => setValue(e.target.value)}
        inputProps={{ id: "my-select" }}
      >
        <MenuItem value="first">First</MenuItem>
        <MenuItem value="second">Second</MenuItem>
        <MenuItem value="third">Third</MenuItem>
      </Select>
    </FormControl>
  );
}
```



Basic React – Exercise 1.5 – Button

```
import "typeface-roboto";
import React, { useState } from "react";
import Button from "@mui/material/Button";
import Grid from "@mui/material/Grid";
import IconButton from "@mui/material/IconButton";
import AndroidIcon from "@mui/icons-material/Android";

const buttonStyle = { margin: 10 };

function toggleColor(setter, value) {
  setter(value === "default" ? "primary" : "default");
}

export default function App() {
  const [contained, setContained] = useState("default");
  const [text, setText] = useState("default");
  const [outlined, setOutlined] = useState("default");
  const [icon, setIcon] = useState("default");

  return (
    <Grid container>
      <Grid item>
        <Button
          variant="contained"
          style={buttonStyle}
          color={contained}
          onClick={() => toggleColor(setContained, contained)}
        >
          Contained
        </Button>
      </Grid>
      <Grid item>
        <Button
          component={Button}
          style={buttonStyle}
          color={text}
          onClick={() => toggleColor(setText, text)}
        >
          Text
        </Button>
      </Grid>
      <Grid item>
        <Button
          variant="outlined"
          style={buttonStyle}
          color={outlined}
          onClick={() => toggleColor(setOutlined, outlined)}
        >
          Outlined
        </Button>
      </Grid>
      <Grid item>
        <IconButton
          style={buttonStyle}
          color={icon}
          onClick={() => toggleColor(setIcon, icon)}
        >
          <AndroidIcon />
        </IconButton>
      </Grid>
    </Grid>
  );
}
```



Basic React – Exercise 1.6 – Validate form

Multiple fields

Address 1:

Too short!

Address 2:

Required

Address 3:

Required

Address 4:

Required

Price:

Must be at least 102

Required By:

Required

Submit!

Current value:

```
{  
  "address1": "AB",  
  "price": "101"  
}
```

Valid?

false

Errors?

```
{  
  "address1": "Too short!",  
  "address2": "Required",  
  "address3": "Required",  
  "address4": "Required",  
  "price": "Must be at least 102",  
  "requiredBy": "Required"  
}
```



Basic React – Exercise 1.6 – Validate form

```
import { useState } from 'react'
import './App.css'
import FormExample0 from './FormExample0'
import FormExample1 from './FormExample1'
import ShowData from './ShowData'

const onSubmit = (v) =>
  alert(`Submit value: ${JSON.stringify(v, null, 2)}`)

function App() {
  const [formFields, setFormFields] = useState({})
  const [errors, setErrors] = useState({})
  const [valid, setValid] = useState()
  const [firstForm, setFirstForm] = useState(true)

  return (
    <div className="App">
      <nav>
        <select
          onChange={(evt) =>
            setFirstForm(evt.target.value === 'first')}
        >
          <option value="first">Single field</option>
          <option value="second">Multiple fields</option>
        </select>
      </nav>
      <main>
        {firstForm ? (
          <FormExample0
            onChange={(ff, v, e) => {
              setFormFields(ff)
              setValid(v)
              setErrors(e)
            }}
          >
        ) : (
          <FormExample1
            onChange={(ff, v, e) => {
              setFormFields(ff)
              setValid(v)
              setErrors(e)
            }}
            onSubmit={onSubmit}
            initialValue={{
              field1: 'Some stuff',
            }}
          >
        )}
      </main>
    </div>
  )
}

export default App
```

```
  >>>
  >>> onSubmit={onSubmit}
  initialValue={{
    field1: 'Some stuff',
  }}
  />
) : (
<FormExample1
  onChange={(ff, v, e) => {
    setFormFields(ff)
    setValid(v)
    setErrors(e)
  }}
  onSubmit={onSubmit}
  initialValue={{
    address1: '1 Main Street',
  }}
  />
)

<ShowData
  formFields={formFields}
  errors={errors}
  valid={valid}
  />
</main>
</div>
}

export default App
```



Basic React – Exercise 1.6 – Validate form

```
import { useEffect, useState } from 'react'
import './App.css'
import SimpleForm from './SimpleForm'
import InputField from './InputField'

const FormExample0 = ({ onSubmit, onChange, initialValue = {} }) => {
  const [formFields, setFormFields] = useState(initialValue)

  const [valid, setValid] = useState(true)
  const [errors, setErrors] = useState({})

  useEffect(() => {
    if (onChange) {
      onChange(formFields, valid, errors)
    }
  }, [onChange, formFields, valid, errors])

  return (
    <div className="TheForm">
      <h1>Single field</h1>

      <SimpleForm
        value={formFields}
        onChange={setFormFields}
        onValid={(v, errs) => {
          setValid(v)
          setErrors(errs)
        }}
      >
        <InputField
          name="field1"
          onValidate={(v) =>
            !v || v.length < 3 ? 'Too short!' : null
          }
        />
      
```

```
      <button
        onClick={() => onSubmit && onSubmit(formFields)}
        disabled={!valid}
      >
        Submit!
      </button>
    </SimpleForm>
  </div>
}

export default FormExample0
```



Basic React – Exercise 1.6 – Validate form

```
> import { useEffect, useState } from 'react'
import './App.css'
import SimpleForm from './SimpleForm'
import InputField from './InputField'

const FormExample1 = ({ onSubmit, onChange, initialValue = {} }) => {
  const [formFields, setFormFields] = useState(initialValue)

  const [valid, setValid] = useState(true)
  const [errors, setErrors] = useState({})

  useEffect(() => {
    if (onChange) {
      onChange(formFields, valid, errors)
    }
  }, [onChange, formFields, valid, errors])

  return (
    <div className="TheForm">
      <h1>Multiple fields</h1>

      <SimpleForm
        value={formFields}
        onChange={setFormFields}
        onValid={(v, errs) => {
          setValid(v)
          setErrors(errs)
        }}
      >
        <InputField
          name="address1"
          onValidate={(v) =>
            !v || v.length < 3 ? 'Too short!' : null
          }
        />
      </SimpleForm>
    </div>
  )
}
```

```
<InputField
  name="address2"
  onValidate={(v) => (v ? null : 'Required')}
/>

<InputField
  name="price"
  type="number"
  onValidate={(v) =>
    !v || parseInt(v) < 102 ? 'Must be at least 102' : null
  }
/>

<InputField
  name="requiredBy"
  type="date"
  onValidate={(v) => (v ? null : 'Required')}
/>

<button
  onClick={() => onSubmit && onSubmit(formFields)}
  disabled={!valid}
>
  Submit!
</button>
</SimpleForm>
</div>

}

export default FormExample1
```



Basic React – Exercise 1.6 – Validate form

```
import { useCallback, useEffect, useState } from 'react'
import FormContext from './FormContext'
import './SimpleForm.css'

const SimpleForm = ({ children, value, onChange, onValid }) => {
  const [values, setValues] = useState(value || {})
  const [dirtyFields, setDirtyFields] = useState({})
  const [invalidFields, setInvalidFields] = useState({})

  useEffect(() => {
    setValues(value || {})
  }, [value])

  useEffect(() => {
    if (onChange) {
      onChange(values)
    }
  }, [onChange, values])

  useEffect(() => {
    if (onValid) {
      onValid(
        Object.keys(invalidFields).every((i) => !invalidFields[i]),
        invalidFields
      )
    }
  }, [onValid, invalidFields])

  const setValue = useCallback(
    (field, v) => setValues(({ ...vs, [field]: v })),
    [setValues]
  )
  const getValue = useCallback((field) => values[field], [values])
  const setDirty = useCallback(
    (field) => setDirtyFields({ ...df, [field]: true }),
    [setDirtyFields]
  )
}
```

```
const getDirty = useCallback(
  (field) => Object.keys(dirtyFields).includes(field),
  [dirtyFields]
)
const setInvalid = useCallback(
  (field, error) => {
    setInvalidFields((i) => ({
      ...i,
      [field]: error ? error : undefined,
    }))
  },
  [setInvalidFields]
)
const form = {
  setValue: setValue,
  value: getValue,
  const setDirty: (field: any) => void
  setDirty: setDirty,
  isDirty: getDirty,

  setInvalid: setInvalid,
}

return (
  <div className="SimpleForm-container">
    <FormContext.Provider value={form}>
      {children}
    </FormContext.Provider>
  </div>
)
}

export default SimpleForm
```



Basic React – Exercise 1.6 – Validate form

```
// import { useContext, useEffect, useState } from 'react'
import FormContext from './FormContext'

import './InputField.css'

// const splitCamelCase = (s) =>
//   s
//     .replace(/([a-z0-9])([A-Z0-9])/g, '$1 $2')
//     .replace(/\w([a-z])/g, (x) => x.toUpperCase())

// const InputField = (props) => {
//   const form = useContext(FormContext)

//   const [error, setError] = useState('')

//   const { onValidate, name, label, ...otherProps } = props

//   let value = form.value && form.value(name)

//   useEffect(() => {
//     if (onValidate) {
//       setError(onValidate(value))
//     }
//   }, [onValidate, value])

//   const setInvalid = form.setInvalid

//   useEffect(() => {
//     if (setInvalid) {
//       setInvalid(name, error)
//     }
//   }, [setInvalid, name, error])

//   if (!form.value) {
//     return 'InputField should be wrapped in a form'
//   }

//   return (
//     <div className="InputField">
//       <label htmlFor={name}>{label || splitCamelCase(name)}:</label>
//       <input
//         id={name}
//         onBlur={() => form.setDirty(name)}
//         value={value || ''}
//         onChange={(event) => {
//           form.setDirty(name)
//           form.setValue(name, event.target.value)
//         }}
//         {...otherProps}
//       />
//       {
//         <div className="InputField-error">
//           {form.isDirty(name) && error ? error : <br/>}
//         </div>
//       }
//     </div>
//   )
// }

// export default InputField
```

```
return (
  <div className="InputField">
    <label htmlFor={name}>{label || splitCamelCase(name)}:</label>
    <input
      id={name}
      onBlur={() => form.setDirty(name)}
      value={value || ''}
      onChange={(event) => {
        form.setDirty(name)
        form.setValue(name, event.target.value)
      }}
      {...otherProps}
    />
    {
      <div className="InputField-error">
        {form.isDirty(name) && error ? error : <br/>}
      </div>
    }
  </div>
)

export default InputField
```



Basic React – Exercise 1.6 – Validate form

```
import { createContext } from 'react'

const FormContext = createContext({})

export default FormContext
```

```
import './ShowData.css'

const ShowData = ({ formFields, valid, errors }) => (
  <div className="ShowData">
    <dl>
      <dt>Current value:</dt>
      <dd>{JSON.stringify(formFields, null, 2)}</dd>
      <dt>Valid?</dt>
      <dd>{JSON.stringify(valid)}</dd>
      <dt>Errors?</dt>
      <dd>{JSON.stringify(errors, null, 2)}</dd>
    </dl>
  </div>
)

export default ShowData
```



Basic React – Exercise 1.7 – API

Access to this link and do exercises:

<https://docs.google.com/document/d/1o3y65xOtyWJGaY5-oRCH9OgBMSESyFKp9c-sjVJ2O0o>

Basic React – Exercise 1.7.1 – Connect to simple BE



```
import './App.css'
import { useState } from 'react'
import useMessages from './useMessages'

function App() {
  const [forum, setForum] = useState('nasa')
  const {
    data: messages,
    loading: messagesLoading,
    error: messagesError,
  } = useMessages(forum)

  return (
    <div className="App">
      <button onClick={() => setForum('nasa')}>NASA</button>
      <button onClick={() => setForum('notNasa')}>Not NASA</button>
      {messagesError ? (
        <div className="error">
          Something went wrong:
          <div className="error-contents">
            {messagesError.message}
          </div>
        </div>
      ) : messagesLoading ? (
        <div className="loading">Loading...</div>
      ) : messages && messages.length ? (
        <dl>
          {messages.map((m) => (
            <>
              <dt>{m.author}</dt>
              <dd>{m.text}</dd>
            </>
          ))}
        </dl>
      ) : (
        'No messages'
      )
    </div>
  )
}
```

```
import { useEffect, useState } from 'react'

const useMessages = (forum) => {
  const [data, setData] = useState([])
  const [loading, setLoading] = useState(false)
  const [error, setError] = useState()

  useEffect(() => {
    let didCancel = false
    setError(null)
    if (forum) {
      ;(async () => {
        try {
          setLoading(true)
          const response = await fetch(`/messages/${forum}`)
          if (!response.ok) {
            const text = await response.text()
            throw new Error(
              `Unable to read messages for ${forum}: ${text}`
            )
          }
          const body = await response.json()
          if (!didCancel) {
            setData(body)
          }
        } catch (err) {
          setError(err)
        } finally {
          setLoading(false)
        }
      })()
    } else {
      setData([])
      setLoading(false)
    }
  }, [forum])
}

export default useMessages
```

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'
import App from './App'
import reportWebVitals from './reportWebVitals'

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
)
```



Basic React – Exercise 1.7.1 – Connect to services

```
const express = require('express')
const app = express()

app.use(express.json())

const messages = [
  {
    author: 'SC',
    text: 'Rolls complete and a pitch is program. One BRAVO.',
  },
  {
    author: 'PAO',
    text: 'One BRAVO is an abort control model. Altitude is 2 miles.',
  },
  {
    author: 'CAPCOM',
    text: 'All is well at Houston. You are good at 1 minute.',
  },
]

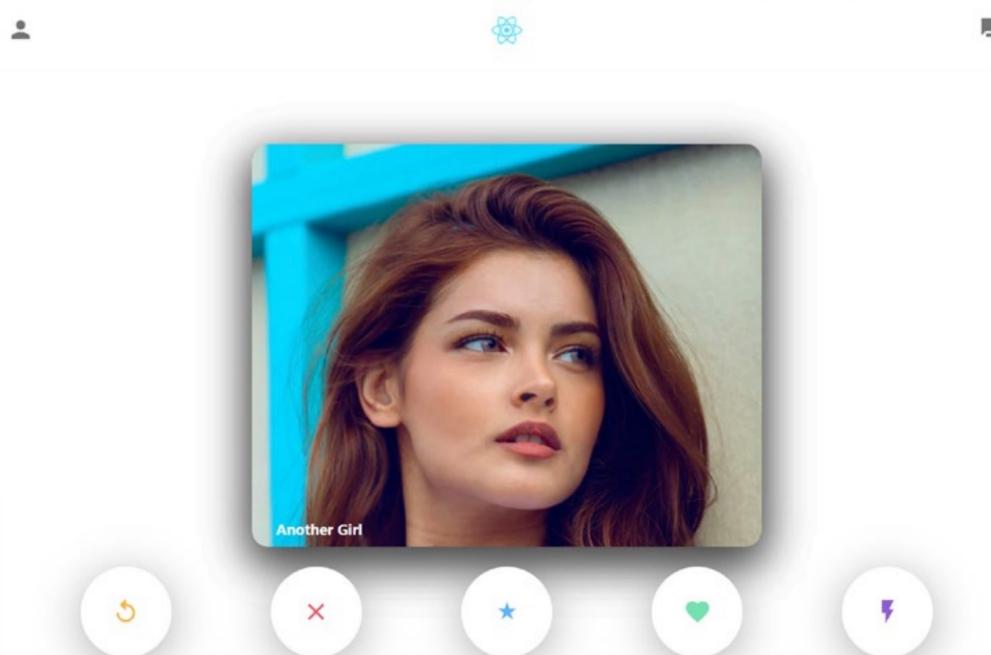
app.get('/messages/:forum', (request, response) => {
  if (request.params && request.params.forum === 'nasa') {
    return response.send(messages)
  }
  return response.status(404).send({ error: 'Unknown forum' })
})

app.post('/messages/:forum', (request, response) => {
  if (request.params && request.params.forum === 'nasa') {
    messages.push(request.body)
    return response.send('OK')
  }
  return response.status(404).send({ error: 'Unknown forum' })
})

app.listen(5000, () => console.log('⚡ Launched on port 5000!'))
```



Basic React – Exercise 2 – Create tinder "fake" app





Basic React – Exercise 2 – Create tinder "phake" app – Front End – app.js and Header

```
import './App.css';
import DatingCards from './components/DatingCards';
import Header from './components/Header';
import SwipeButtons from './components/SwipeButtons';

function App() {
  return (
    <div className="app">
      <Header />
      <DatingCards />
      <SwipeButtons />
    </div>
  );
}

export default App;
```

```
import React from 'react'
import './Header.css'
import PersonIcon from '@material-ui/icons/Person'
import IconButton from '@material-ui/core/IconButton'
import ForumIcon from '@material-ui/icons/Forum'

const Header = () => [
  return (
    <div className="header">
      <IconButton>
        <PersonIcon fontSize="large" className="header__icon" />
      </IconButton>
      
      <IconButton>
        <import IconButton size="large" className="header__icon" />
      </IconButton>
    </div>
  )
]

export default Header
```



Basic React – Exercise 2 – Create tinder "phake" app – Front End - DatingCard

```
import React, { useState, useEffect } from 'react'
import DatingCard from 'react-tinder-card'
import './DatingCards.css'
import axios from './axios'

const DatingCards = () => {
  const [people, setPeople] = useState([])

  useEffect(() => {
    async function fetchData() {
      const req = await axios.get("/dating/cards")
      setPeople(req.data)
    }
    fetchData()
  }, [])

  const swiped = (direction, nameToDelete) => {
    console.log("receiving " + nameToDelete)
  }

  const outOfFrame = (name) => {
    console.log(name + " left the screen!!")
  }

  return (
    <div className="datingCards">
      <div className="datingCards__container">
        {people.map((person) => (
          <DatingCard
            className="swipe"
            key={person.name}
            preventSwipe={[ 'up', 'down' ]}
            onSwipe={(dir) => swiped(dir, person.name)}
            onCardLeftScreen={() => outOfFrame(person.name)}
          >
            <div style={{ backgroundImage: `url(${person.imgUrl})` }} className=
              <h3>{person.name}</h3>
            </div>
        ))
      )
    </div>
  )
}

export default DatingCards
```

```
</DatingCard>
  )})
</div>
</div>
}

export default DatingCards
```



Basic React – Exercise 2 – Create tinder "phake" app – Front End – Swipe Button

```
import React from 'react'
import './SwipeButtons.css'
import ReplayIcon from '@material-ui/icons/Replay'
import CloseIcon from '@material-ui/icons/Close'
import StarRateIcon from '@material-ui/icons/StarRate'
import FavoriteIcon from '@material-ui/icons/Favorite'
import FlashOnIcon from '@material-ui/icons/FlashOn'
import IconButton from '@material-ui/core/IconButton'

const SwipeButtons = () => {
  return (
    <div className="swipeButtons">
      <IconButton className="swipeButtons__repeat">
        <ReplayIcon fontSize="large" />
      </IconButton>
      <IconButton className="swipeButtons__left">
        <CloseIcon fontSize="large" />
      </IconButton>
      <IconButton className="swipeButtons__star">
        <StarRateIcon fontSize="large" />
      </IconButton>
      <IconButton className="swipeButtons__right">
        <FavoriteIcon fontSize="large" />
      </IconButton>
      <IconButton className="swipeButtons__lightning">
        <FlashOnIcon fontSize="large" />
      </IconButton>
    </div>
  )
}

export default SwipeButtons
```



Basic React – Exercise 2 – Create tinder "phake" app – Front End – Connect to BE

```
import axios from 'axios'

const instance = axios.create({
  baseURL: "YourBackendURL"
})

export default instance
```



Basic React – Exercise 2 – Create tinder "phase" app – Back end

- Set up firebase
- Set up mongoDB
 - npm i express mongoose
 - npm i nodemon



Basic React – Exercise 2 – Create tinder "phase" app – Backend – database card schema

```
import mongoose from 'mongoose'

const cardSchema = mongoose.Schema({
  name: String,
  imgUrl: String
})

export default mongoose.model('cards', cardSchema)
```



Basic React – Exercise 2 – Create tinder "fake" app – Backend – Server

```
import express from 'express'
import mongoose from 'mongoose'
import Cors from 'cors'
import Cards from './dbCards.js'

//App Config
const app = express()
const port = process.env.PORT || 8001
const connection_url = 'mongodb+srv://admin:your_password@cluster0.lggjc.mongodb.net/datingDB?retryWri

//Middleware
app.use(express.json())
app.use(Cors())

//DB Config
mongoose.connect(connection_url, {
  useNewUrlParser: true,
  useCreateIndex: true,
  useUnifiedTopology: true
})
```



Basic React – Exercise 2 – Create tinder "phake" app – Backend – Server

```
//API Endpoints
app.get("/", (req, res) => res.status(200).send("Hello TheWebDev"))

app.post('/dating/cards', (req, res) => {
  const dbCard = req.body
  Cards.create(dbCard, (err, data) => {
    if(err) {
      res.status(500).send(err)
    } else {
      res.status(201).send(data)
    }
  })
}

app.get('/dating/cards', (req, res) => {
  Cards.find((err, data) => {
    if(err) {
      res.status(500).send(err)
    } else {
      res.status(200).send(data)
    }
  })
})

//Listener
app.listen(port, () => console.log(`Listening on localhost: ${port}`))
```



React – Homework

- Access to this link and do exercise:
- https://drive.google.com/file/d/1gqA6EuGwg08H_3nxcw4fRE8WBM7Af6m/view?usp=sharing



Q & A



**Thank you for cooperating
Gét gó**

*“Coming together is a beginning;
Keeping together is progress;
Working together is success.”*

- HENRY FORD