

create-react-app-example

- ☐ What is react
- ☐ Website
- ☐ create-react-app (huh???)
- ☐ GitHub – <https://developer.github.com/v3/>
- ☐ Material-UI / React Router (react libraries, gives us components to reuse)
- ☐ layout picture
- ☐ demo time!

Application Setup

```
create-react-app repoBrowser  
cd repoBrowser  
yarn start
```

- ☐ Demo live reload / etc, explain everything that CRA does so you don't have to!

```
yarn add material-ui@next  
yarn add material-ui-icons@next  
yarn add react-router  
yarn add react-router-dom
```

Add Routing

- ☐ index.js changes
- ☐ discuss the new urls, touch briefly that we will use this later

```

import React from 'react'
import ReactDOM from 'react-dom'
import App from './App'
import { MuiThemeProvider, createMuiTheme } from
'material-ui/styles'
import { BrowserRouter, Route, Redirect, Switch} from
'react-router-dom'

const theme = createMuiTheme()

const repoFinder = () => {
  return (
    <MuiThemeProvider theme={theme}>
      <BrowserRouter>
        <Switch>
          <Route exact path="/user/:username"
component={App} />
          <Redirect from="/" to="/user/nllarson" />
        </Switch>
      </BrowserRouter>
    </MuiThemeProvider>
  )
}

ReactDOM.render(repoFinder(),
document.getElementById('root'))

```

```
yarn start
```

- ☐ Show that react-create-app stil shows up, but notice url

Build Our App

- ☐ Create Placeholder files for **Profile.js**, **RepoList.js**, and **RepoFilter.js**

- ☐ App.js

```
class App extends Component {
  render() {
    const { classes } = this.props
    return (
      <div className={classes.root}>
        <Grid container spacing={24}>
          <Grid item xs={12} sm={3}>
            <Profile/>
          </Grid>
          <Grid item xs={12} sm={9}>
            <RepoFilter/>
            <RepoList/>
          </Grid>
        </Grid>
      </div>
    )
  }
}

const styles = theme => ({
  root: {
    marginTop: theme.spacing.unit * 3,
    width: '100%'
  }
})

export default withStyles(styles)(App)
```

Build Profile Component(s)

- ☐ Profile.js

Profile Dummy State:

```

this.state = {
  user: {
    login: 'nllarson',
    id: 208498,
    avatar_url:
'https://avatars1.githubusercontent.com/u/208498?
v=4',
    html_url: 'https://github.com/nllarson',
    name: 'Nick Larson',
    company: '@objectpartners',
    location: 'Omaha, NE ',
    public_repos: 13,
    followers: 16,
    following: 9
  },

  orgs: [
    {
      login: 'objectpartners',
      id: 1569148,
      avatar_url:
'https://avatars1.githubusercontent.com/u/1569148?
v=4'
    },
    {
      login: 'OJUG',
      id: 7458223,
      avatar_url:
'https://avatars2.githubusercontent.com/u/7458223?
v=4'
    }
  ]
}

```

Profile Render:

```

render() {
  const { user, orgs } = this.state
  const { classes } = this.props
  return (
    <Grid container spacing={24}
direction="column">
      <Grid item>
        <Grid container className=
{classes.bottomborder} direction="column"
align="center" justify="center" spacing={0}>
          <Grid item>
            <Avatar src={user.avatar_url}
className={classes.avatar} />
          </Grid>
          <Grid item>
            <Typography type="headline"
color="secondary">
              {user.name}
            </Typography>
          </Grid>
          <Grid item>
            <Typography type="caption">{user.login}
          </Typography>
        </Grid>
      </Grid>
      <Grid item>
        <Grid container className=
{classes.bottomborder} justify="space-between"
align="center">
          <Grid item>
            <Typography type="headline">
{user.followers}</Typography>
            <Typography type="caption"
component="small">
              followers
            </Typography>
          </Grid>
        </Grid>
      </Grid>
    </Grid>
  )
}

```

```

        <Typography type="headline">
{user.public_repos}</Typography>
        <Typography type="caption"
component="small">
        repositories
        </Typography>
    </Grid>
    <Grid item>
        <Typography type="headline">
{user.following}</Typography>
        <Typography type="caption"
component="small">
        following
        </Typography>
    </Grid>
</Grid>
</Grid>
<Grid item>
    <Grid container className=
{classes.bottomborder} justify="center"
align="center">
        {orgs.map(org => (<Grid item key={org.id}
><Avatar src={org.avatar_url} /></Grid>))}
    </Grid>
</Grid>
</Grid>
)
}

```

/** CSS **/

```

const styles = theme => ({
  bottomborder: {
    textAlign: 'center',
    borderBottom: '1px solid #eee',
    paddingTop: theme.spacing.unit * 2
  },
  avatar: {
    width: 200,
    height: 200,

```

```
        marginBottom: 20
      }
    })
```

More Components!!! (Code Reuse)

- ☐ ProfileStat.js

```
    return (
      <Grid item>
        <Typography type="headline">{value}
      </Typography>
        <Typography type="caption" component="small">
          {label}
        </Typography>
      </Grid>
    )
```

- ☐ Convert **followers**, **repositories**, **following** to **<ProfileStat/>** in Profile.js

```
<ProfileStat value={user.followers} label="followers"
/>
<ProfileStat value={user.public_repos}
label="repositories" />
<ProfileStat value={user.following} label="following"
/>
```

Build RepoList Components()

- ☐ RepoList.js

RepoList Render:

```

render() {
  const { repos } = this.state
  const { filter } = this.props
  return (
    <List>
      {repos.sort((a, b) => {
        return Date.parse(b.pushed_at) -
Date.parse(a.pushed_at)
      })
      .map(repo => {
        return <Repo key={repo.id} repo={repo} />
      })}
    </List>
  )
}

```

RepoList Dummy State:

```

this.state = {
  repos: [
    {
      id: 35244465,
      name: 'awesome-java',
      full_name: 'nllarson/awesome-java',
      html_url:
'https://github.com/nllarson/awesome-java',
      description: 'A curated list of awesome
Java frameworks, libraries and software. Inspired by
awesome-python.',
      pushed_at: '2015-05-07T20:24:00Z',
      stargazers_count: 1,
      language: null,
      forks_count: 0
    },
    {

```



```

      id: 48047948,
      name: 'testing-presentation',
      html_url:
'https://github.com/nllarson/testing-presentation',
      description: 'Why do we test?',
      pushed_at: '2015-12-17T06:23:28Z',
      stargazers_count: 2,
      language: 'JavaScript',
      forks_count: 1
    },
    {
      id: 1087533,
      name: 'dotfiles',
      full_name: 'nllarson/dotfiles',
      html_url:
'https://github.com/nllarson/dotfiles',
      description: 'my dotfiles',
      pushed_at: '2014-04-04T03:31:34Z',
      stargazers_count: 3,
      language: 'Shell',
      forks_count: 2
    }
  ]
}

```

- ☐ Repo.js
- ☐ inline functions as components
- ☐ install moment

```

import React, { Component } from 'react'
import { withStyles } from 'material-ui/styles'
import { ListItem, Typography, Card, CardActions,
CardHeader, CardContent, Chip, Avatar, IconButton }
from 'material-ui'
import { CallSplit, Star, OpenInNew } from 'material-
ui-icons'

```

```

import moment from 'moment'

const StarAvatar = () => {
  return (
    <Avatar>
      <Star />
    </Avatar>
  )
}

const ForkAvatar = () => {
  return (
    <Avatar>
      <CallSplit />
    </Avatar>
  )
}

class Repo extends Component {
  render() {
    const { classes, repo } = this.props
    const timeUpdated =
moment(repo.pushed_at).fromNow()

    const title = (
      <span>
        {repo.name}
        <IconButton href={repo.html_url}
target="_blank" aria-label="open">
          <OpenInNew />
        </IconButton>
      </span>
    )

    return (
      <ListItem>
        <Card className={classes.repo}>
          <CardHeader title={title} />
          <CardContent>

```

```

        <Typography component="p">
{repo.description}</Typography>
        </CardContent>
        <CardActions>
            {repo.language ? <Chip label=
{repo.language} /> : ''}
            <Chip avatar={<StarAvatar />} label=
{repo.stargazers_count || 0} />
            <Chip avatar={<ForkAvatar />} label=
{repo.forks_count || 0} />
            <Typography className=
{classes.lastUpdate} type="caption" color="secondary"
align="right">
                {timeUpdated}
            </Typography>
        </CardActions>
    </Card>
</ListItem>
)
}
}

const styles = theme => ({
  repo: {
    width: `100%`
  },
  lastUpdate: {
    flex: '1 1 auto',
    marginRight: 20
  },
  cardActions: {
    marginRight: 20
  }
})

```

Build RepoFilter and make it work

- ☐ RepoFilter.js

```

import React, { Component } from 'react'
import PropTypes from 'prop-types'
import { TextField } from 'material-ui'

export default class RepoFilter extends Component {
  static propTypes = {
    onUpdate: PropTypes.func.isRequired
  }

  render() {
    return (
      <TextField
        id="repo-filter"
        placeholder="Filter repositories..."
        fullWidth
        margin="normal"
        onChange={({target: {value}}) =>
this.props.onUpdate(value)}
      />
    )
  }
}

```

- ☐ Add filter update handler to App
- ☐ Add filter to App state, pass to RepoList

```

class App extends Component {
  constructor(args) {
    super(args)
    this.state = { filter: '' }
  }

  handleFilterUpdate = filter => {
    this.setState({ filter })
  }
}

```

```

render() {
  const { filter } = this.state
  const username = this.props.match.params.username
  return (
    <Grid container spacing={24}>
      <Grid item xs={12} sm={3}>
        <Profile />
      </Grid>
      <Grid item xs={12} sm={9}>
        <RepoFilter onUpdate=
{this.handleFilterUpdate} />
        <RepoList filter={filter} />
      </Grid>
    </Grid>
  )
}
}

export default App

```

- ☐ Update RepoList.js

```

{repos
  .filter(repo => {
    return (
      (repo.name &&
repo.name.toLowerCase().includes(filter.toLowerCase()
)) ||
      (repo.description &&
repo.description.toLowerCase().includes(filter.toLowecase()))
    )
  })
  .sort((a, b) => {
    return Date.parse(b.pushed_at) -
Date.parse(a.pushed_at)

```

```

    })
    .map(repo => {
      return <Repo key={repo.id} repo={repo} />
    })
  }
}

```

- ☐ Explain that the props of these functions are changing as we type into the filter (`.setState(...)` is happening). This causes React to reconcile te DOM and show the changes! WOW!!!!

Hook into GitHub API

- ☐ Now...let's bring this thing to life!!!!
- ☐ Remember earlier when talking about Router and URL, let's use that as the key to search for the proper GH repo
- ☐ ApiUtils.js and github-api.js

API code / utilities

```

const ApiUtils = {
  checkStatus: function(response) {
    if (response.status >= 200 && response.status <
300) {
      return response;
    } else {
      let error = new Error(response.statusText);
      error.response = response;
      throw error;
    }
  }
}

export default ApiUtils

```

```

import ApiUtils from './ApiUtils'

const BASE_URL = 'https://api.github.com'

const fetchRepos = username => {
  const url = `${BASE_URL}/users/${username}/repos?
per_page=250`
  return fetch(url)
    .then(ApiUtils.checkStatus)
    .then(response => response.json())
}

const fetchUser = username => {
  const url = `${BASE_URL}/users/${username}`
  return fetch(url)
    .then(ApiUtils.checkStatus)
    .then(response => response.json())
}

const fetchOrgs = username => {
  const url = `${BASE_URL}/users/${username}/orgs`
  return fetch(url)
    .then(ApiUtils.checkStatus)
    .then(response => response.json())
}

const fetchUserInformation = username => {
  return Promise.all([fetchUser(username),
fetchOrgs(username)]).then(([user, orgs]) => {
    return { user, orgs }
  })
}

export { fetchRepos, fetchUser, fetchOrgs,
fetchUserInformation }

```

Wire it all together

- ☐ Grab **username** in **App.js** and send it along to **<Profile/>** and **<RepoList/>**
- ☐ Clean state in **Profile.js** and **RepoList.js** and add the **componentDidMount()**

```
render() {
  const { filter } = this.state
  const username = this.props.match.params.username
  return (
    <Grid container spacing={24}>
      <Grid item xs={12} sm={3}>
        <Profile username={username} />
      </Grid>
      <Grid item xs={12} sm={9}>
        <RepoFilter onUpdate=
{this.handleFilterUpdate} />
        <RepoList username={username} filter=
{filter} />
      </Grid>
    </Grid>
  )
}
```

```
class Profile extends Component {
  constructor(args) {
    super(args)

    this.state = {
      user: {},
      orgs: []
    }
  }
}
```



```

    getProfile = username => {
      fetchUserInformation(username).then(({ user, orgs
    }) => {
        this.setState({ user, orgs })
      })
    }

    componentWillMount() {
      this.getProfile(this.props.username)
    }
    ...
    ...
    ...
    ...

```

```

export default class RepoList extends Component {
  constructor() {
    super()

    this.state = {
      repos: []
    }
  }

  getRepos = username => {
    fetchRepos(username).then(repos => {
      this.setState({ repos })
    })
  }

  componentWillMount() {
    this.getRepos(this.props.username)
  }

  ...
  ...

```

...

...