

CST4125: Blockchain Development

Week: 21

Title: React

Dr Ian Mitchell



January 2023

Contact and Office Hours



Contact Details

- Name: Dr Ian Mitchell
- Room: TG10
- Address: Middlesex University, Computer Science, London, NW4 4BT
- email: smerf.net

Contact and Office Hours



Contact Details

- Name: Dr Ian Mitchell
- Room: TG10
- Address: Middlesex University, Computer Science, London, NW4 4BT
- email: smerf.net

Office Hours

- During term time only
- When: Autumn Term: Mondays 1100-1300hrs
- Please read notifications or emails
- There are occasions that these could be arranged online, e.g., due to industrial action or inclement weather

Deadlines



Description	Submission	Weight	Deadline	Feedback	
				Formative	Summative
1. Hyperledger	MyLearning	50%	18 th December 2022	LW11-12	12/01/2023
2. Ethereum	MyLearning	50%	2 nd April 2023	LW23-24	24/04/2023
Resits	MyLearning	50-100%	1 st July 2023	None	None
Deferrals	MyLearning	50-100%	1 st July 2023	None	None

Lecture Objectives



- React
- Single Page Apps
- JSX
- Components
- State
- Events
- Hooks

A brief history

- Static HTML
- Dynamic HTML
- Dynamic Content
- Independent HTML pages
- Web 2.0
- Single Page Application, SPA
- One page loads entire website

SPA

- Asynchronous Javascript and XML, AJAX
- Document Object Model, DOM
- Collection of objects in the document
- React: provides instructions of how the DOM should behave
- React DOM are instructions for how the browser DOM should be created [1]

Getting started



- CDN: Content Delivery Network
- Install on VM
- `mkdir react && cd react`
- `npx create-react-app myDir`
- `cd myDir`
- `npm start`



Declarative Programming



- Functional Programming
- Imperative Programming
- Declarative – What
- Imperative – How
- Immutable
 - 1 data is immutable
 - 2 change copies
- Purity
 - do not cause side-effects
 - no changes to non-local (global) variables
 - no changes to constants or static variables
 - always return a value or a function
 - do not change application state
 - they are testable

Structure



- node_modules
- public
- src
- gitignore
- package.json
- remove node_modules
- npm install
- gitignore does **not** update public
 - index.html
 - root element
- src
 - App.css, App.js, index.css, index.js
 - Remove test files
 - index.js makes a call to the App element
 - index.js includes App element in root element
 - App.js has function

JSX



- Javascript for XML
- Create bespoke tags, e.g. `<Cricket />`
- class is reserved in JS, so use `className` in JSX
- Javascript expressions, e.g., `<h2> {title} </h2>`
- Evaluation, e.g., `<h2> {"Hello" +title} </h2>`
- Back-tick allows breaks and other blanks

Linting or Hinting



- Cannot evaluate react
- This goes for Javascript
- Load it in browser and hope for the best
- No compiler
- eslint.org
- Warnings and errors appear

React & Bulma



- Download Bulma into src/css
- `cd src`
- `mkdir css && cd css`
- `wget https://github.com/jgthms/bulma/blob/master/css/bulma?raw=0 bulma.css`
- Import css into App.js

React Bulma nested example

Header

```
1 import './css/bulma2.css';
2 function Header(){
3   return (
4     <div class="block">
5       <section class="hero is-info">
6         <div class="hero-body">
7           <div class="container">
8             <h1 class="title">
9               Smerf
10            </h1>
11            <h2 class="subtitle">
12              Website with React and Bulma
13            </h2>
14          </div>
15        </section>
16      </div>
17    </div>
18    <article class="message is-primary">
19      <div class="message-header">
20        <p>Message Hdr</p>
21      </div>
22      <div class="message-body">
23        Smart miss day which begin. Snug do sold mr smith random.
24      </div>
25    </article>
26  );
27 }
28 export default Header;
```

smerf.net

CST4125:L21

February 22, 2023

12 / 41

React Bulma nested example

List

```
1 function List(){
2   return (
3     <div class="notification is-primary">
4       <ol type="1">
5         <li> Carrots </li>
6         <li> Broccoli</li>
7         <li> Brussels</li>
8       </ol>
9     </div>
10  );
11 }
12 export default List;
```

smerf.net

CST4125:L21

February 22, 2023

13 / 41

React Bulma nested example

App

```
1 import Header from './Header.js';
2 import List from './List.js';
3
4 function App () {
5   return(
6     <Header />
7     <List />
8   );
9 }
10
11 export default App;
```

smerf.net

CST4125:L21

February 22, 2023

14 / 41

React Bulma nested example

Index

```
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import App from './App';
4
5 const root = ReactDOM.createRoot(document.getElementById('root'));
6 root.render(
7   <App />
8 );
```

smerf.net

CST4125:L21

February 22, 2023

15 / 41

Font Awesome

Install

```
npm i icons
```

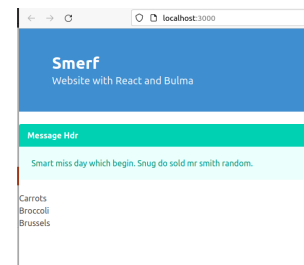
smerf.net

CST4125:L21

February 22, 2023

16 / 41

Output - wide



smerf.net

CST4125:L21

February 22, 2023

17 / 41

Font Awesome Code

```
1 import { FaStar, FaGitSquare, FaGithub, FaFacebook, FaTwitter, FaLinkedin } from "react-  
  icons/fa";  
2  
3 function List(){  
4   return (  
5     <div class="notification is-primary">  
6       <ol type="1">  
7         <li> Carrots </li>  
8         <li> Broccoli</li>  
9         <li> Brussels</li>  
10        <li> Star Red <FaStar color="red" /> </li>  
11        <li> Star Grey <FaStar color="grey" /> </li>  
12        <li> Git Sqr <FaGitSquare /> </li>  
13        <li> Git hub <FaGithub /> </li>  
14        <li> FaceBook <FaFacebook /> </li>  
15        <li> Twitter <FaTwitter /> </li>  
16        <li> LinkedIn <FaLinkedin /> </li>  
17      </ol>  
18    </div>  
19  );  
20 }  
21 }  
22  
23 export default List;
```

smerf.net

CST4125:L21

February 22, 2023

18 / 41

Properties

- Lists?
- Repeatable.
- Create three arrays
- Have a list component that accepts parameters
- Repeat each list in a different column
- Pass the style as a Property
- Pass the title as a Property
- Pass the array as a Property
- Make these columns responsive
- Complete this with react & bulma

smerf.net

CST4125:L21

February 22, 2023

19 / 41

index.js

```
1 import React from 'react';  
2 import ReactDOM from 'react-dom/client';  
3 import App from './App';  
4 //import List from './List';  
5 // import Header from './Header'  
6  
7 const root = ReactDOM.createRoot(document.getElementById('root'));  
8 root.render(  
9   < /* <Header />  
10   <List /> */  
11   <App />  
12 </>  
13 );  
14 );
```

smerf.net

CST4125:L21

February 22, 2023

20 / 41

App.js

```
1 //import './css/bulma2.css';  
2 import Header from './Header.js';  
3 import List from './List.js';  
4 import CreateList from './CreateList.js';  
5  
6 const fruit=["Apple", "Pear", "Orange"];  
7 const veg=["Swede", "Kale", "Turnip"];  
8 const meat=["Fish", "Lamb", "Chicken"];  
9 function App () {  
10   return(  
11     < /* <Header />  
12     <List /> */  
13     <div class="columns">  
14       <div class="column">  
15         <CreateList title="Fruits" items={fruit} style="is-warning" />  
16       </div>  
17       <div class="column">  
18         <CreateList title="Vegetables" items={veg} style="is-danger" />  
19       </div>  
20       <div class="column">  
21         <CreateList title="Meat" items={meat} style="is-primary" />  
22       </div>  
23     </div>  
24   </>  
25   );  
26 }  
27 }  
28  
29 export default App;
```

smerf.net

CST4125:L21

February 22, 2023

21 / 41

CreateList.js

```
1 function CreateList(props) {  
2   let bulmaStyleProp = `notification ${ props.style }`;  
3   return(  
4     <div class={bulmaStyleProp}>  
5       <h2 class="title is-2"> {props.title} </h2>  
6       <ol type="1">  
7         {props.items.map( (item) =>  
8           <li> {item} </li>  
9         )}  
10      </ol>  
11    </div>  
12  );  
13 }  
14  
15 export default CreateList;
```

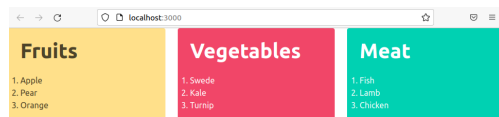
smerf.net

CST4125:L21

February 22, 2023

22 / 41

Output - wide



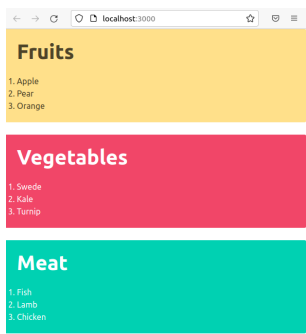
smerf.net

CST4125:L21

February 22, 2023

23 / 41

Output - narrow



State

- `import { useState } from "react";`
- React useState Hook tracks state in a function component
- Where state is data or a property
- The current state, a function that updates the state
- `const [style, setStyle] = useState("");`
- Can we change the style of our set up?
- Select a button, that calls a function and changes the state
- This should automatically update

useState in List.js I

```
1 import { FaStar, FaGitSquare, FaGithub, FaFacebook, FaTwitter, FaLinkedin } from "react-  
  icons/fa";  
2 import { useState } from "react";  
3 function List(){  
4   const notification = 'notification';  
5   const [name, changeName] = useState('Ian');  
6   const [styleName, changeStyle] = useState(notification + ' is-primary');  
7   const handleClick1 = ()=>{  
8     if(name === 'Ian'){  
9       changeName('Sukhy');  
10    } else {  
11      changeName('Ian');  
12    }  
13  }  
14  const handleClick2 = ()=>{  
15    if(styleName === 'notification is-primary'){  
16      changeStyle(notification + ' is-info');  
17    } else {  
18      changeStyle(notification + ' is-primary');  
19    }  
20  }  
21  
22  
23  
24  return (  
25    <div class={styleName}>  
26      <h2 class="title is-2"> List for {name} </h2>  
27      <ol type="1">  
28        <li> Git Sqr <FaGitSquare /> </li>  
29        <li> Git hub <FaGithub /> </li>
```

useState in List.js II

```
30     <li> FaceBook <FaFacebook /> </li>  
31     <li> Twitter <FaTwitter /> </li>  
32   </ol>  
33   <button class="button is-link" onClick={handleClick1}> {name} </button>  
34   <button class="button is-warning" onClick={handleClick2}> {styleName} </  
35   button>  
36 </div>  
37 );  
38 }  
39  
40 export default List;
```

App.js

```
1 //import './css/bulma2.css';  
2 import Header from './Header.js';  
3 import List from './List.js';  
4 import CreateList from './CreateList.js';  
5  
6 const fruit=["Apple", "Pear", "Orange"];  
7 const veg=["Swede", "Kale", "Turnip"];  
8 const meat=["Fish", "Lamb", "Chicken"];  
9 function App () {  
10  return(  
11    <  
12      <Header />  
13      <List />  
14      <div class="columns">  
15        <div class="column">  
16          <CreateList title="Fruits" items={fruit} style="is-warning" />  
17        </div>  
18        <div class="column">  
19          <CreateList title="Vegetables" items={veg} style="is-danger" />  
20        </div>  
21        <div class="column">  
22          <CreateList title="Meat" items={meat} style="is-primary" />  
23        </div>  
24      </div>  
25    </>  
26  );  
27 }  
28  
29 export default App;
```

Problem

- Spot any?

Problem



- Spot any?
- When I click Ian, I get Sukhy and the button changes to Sukhy.



Problem



- Spot any?
- When I click Ian, I get Sukhy and the button changes to Sukhy.
- I want the complement to display
- When I push the new value, I want the old value to display
- Any ideas?



useState in List.js I



```
1 import { FaGitSquare, FaGithub, FaFacebook, FaTwitter, FaLinkedin } from "react-icons/fa"
2 import { useState } from "react";
3 const arrName = ["Ian", "Sukhy"];
4 function List(){
5   const notification = 'notification';
6   const [name, changeName] = useState(0);
7   const [oldName, keepName] = useState(1);
8   const [styleName, changeStyle] = useState(notification + ' is-primary');
9   const handleClick1 = ()=>{
10     if (name === 0) {
11       keepName(0);
12       changeName(1);
13     } else {
14       keepName(1);
15       changeName(0);
16     }
17   }
18   const handleClick2 = ()=>{
19     if (styleName === 'notification is-primary'){
20       changeStyle(notification + ' is-info');
21     } else {
22       changeStyle(notification + ' is-primary');
23     }
24   }
25   return (
26     <div class={styleName}>
27       <h2 class="title is-2"> List for {arrName[name]} </h2>
28       <ol type="1">
```



useState in List.js II



```
30 <li> Git Sqr <FaGitSquare /> </li>
31 <li> Git hub <FaGithub /> </li>
32 <li> FaceBook <FaFacebook /> </li>
33 <li> Twitter <FaTwitter /> </li>
34 <li> LinkedIn <FaLinkedin /> </li>
35 </ol>
36 <button class="button is-link" onClick={handleClick1}> {arrName[oldName]} </
37 <button class="button is-warning" onClick={handleClick2}> {styleName} </
38 </div>
39 );
40 );
41 }
42
43 export default List;
```



App.js



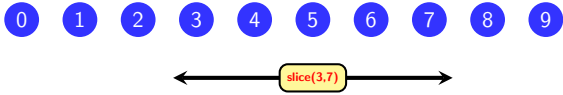
```
1 //import './css/bulma2.css';
2 import Header from './Header.js';
3 import List from './List.js';
4 import CreateList from './CreateList.js';
5
6 const fruit=["Apple", "Pear", "Orange"];
7 const veg=["Svede", "Kale", "Turnip"];
8 const meat=["Fish", "Lamb", "Chicken"];
9 function App () {
10   return (
11     <Header />
12     <List />
13     <div class="columns">
14       <div class="column">
15         <CreateList title="Fruits" items={fruit} style="is-warning" />
16       </div>
17       <div class="column">
18         <CreateList title="Vegetables" items={veg} style="is-danger" />
19       </div>
20       <div class="column">
21         <CreateList title="Meat" items={meat} style="is-primary" />
22       </div>
23     </div>
24   );
25 }
26
27 export default App;
```



Spread recap



Spread recap



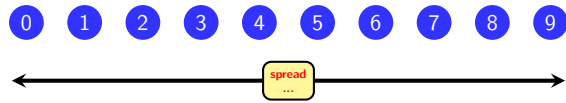
Spread recap



Spread recap



Spread recap



Slice Array I

```
1 import {useState} from 'react';
2 import './css/bulma2.css';
3 function ArrayIntegerAsList() {
4   const arrayInt = [ 51, 52, 53, 54, 55];
5   const [listInt, setInt] = useState(arrayInt);
6
7   const getRandom = () => {
8     return Math.floor(Math.random()*100);
9   }
10
11  const handleAddIntStart = () => {
12    let r = getRandom();
13    console.log("Random:"+r);
14    setInt( existingInts => {
15      return [r, ...existingInts]
16    });
17  }
18  const handleAddIntEnd = () => {
19    let r = getRandom();
20    setInt( existingInts => {
21      return [...existingInts, r]
22    });
23  }
24
25  const handleRmIntStart = () => {
26    setInt( existingInts => {return existingInts.slice(1)});
27  }
28
29  const handleRmIntEnd = () => {
30    setInt( existingInts => {return existingInts.slice(0, existingInts.length-1)});
31  }
32}
```

Slice Array II

```
31 }
32 return (
33   <div class="notification is-primary">
34     <h2 class="title is-large">
35       List
36     </h2>
37     <div class="content">
38       <ul>
39         {listInt.map(
40           (item, idx) => {
41             return (
42               <li key={idx}>
43                 <span>
44                   {item}
45                 </span>
46             )
47           )}
48         </ul>
49       </div>
50     </div>
51     <div class="block">
52       <button class="button" onClick={handleAddIntStart}> Add Integer to Start
53     </button>
54     <button class="button" onClick={handleAddIntEnd}> Add Integer to End</
55     button>
56     <button class="button is-danger" onClick={handleRmIntStart}> Remove Int
57     from Start</button>
58   </div>
59 )
```

Slice Array III



```
58     <button class="button is-danger" onClick={handleRmIntEnd}> Remove Int  
59     from End</button>  
60   </div>  
61 );  
62 }  
63 }  
64 export default ArrayIntegerAsList;
```

smerf.net

CST4125:L21

February 22, 2023

36 / 41

useEffect



- Typically, used for temporal events
- Pure function
- Timed events
- Side effects
- Rendering first, then side effects
- Every component mount
- Every component render
- Every component render with a condition
- When component unmounts
- no dependencies, then execute every render

```
import { useEffect } from 'react';  
function MyComponent(){  
  useEffect(function, [dependencies])  
  return ( ... )  
}
```

smerf.net

CST4125:L21

February 22, 2023

37 / 41

React



- `npx create-react-app mdx-coin`
- Developers will `gitignore` module files, when uploading to github
- Download development code and rebuild
- `git clone` repos
- `cd` repos
- `npm install`
- Sometimes, `rm package-lock.json`

smerf.net

CST4125:L21

February 22, 2023

38 / 41

Reading



- Ch 2 & 3 in [1]
- Practice React on [React Tutorial](#)
- [Read the Docs](#)

smerf.net

CST4125:L21

February 22, 2023

39 / 41

Summary



- React
- Component
- Declarative
- Hooks
- `useState`
- `useEffect`
- SPAs

smerf.net

CST4125:L21

February 22, 2023

40 / 41

References I



- [1] Alex Banks and Eve Porcello. *Learning React: Modern Patterns for Developing React Apps*. O'Reilly Media, 2020.

smerf.net

CST4125:L21

February 22, 2023

41 / 41