Integrating React with APIs

Objectives

- Connect a React App with an API
- Name the steps to make a GET request and update state in a React.Component
- Name the steps to make a POST request and update state in a React.Component

Steps to populate state based on an API

1. define initial state in a class property `this.state = { people: [] }`
2. define an async componentDidMount method
3. fetch the data and await the response (inside componentDidMount)
4. turn it into json and await the response

Fetch Data Example

- define an async method on your component
- fetch the data and await the response
- turn it into json and await the response
- call setState()

```
class App extends React.Component {

  state = {
    people: []
  }

  async componentDidMount() {
    const response = await fetch('http://localhost:8181/api/people')
    const json = await response.json()
    this.setState({people: json})
  }

  render() {
    return <div>...</div>
  }

}
```

Steps for handling a POST request in a React.Component

1. Define an async method that takes the data as a parameter. Note an event handler needs to use this method after it gets information out of the DOM (such as a form)
2. use fetch to do the POST request
3. get the added data from the json
4. use this.setState() to update state with the new item

Post

```
class App extends React.Component {

  async createItem(item) {
    const response = await fetch('http://localhost:8181/api/people', {
      method: 'POST',
      body: JSON.stringify(item),
      headers: {
        'Content-Type': 'application/json',
        'Accept': 'application/json',
      }
    })
    const person = await response.json()
    this.setState({people: [...this.state.people, person]})
  }

  render() {
    return <div>...</div>
  }

}
```

Objectives

- Connect a React App with an API