

# From Zero to Hero with ReactJS

Maurice de Beijer

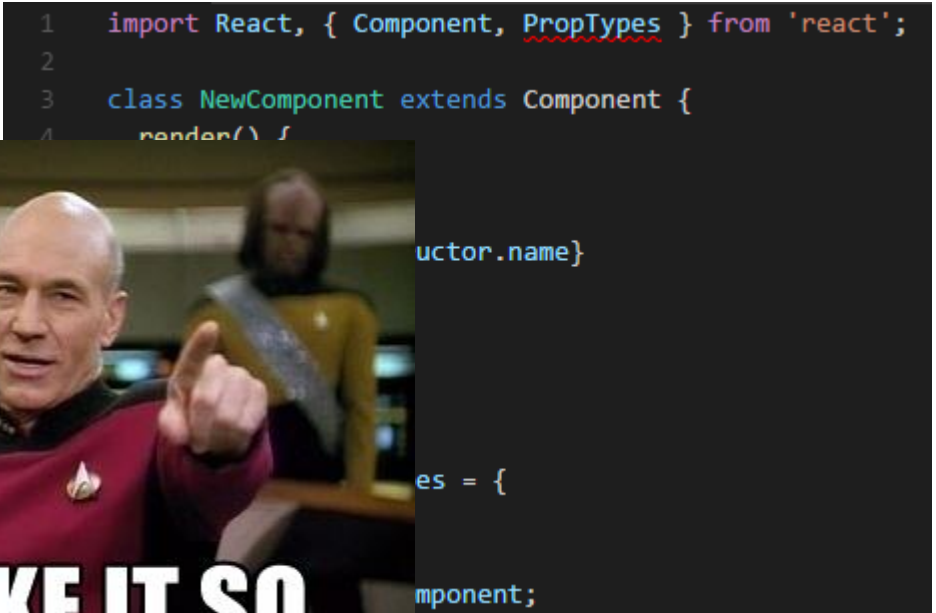
@mauricedb

@react\_tutorial

# Workshop goal

- Start a new React project using create-react-app
- Compose the user interface using multiple components
- Add the Bootstrap CSS framework
- Using state and props in components
- Using input controls in React
- Container versus Presentational components

# Follow along



- Git repository: <https://github.com/mauricedb/nitflex>
- Slides: <https://raw.githubusercontent.com/mauricedb/nitflex/master/react-workshop.pdf>

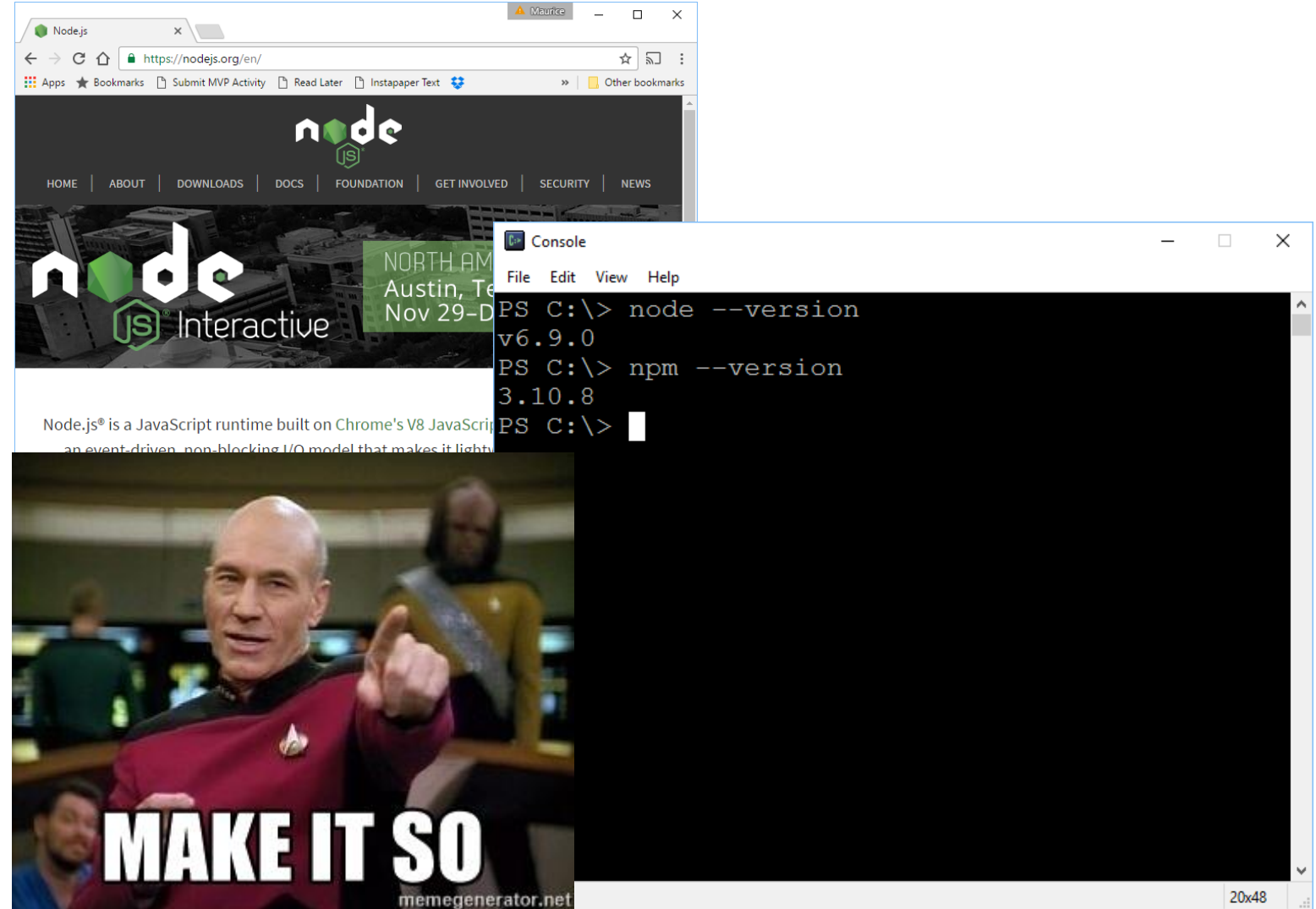
Type it out  
by hand?

*"Typing it drills it into your brain much better than simply copying and pasting it. You're forming new neuron pathways. Those pathways are going to help you in the future. Help them out now!"*

# Prerequisites

Install Node & NPM

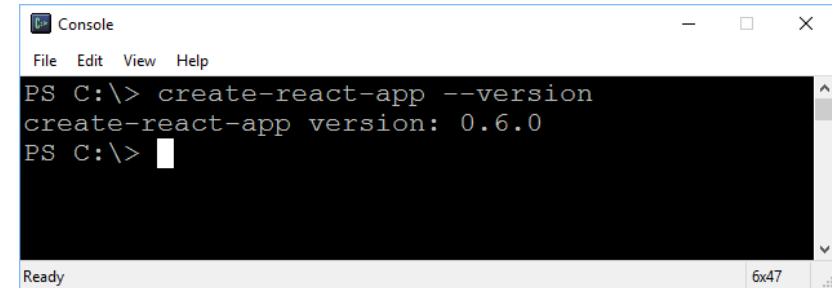
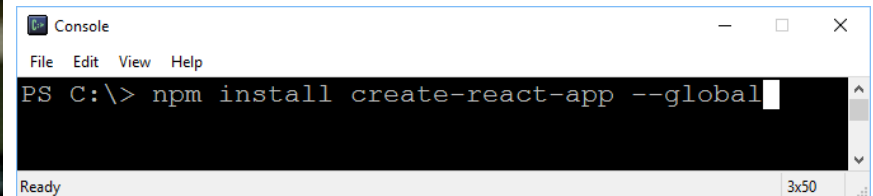
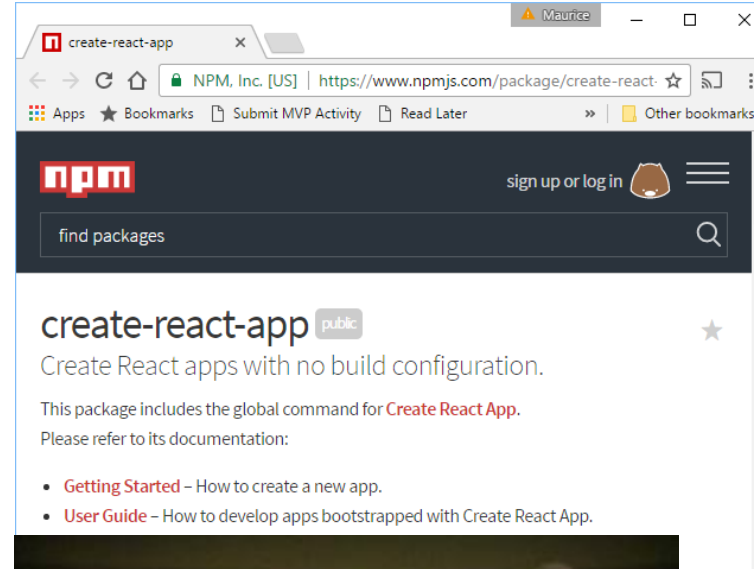
# Install Node.js & NPM





# Install Create-React-App

# Install create-react-app

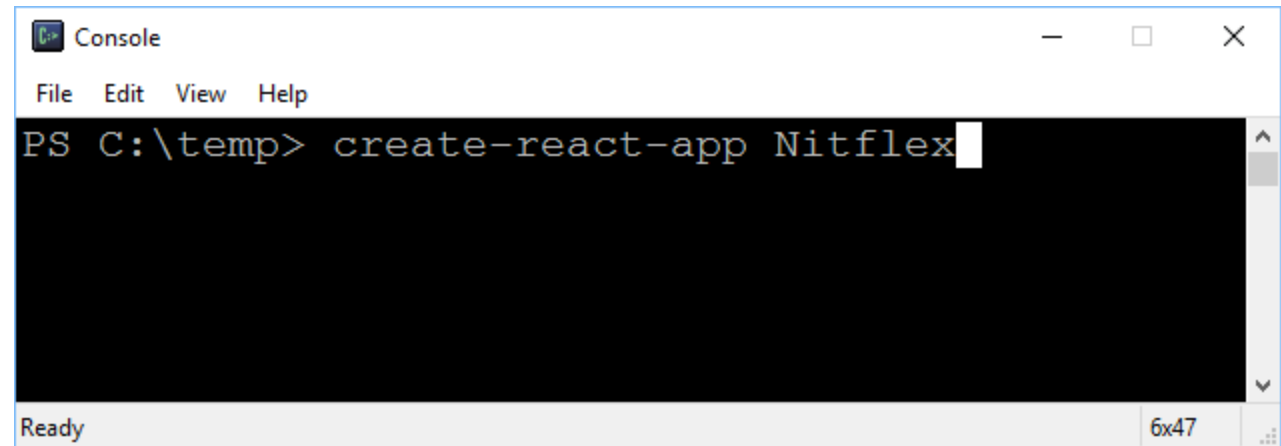






# Create a basic application

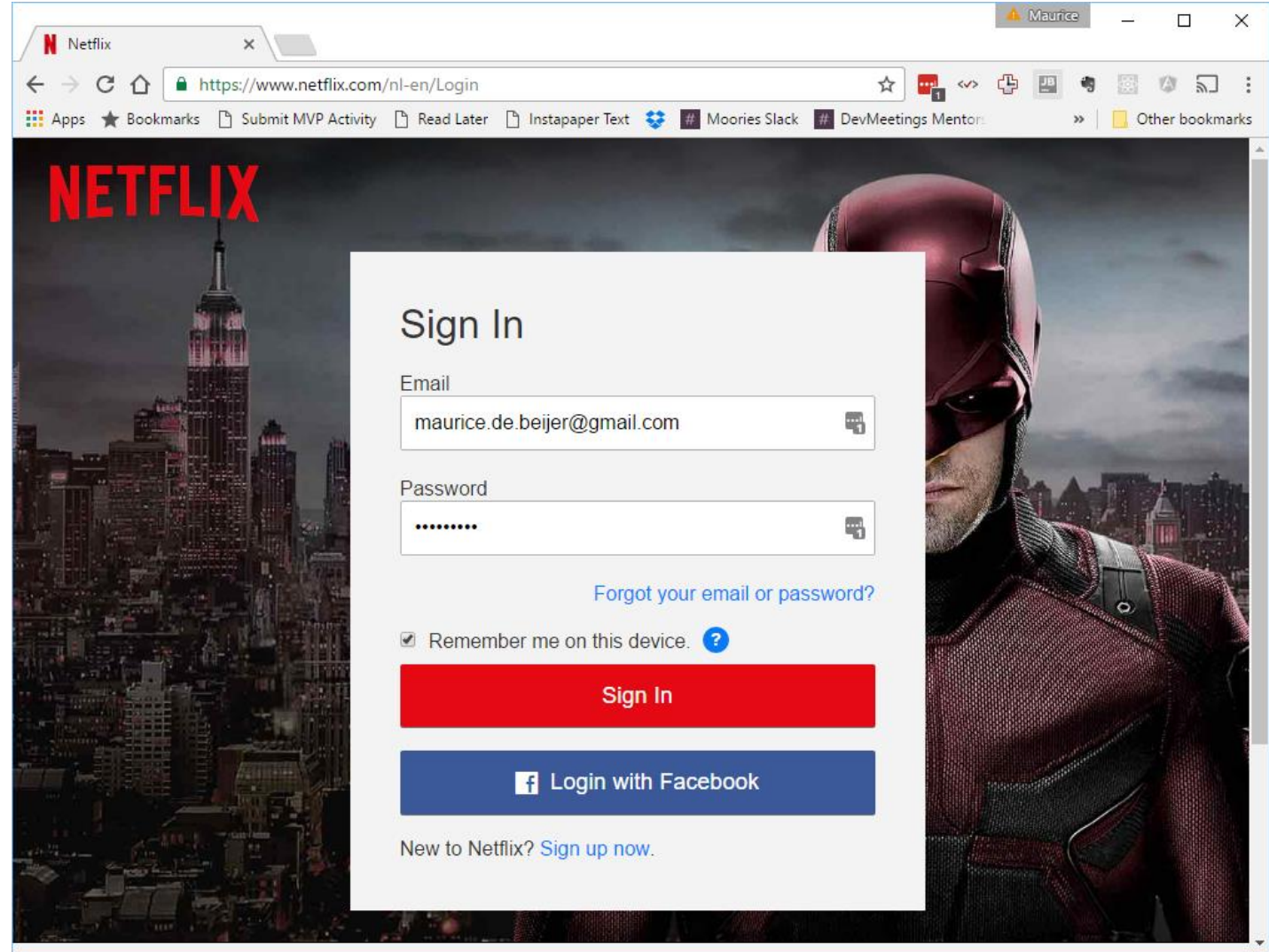
# Create app



```
PS C:\temp> create-react-app Nitflex
```



# Netflix Login



Netflix

Sign In

Email  
maurice.de.beijer@gmail.com

Password  
.....

[Forgot your email or password?](#)

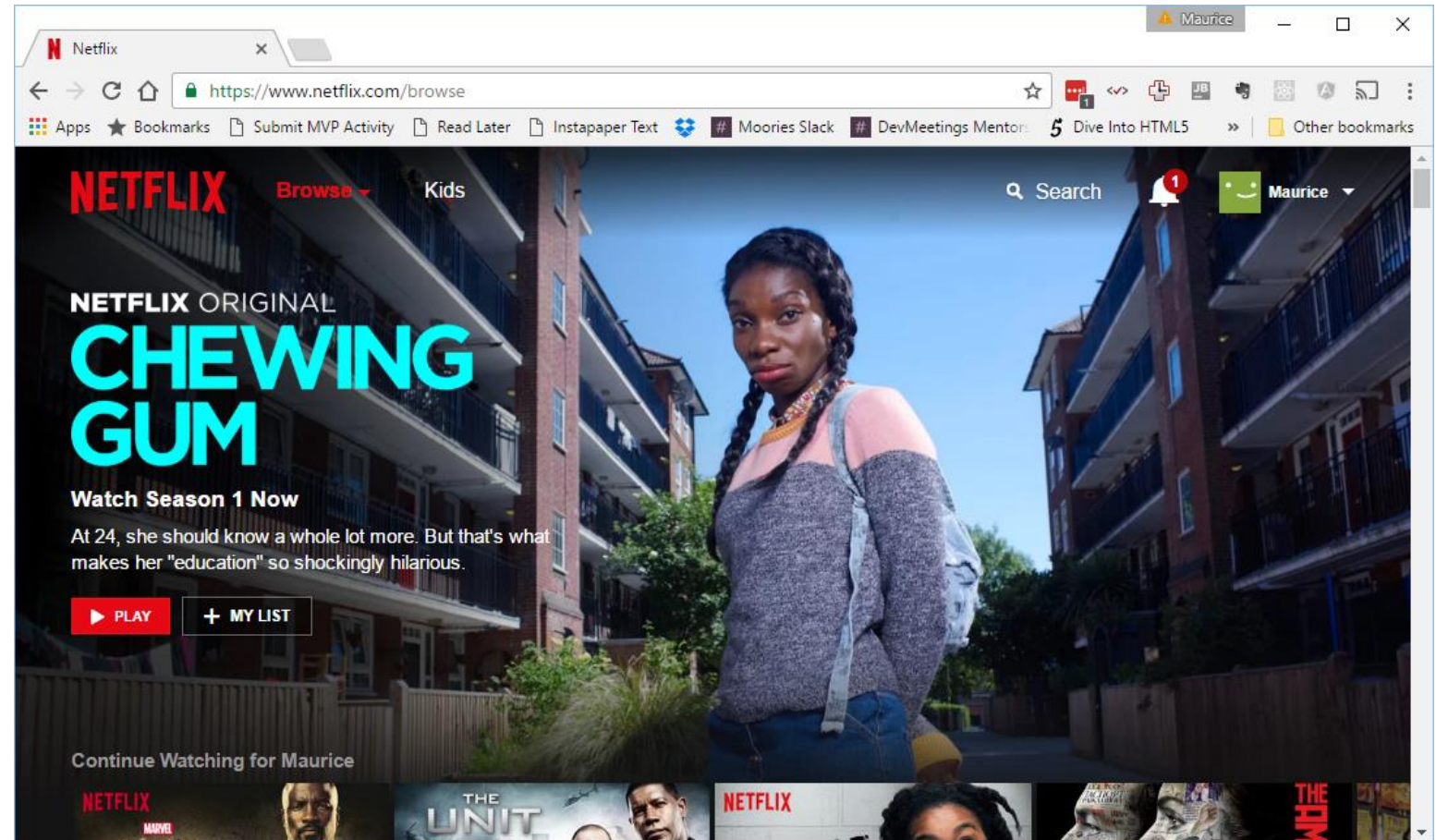
☒ Remember me on this device. [?](#)

[Sign In](#)

[Login with Facebook](#)

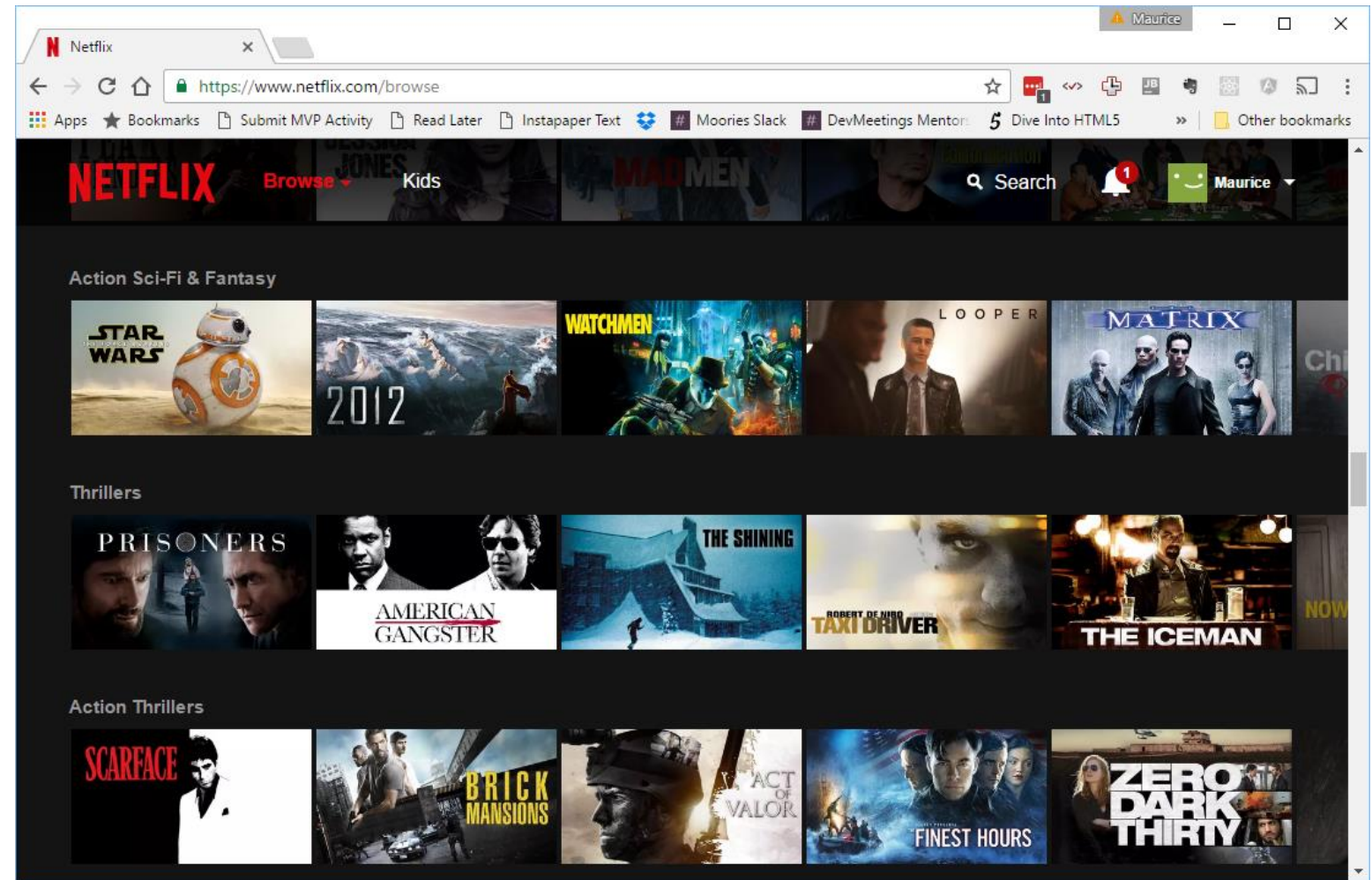
New to Netflix? [Sign up now.](#)

# Header & Billboard

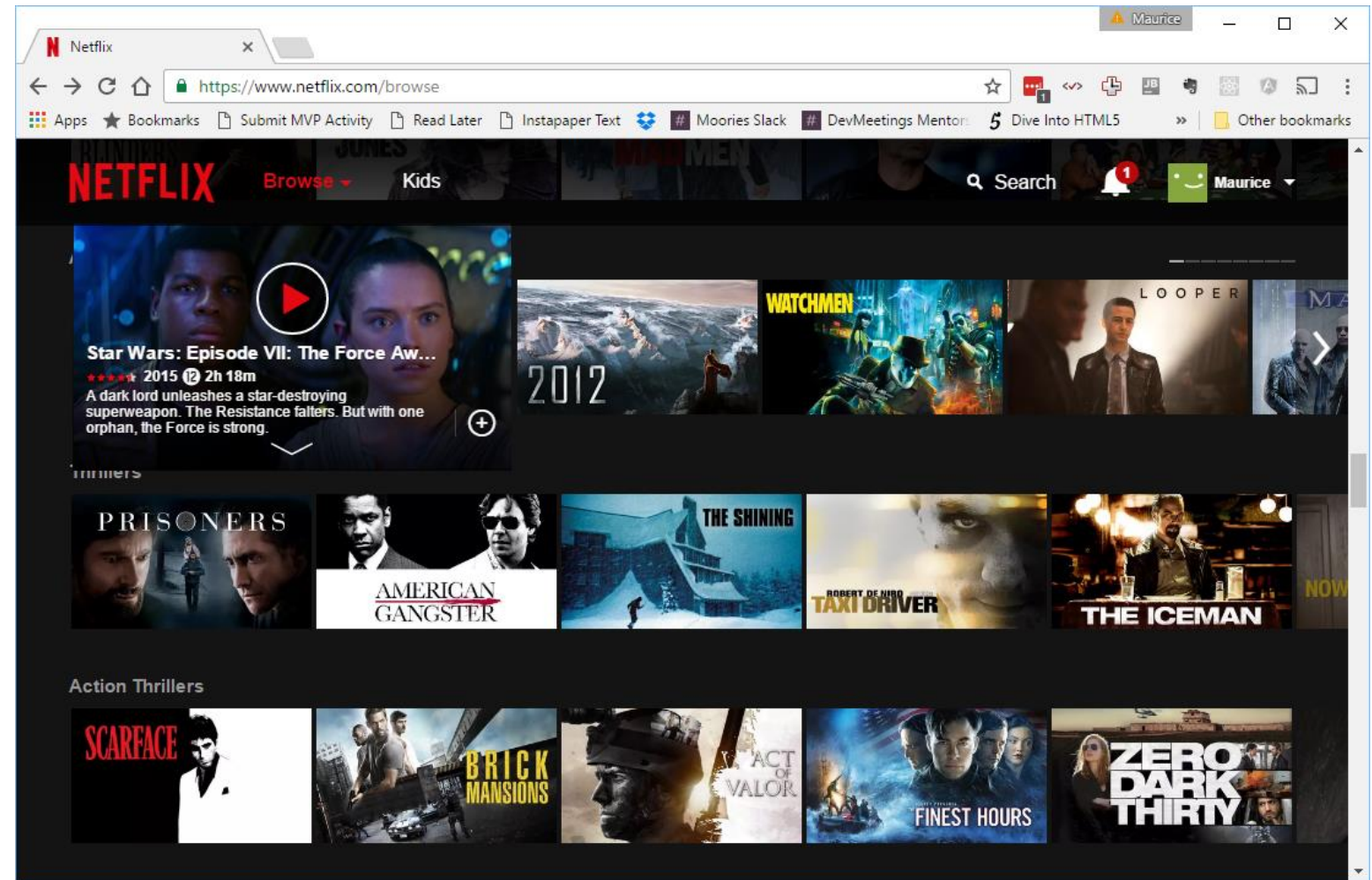




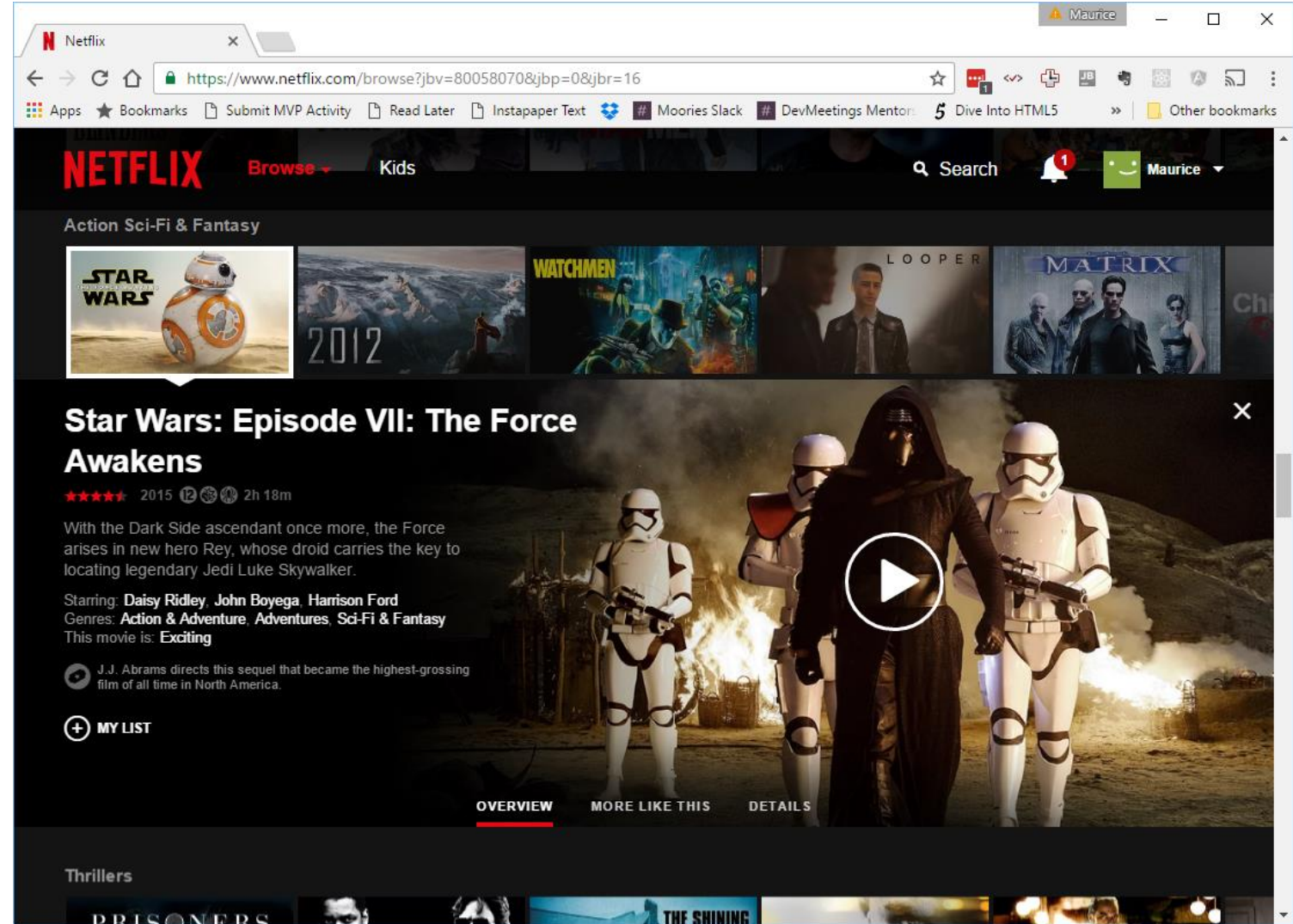
# Genres



# Hover effect



# Movie Expanded





# React Developer Tools

Developer Tools - https://www.netflix.com/browse

Elements Console Sources Network Timeline Profiles Application Security Audits Adblock AngularJS React

Trace React Updates Highlight Search Use Regular Expressions

```
▼ Router model={title: "Netflix", models: {...}, pathEvaluator: {...}, ...} renderSource="client" routeHandler={route: {...}, components: Array[3], uiView: "homeScreen", ...}
▼ LayoutContext model={title: "Netflix", models: {...}, pathEvaluator: {...}, ...} renderSource="client" routeHandler={route: {...}, components: Array[3], uiView: "homeScreen", ...}
▼ BaseRouteHandler model={title: "Netflix", pathEvaluator: {...}, node: {...}} renderSource="client" routeHandler={route: {...}, components: Array[3], uiView: "homeScreen", ...}
▼ div ref="main" lang="nl" dir="ltr" onTouchStart=bound handleTouchStart(...)
▼ AkiraLayout model={title: "Netflix", pathEvaluator: {...}, node: {...}} renderSource="client" routeHandler={route: {...}, components: Array[3], uiView: "homeScreen", ...}
▼ div className="bd dark-background" lang="nl"
  ▼ Header ref="header" model={title: "Netflix", pathEvaluator: {...}, node: {...}} showOVLLink=false pathEvaluator={...streaming: false, __next: Array[2], __head: Array[2], ...}
    ▼ div ref="hdSpace" className="hdSpace" role="header"
      ▼ div id="hdPrintarget" className="akira-header head-opaque" onMouseDown=bound onMouseDown(...)
        ► MainViewLink resolver=getHomeRoute() className="logo icon-logoupdate" aria-label="Netflix" ...</MainViewLink>
        ► ul role="navigation" ...</ul>
        ► div className="secondary-nav" ...</div>
        ► SystemNotifications model={title: "Netflix", pathEvaluator: {...}, node: {...}} ...</SystemNotifications>
      </div>
    </Header>
    ► ct connected=false onCastStop() onCastStop=...</t>
    ► FreezeWrapper key="FreezeWrapper" shouldFreeze=false
    ▼ TimeoutTransitionGroup key="mainView" className="mainView" role="main" component="div" ...
    ▼ ReactTransitionGroup className="mainView" role="main" component="div" ...
      ▼ div className="mainView" role="main"
        ▼ TimeoutTransitionGroupChild key="$/browse" ref="/$/browse" enterTimeout=200 leaveTimeout=200 afterTransitionLeave=bound afterPageTransitionLeave(...)
          ▼ FreezeOnChangedPageKey key="/browse"
            ▼ HomeContent key="/browse" model={title: "Netflix", pathEvaluator: {...}, node: {...}} renderSource="client" routeHandler={route: {...}, components: Array[3], uiView: "homeScreen", ...}
              ▼ Lolomo ref="lolomo" model={path: Array[1], streaming: false, __next: Array[2], ...} renderSource="client" routeHandler={route: {...}, components: Array[3], uiView: "homeScreen", ...}
                ▼ div className="lolomo is-fullbleed"
                  ▼ BillboardRow key="0" model={path: Array[2], streaming: false, __next: Array[2], ...} rowNum=0 listContext="billboard" ...
                    ▼ div ref="billboard-row" className="billboard-row" onClick=bound stopAutoCycle() onMouseEnter=bound stopAutoCycle(...)
                      ► div viewport(billboardOriginals) key="bb-0" ref="billboard-originals" model={path: Array[6], streaming: false, __next: Array[2], ...} videoModel={path: Array[3], streaming: false, __next: Array[2], ...}
                      </div>
                    </BillboardRow>
                  ▼ LolomoRow key="1" model={path: Array[2], streaming: false, __next: Array[2], ...} rowNum=1 listContext="genre" ...
                    ▼ div key="genre1" className="lolomoRow lolomoRow_title_card" data-list-context="genre"
                      ► RowHeader id=9889 title="Films gebaseerd op een boek" ...</RowHeader>
                      ► Row model={path: Array[2], streaming: false, __next: Array[2], ...} totalItems=40 rowNum=1 ...
                    ▼ div className="rowContainer rowContainer_title_card" id="row-1"
                      ▼ PreTrackedContainers
                        ▼ div className="ptrack-container"
                          ▼ div className="rowContent slider-hover-trigger-layer" onMouseLeave=bound handleRowBlur(...)
                            ▼ Slider ref="slider" itemsInRow=4 totalItems=40 onSliderMove=bound handleSliderMove(...)
                              ▼ div className="slider"
                                ► PaginationIndicator totalPages=7 activePage=0 ...</PaginationIndicator>
                                ▼ div className="sliderMask showPeek" onMouseLeave=bound handleMouseLeavesSliderMask(...)
                                  ▼ div ref="sliderContent" className="sliderContent row-with-x-columns" style=null onTouchStart=bound handleTouchStart(...)
                                    ► SliderItem key="item_0/$title_80000438_1_0" ref="item_0" viewportIndex=0 viewportPosition="leftEdge" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_1/$title_80041653_1_1" ref="item_1" viewportIndex=1 viewportPosition="middle" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_2/$title_80075560_1_2" ref="item_2" viewportIndex=2 viewportPosition="middle" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_3/$title_70121494_1_3" ref="item_3" viewportIndex=3 viewportPosition="middle" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_4/$title_700652701_1_4" ref="item_4" viewportIndex=4 viewportPosition="middle" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_5/$title_70114023_1_5" ref="item_5" viewportIndex=5 viewportPosition="rightEdge" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_6/$title_70142827_1_6" ref="item_6" viewportIndex=6 viewportPosition="rightPeek" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_7/$title_60024942_1_7" ref="item_7" viewportIndex="" viewportPosition="" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_8/$loading-title-0" ref="item_8" viewportIndex="" viewportPosition="" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_9/$loading-title-1" ref="item_9" viewportIndex="" viewportPosition="" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_10/$loading-title-2" ref="item_10" viewportIndex="" viewportPosition="" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_11/$loading-title-3" ref="item_11" viewportIndex="" viewportPosition="" additionalClasses={...}</SliderItem>
                                    ► SliderItem key="item_12/$loading-title-4" ref="item_12" viewportIndex="" viewportPosition="" additionalClasses={...}</SliderItem>
                                  </div>
                                ▼ span ref="handleNext" className="handle handleNext active" tabIndex=0 onClick=bound advanceNext(...)
                                  <div className="indicator-icon icon-rightCaret"/>
                                </span>
                              </div>
                            </div>
                          </Slider>
                        </div>
                      ► JawBoneRow model={path: Array[2], streaming: false, __next: {...}, ...} sliderRef=t(...) jawBoneRankNum=null ...</JawBoneRow>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
  Router LayoutContext BaseRouteHandler div AkiraLayout div FreezeWrapper TimeoutTransitionGroup ReactTransitionGroup div TimeoutTransitionGroupChild FreezeOnChangedPageKey HomeContent Lolomo div LolomoRow
```

Search by Component Name

<LolomoRow> (\$ in the console)

Key "25"

Props

- columnsInRow: 6
- jawBoneRankNum: null
- listContext: "genre"
- listType: "Flat"
- model: {...}
- rowNum: 25
- showJawBone: false

State

- dispose: null
- fullDataLoaded: false
- generation: -1
- key: ""
- prevGeneration: 1
- prevKey: "8003f7b0-4cca-4fcc-8097-6a0cc698a6a\_13955824"

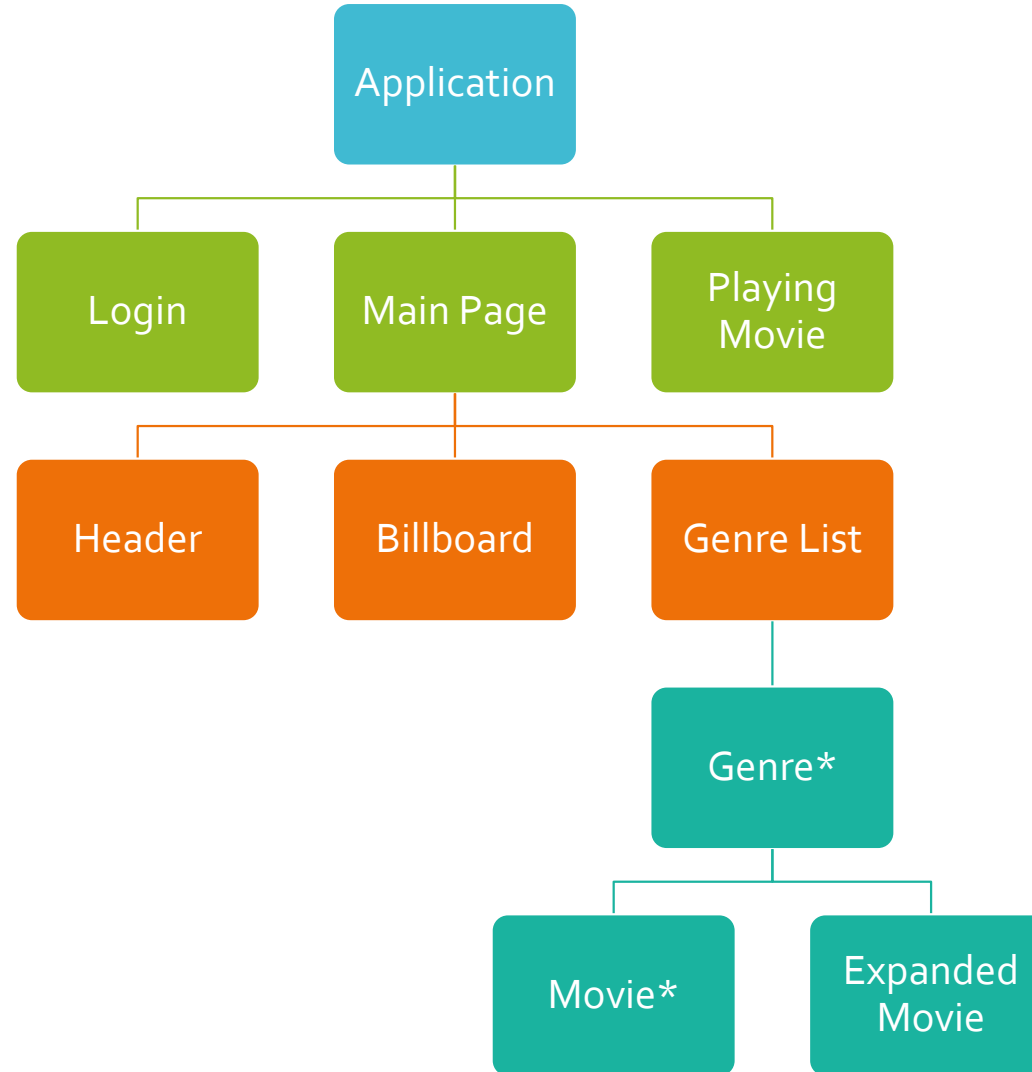
Context

- columnsInRow: 6
- currentPage: "homeScreen"
- isInitialRender: false
- isKidsPage: false
- isRTL: false
- models: {...}
- node: {...}
- requestId: "63feef8-2c0e-43f3-974f-06de8a568b30-210595269"
- routeHandler: {...}



# Main components

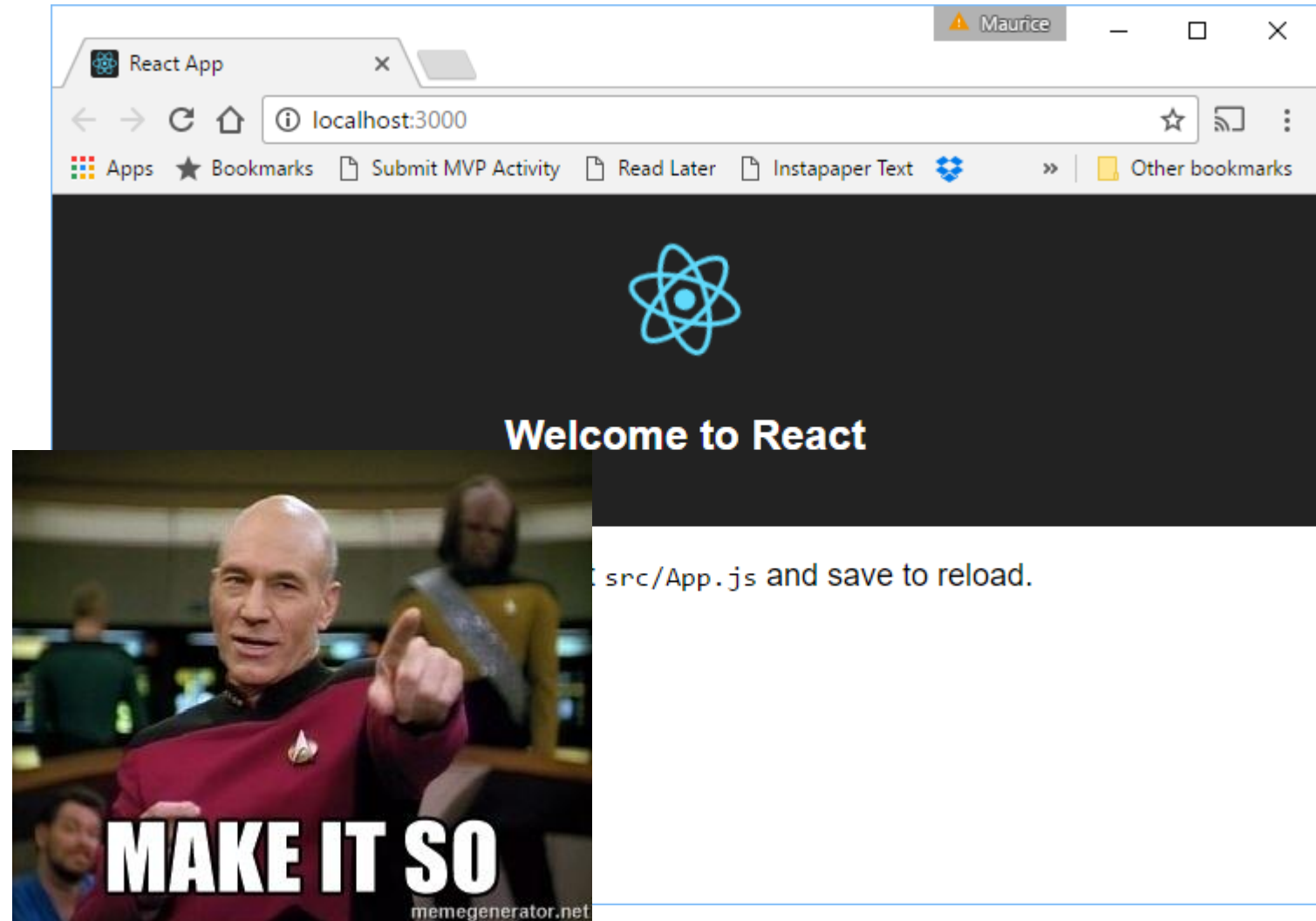
# Layout





# Running the basic application

npm start



# Top level skeleton

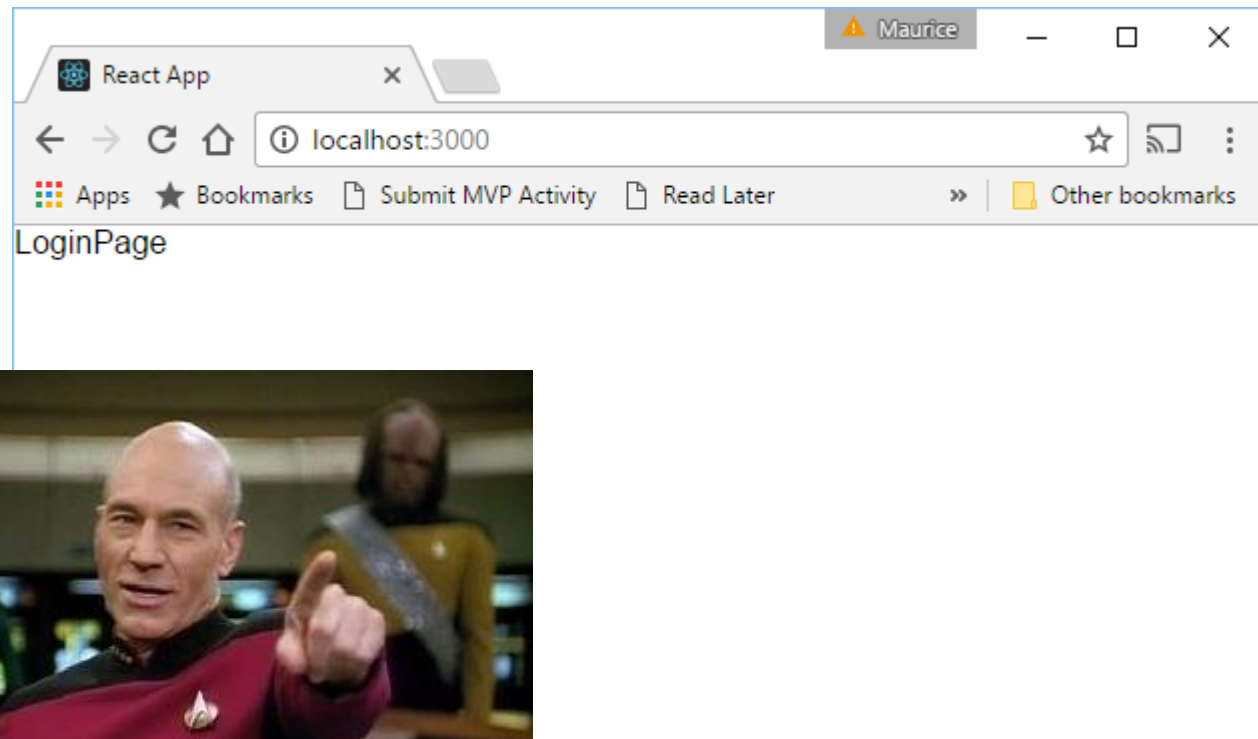
- Create the following three folders and an **index.jsx** in each of them
  - login-page
  - main-page
  - playing-movie
- Create a skeleton component in each of these files
  - Name the components **LoginPage**, **MainPage** and **PlayingMovie**

```
1  import React, { Component, PropTypes } from 'react';
2
3  class NewComponent extends Component {
4    render() {
5      return (
6        <div>
7          {this.constructor.name}
8        </div>
9      );
10   }
11 }
12
13 NewComponent.propTypes = {
14 };
15
16 export default NewComponent;
```

# App.jsx

```
1  import React, { PropTypes } from 'react';
2  import LoginPage from './login-page';
3  import PlayingMovie from './playing-movie';
4  import MainPage from './main-page';
5
6  const App = ({ user, movies, playing, loginAsUser, startPlaying, stopPlaying }) => {
7    let component = null;
8
9    if (!user) {
10      component = <LoginPage loginAsUser={loginAsUser} />;
11    } else if (playing) {
12      component = <PlayingMovie movie={playing} stopPlaying={stopPlaying} />;
13    } else {
14      component = <MainPage user={user} movies={movies} startPlaying={startPlaying} />;
15    }
16
17    return (
18      <div className="container">
19        {component}
20      </div>
21    );
22  };
23
24  App.propTypes = {
25    user: PropTypes.object,
26    movies: PropTypes.array,
27    playing: PropTypes.object,
28    loginAsUser: PropTypes.func.isRequired,
29    startPlaying: PropTypes.func.isRequired,
30    stopPlaying: PropTypes.func.isRequired,
31  };
32
33  export default App;
```

# The browser





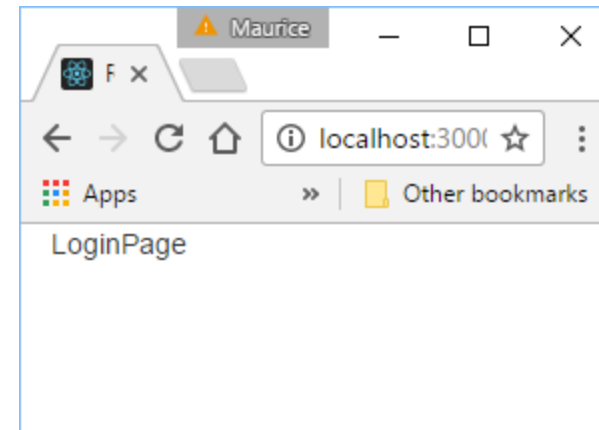
# Bootstrap

Adding a bit of style



# Add Bootstrap

```
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import App from './App';
4
5  // npm install bootstrap --save
6  import '../node_modules/bootstrap/dist/css/bootstrap.min.css';
7
8  ReactDOM.render(
9    <App />,
10    document.getElementById('root')
11  );
```



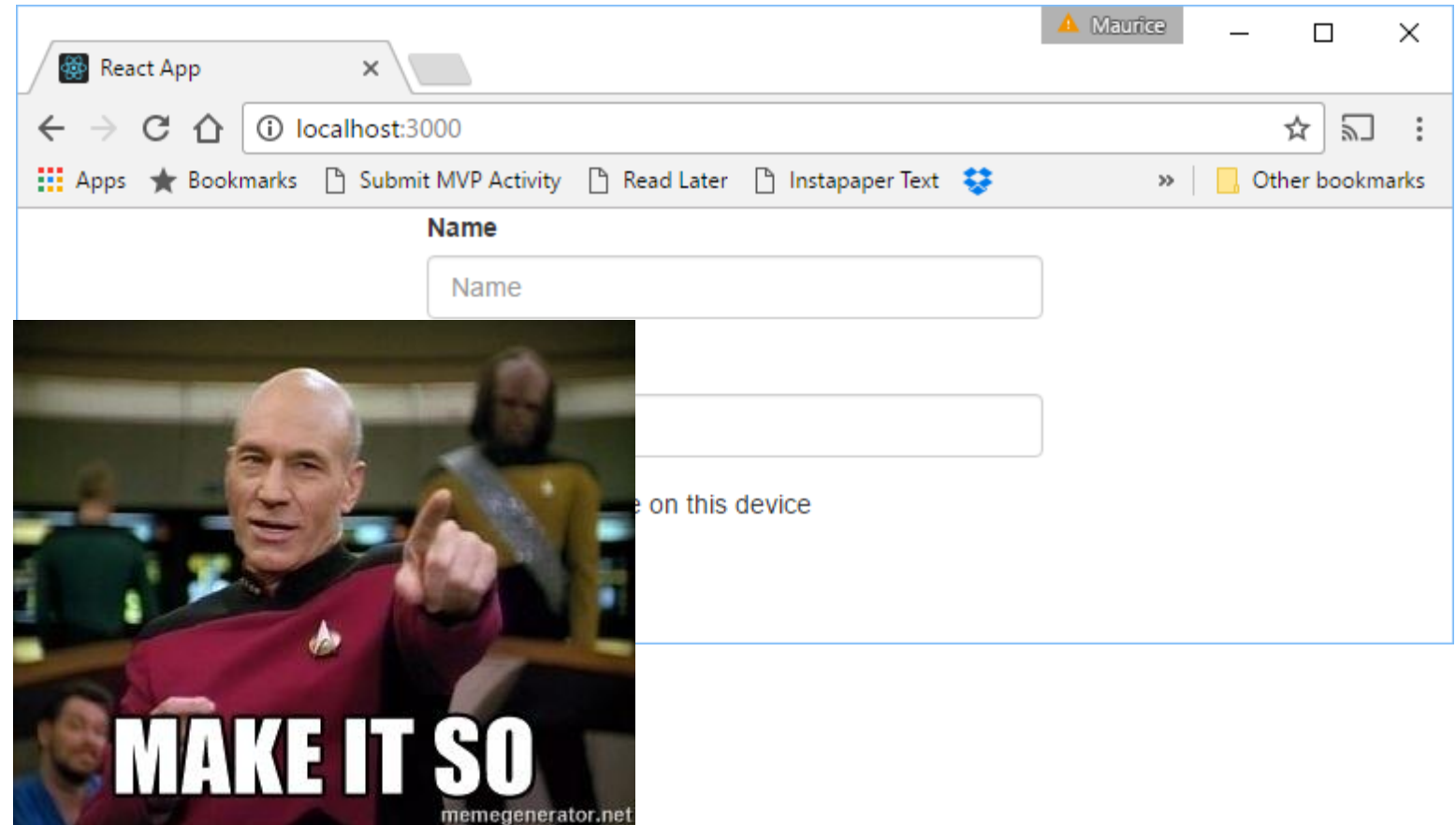


# Login Page UI

# Login Page

```
3  class LoginPage extends Component {
4    render() {
5      return (
6        <form className="col-sm-6 col-sm-offset-3">
7          <div className="form-group">
8            <label htmlFor="username">Name</label>
9            <input
10              type="text"
11              className="form-control"
12              placeholder="Name"
13              id="username"
14            />
15          </div>
16          <div className="form-group">
17            <label htmlFor="password">Password</label>
18            <input
19              type="password"
20              id="password"
21              className="form-control"
22              placeholder="Password"
23            />
24          </div>
25          <div className="checkbox">
26            <label htmlFor="rememberMe">
27              <input
28                type="checkbox"
29                id="rememberMe"
30              />
31              Remember me on this device
32            </label>
33          </div>
34          <button className="btn btn-primary">Sign In</button>
35        </form>
36      );
37    }
38  }
```

# Login Page



# Login Page State

# Login Page State

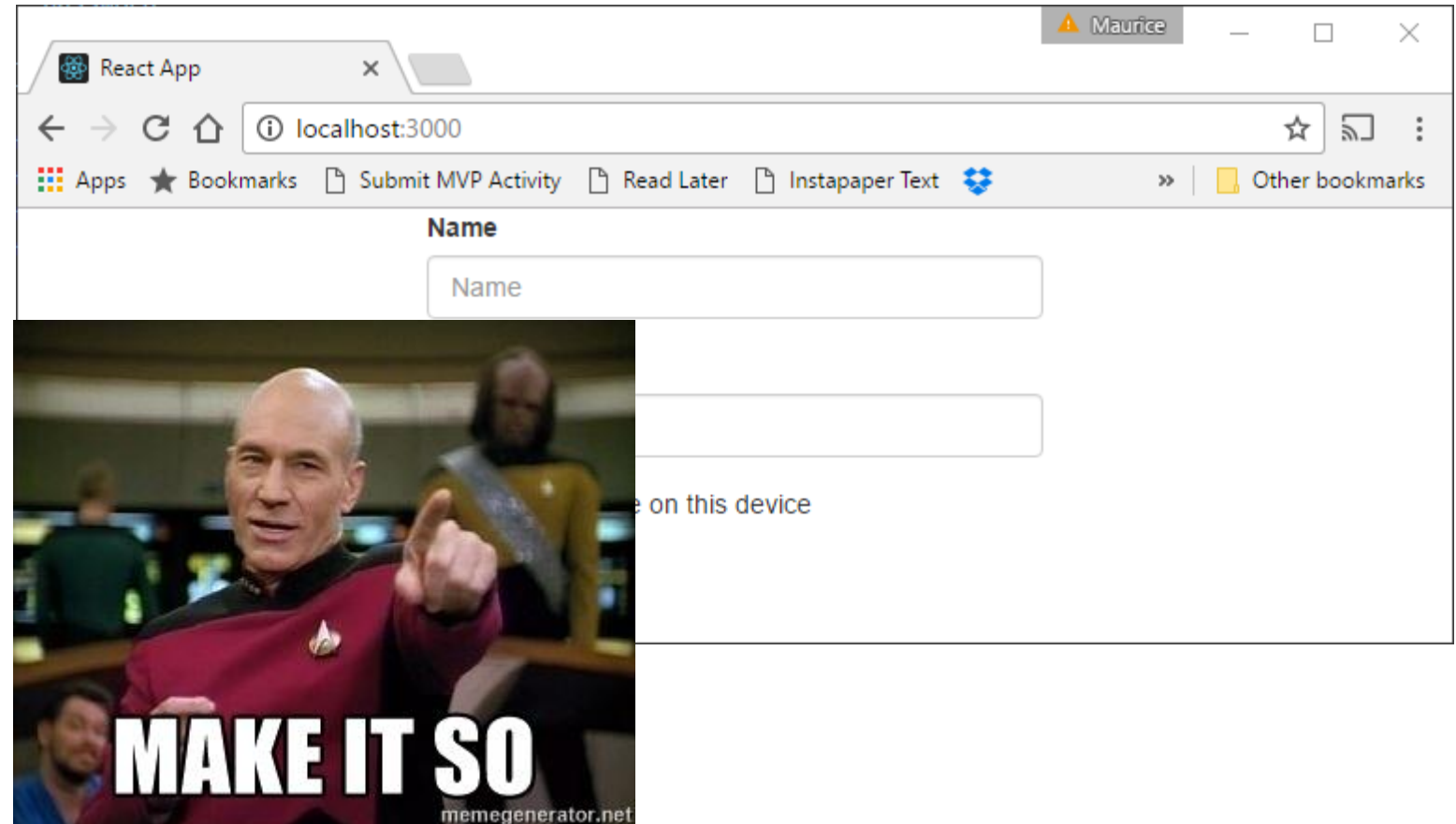
```
3 class LoginPage extends Component {
4   constructor() {
5     super();
6
7     this.state = {
8       name: '',
9       password: '',
10      rememberMe: false,
11    };
12
13    this.onChangeName = this.onChangeName.bind(this);
14    this.onChangePassword = this.onChangePassword.bind(this);
15    this.onChangeRememberMe = this.onChangeRememberMe.bind(this);
16    this.login = this.login.bind(this);
17  }
18
19  onChangeName(e) {
20    const name = e.target.value;
21    this.setState({ name });
22  }
23
24  onChangePassword(e) {
25    const password = e.target.value;
26    this.setState({ password });
27  }
28
29  onChangeRememberMe() {
30    const rememberMe = !this.state.rememberMe;
31    this.setState({ rememberMe });
32  }
33
34  login(e) {
35    e.preventDefault();
36    this
37      .props
38      .loginAsUser({ name: this.state.name, rememberMe: this.state.rememberMe });
39  }
40}
```

# Login Page State

```
43 <form className="col-sm-6 col-sm-offset-3">
44   <div className="form-group">
45     <label htmlFor="username">Name</label>
46     <input
47       type="text"
48       className="form-control"
49       placeholder="Name"
50       id="username"
51       value={this.state.name}
52       onChange={this.onNameChange}
53     />
54   </div>
55   <div className="form-group">
56     <label htmlFor="password">Password</label>
57     <input
58       type="password"
59       id="password"
60       className="form-control"
61       placeholder="Password"
62       value={this.state.password}
63       onChange={this.onPasswordChange}
64     />
65   </div>
66   <div className="checkbox">
67     <label htmlFor="rememberMe">
68       <input
69         type="checkbox"
70         id="rememberMe"
71         checked={this.state.rememberMe}
72         onChange={this.onRememberMeChange}
73       />
74       Remember me on this device
75     </label>
76   </div>
77   <button
78     className="btn btn-primary"
79     disabled={this.state.name.length === 0 || this.state.password.length === 0}
80     onClick={this.login}
81   >
82     Sign In
83   </button>
84 </form>
```

# Login Page

## State







# Application State

# Application State

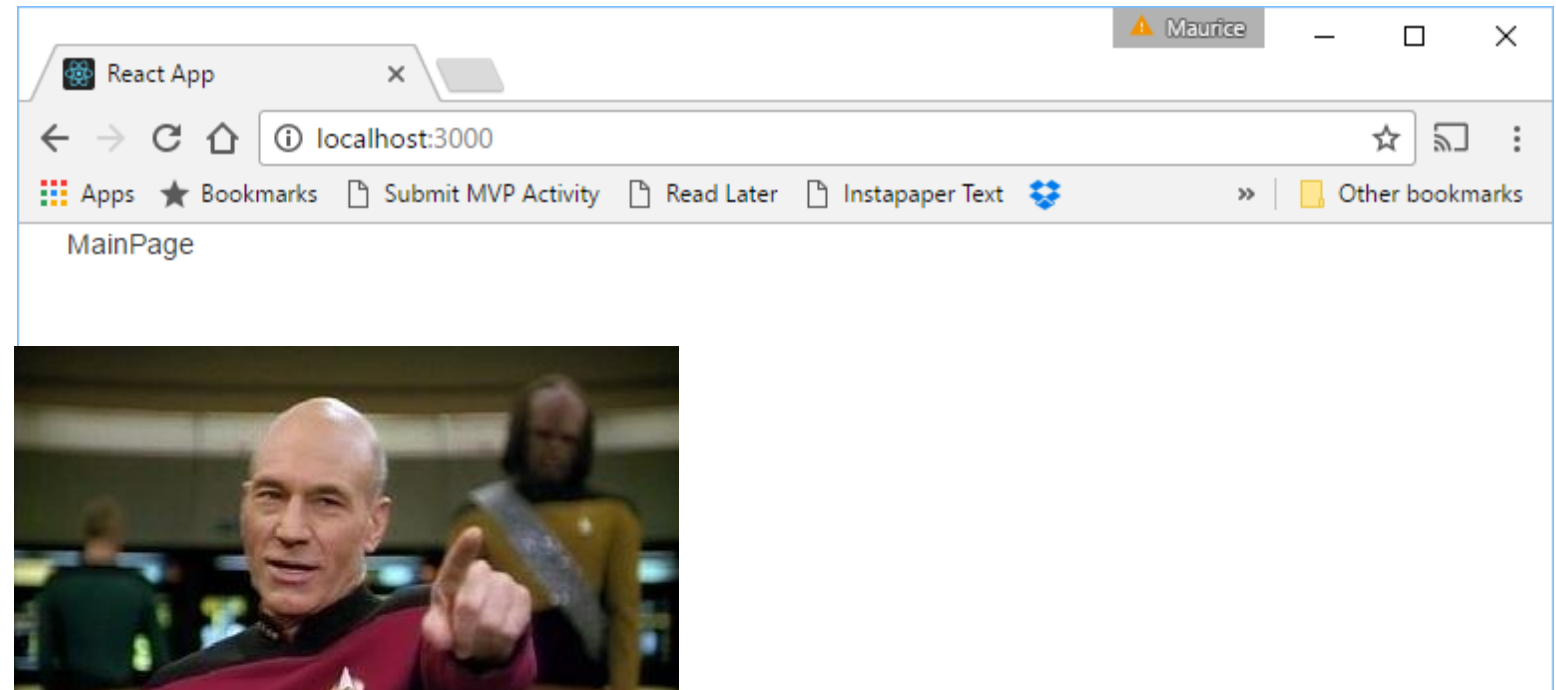
- Split application state from application presentation
  - Rename **App.js** => **app-presentation.jsx**
- Add **app-container.jsx**
  - Import **app-container.jsx** in **index.js**

```
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import AppContainer from './app-container';
4
5  // npm install bootstrap --save
6  import '../node_modules/bootstrap/dist/css/bootstrap.min.css';
7
8  ReactDOM.render(
9    <AppContainer />,
10    document.getElementById('root')
11  );
```

# AppContainer

```
1  import React, { Component } from 'react';
2  import AppPresentation from './app-presentation';
3
4  class AppContainer extends Component {
5    constructor(props) {
6      super(props);
7
8      this.state = { user: null };
9
10     this.loginAsUser = this.loginAsUser.bind(this);
11   }
12
13   loginAsUser(user) {
14     this.setState({ user });
15   }
16
17   render() {
18     const { user } = this.state;
19
20     return (
21       <AppPresentation
22         user={user}
23         loginAsUser={this.loginAsUser}
24       />
25     );
26   }
27 }
28
29 export default AppContainer;
```

# Login



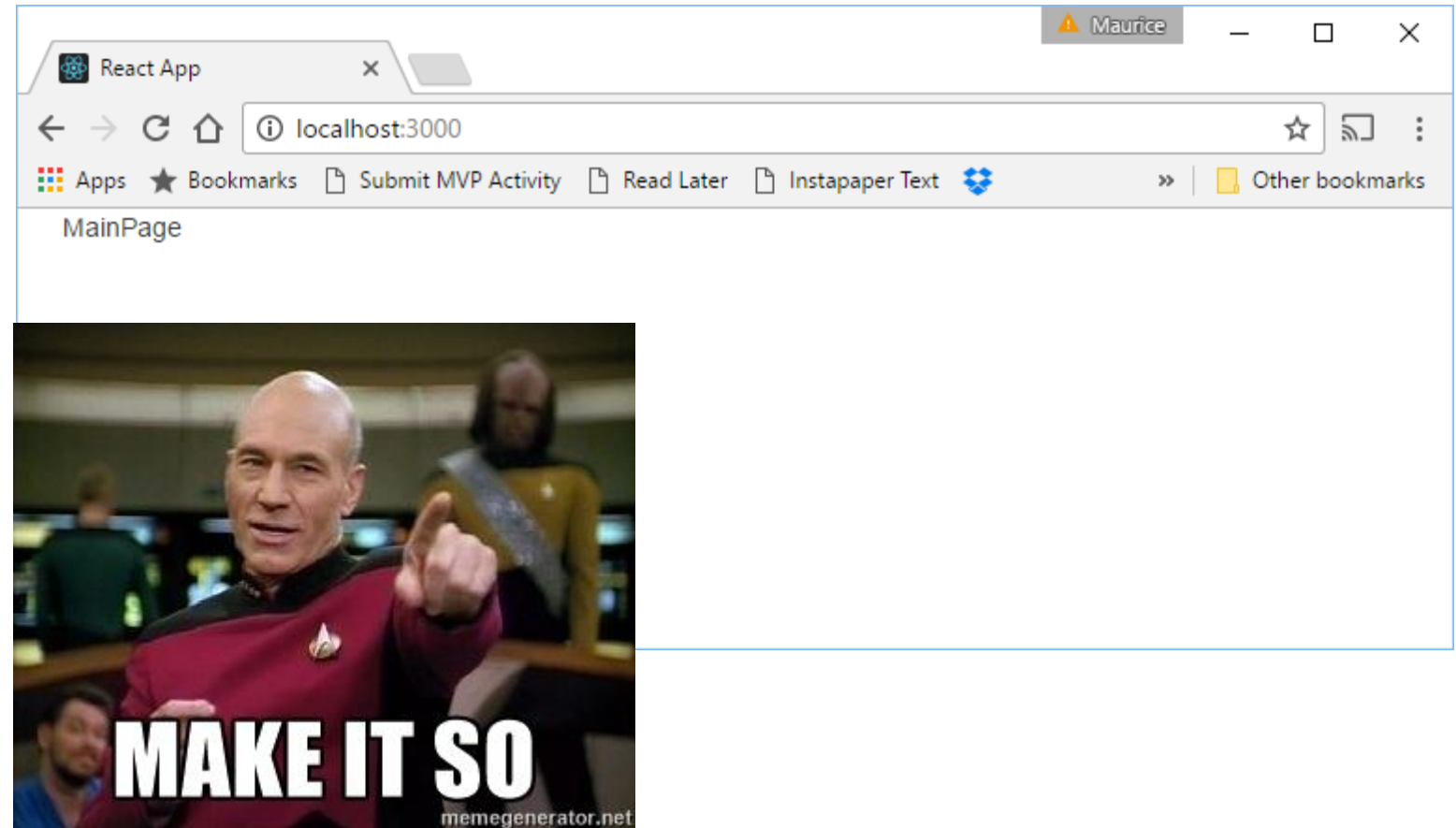


# Remember the user

# Remember me

```
4 class AppContainer extends Component {
5   constructor(props) {
6     super(props);
7
8     this.state = { user: null };
9     this.loginAsUser = this.loginAsUser.bind(this);
10  }
11
12  componentWillMount() {
13    if (localStorage.user) {
14      try {
15        const user = JSON.parse(localStorage.user);
16        this.setState({ user });
17      } catch (e) {
18        // eslint-disable-next-line no-console
19        console.error(e);
20      }
21    }
22  }
23
24  loginAsUser(user) {
25    if (user.rememberMe) {
26      localStorage.user = JSON.stringify(user);
27    }
28    this.setState({ user });
29  }
```

# Persisted Login



# Main Page Skeleton



# Main Page Skeleton

- Add three new components **Header**, **Billboard** and **GenreList** in the **main-page** folder.
  - Use the same component skeleton as before.

# MainPage

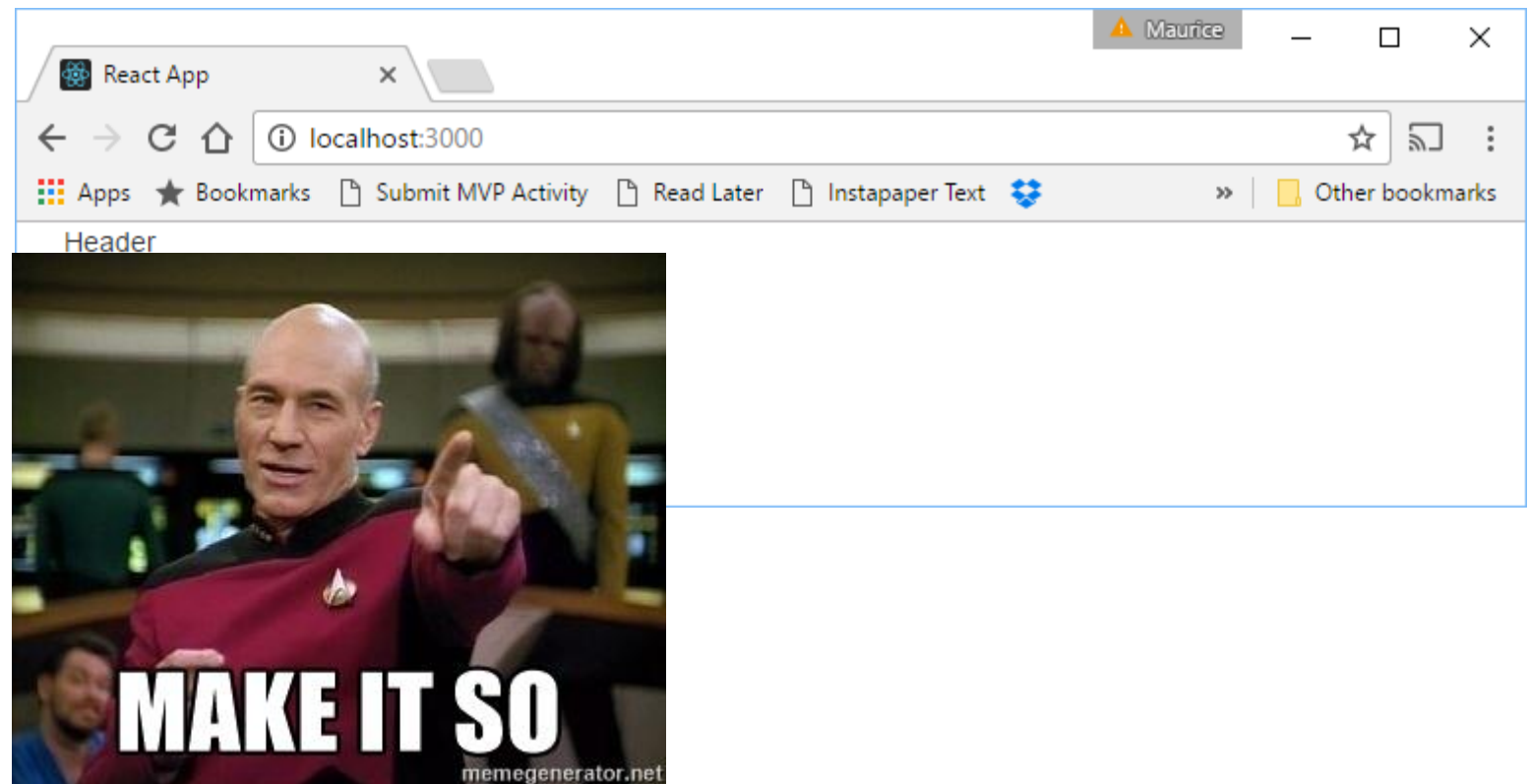
```
import React, { PropTypes } from 'react';
import Header from './header';
import Billboard from './billboard';
import GenreList from './genre-list';

const MainPage = ({ user, movies, startPlaying }) => (
  <div>
    <Header user={user} />
    <Billboard />
    <GenreList movies={movies} startPlaying={startPlaying} />
  </div>
);

MainPage.propTypes = {
  user: PropTypes.object.isRequired,
  movies: PropTypes.array.isRequired,
  startPlaying: PropTypes.func.isRequired,
};

export default MainPage;
```

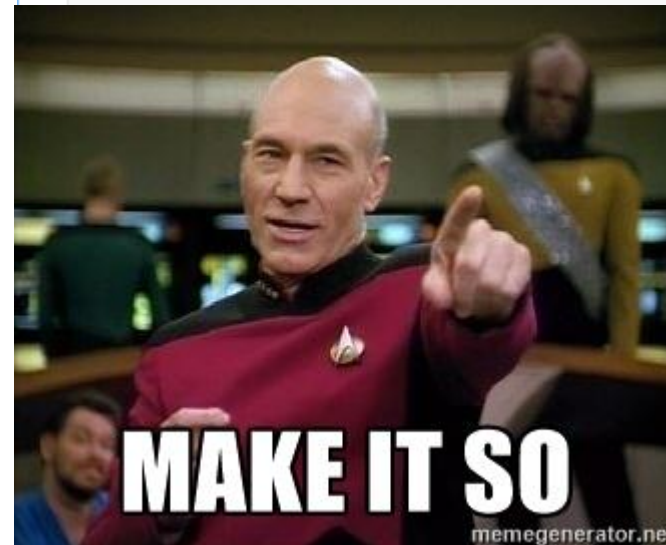
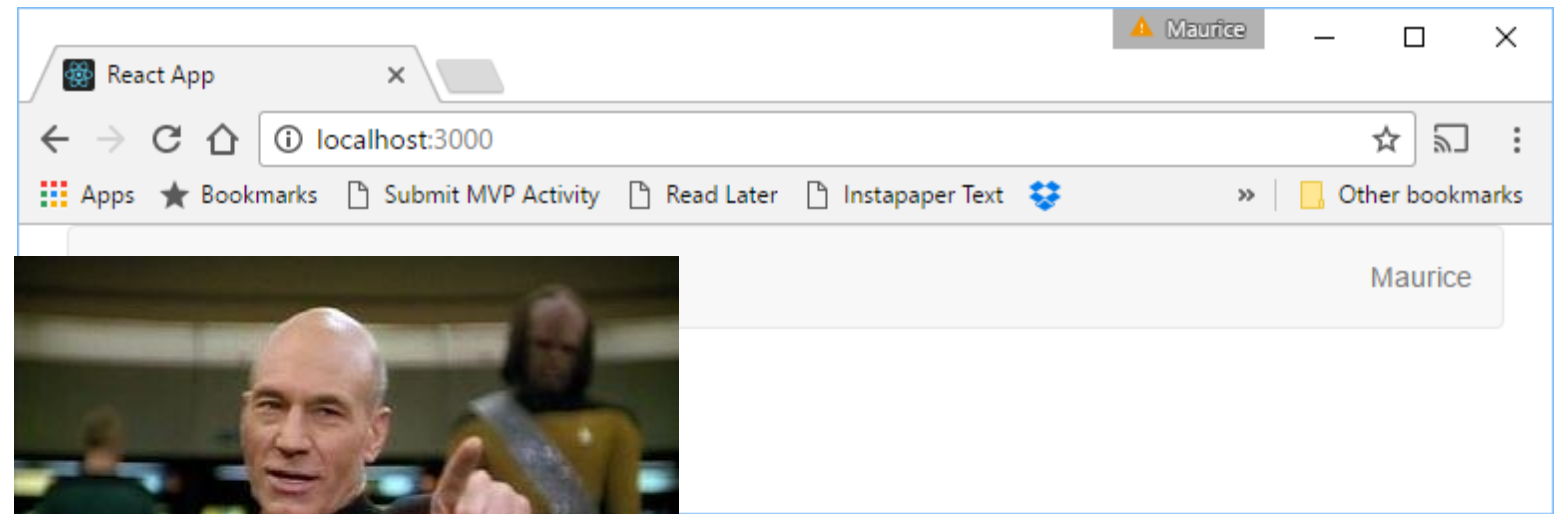
# MainPage Skeleton



# Adding the Header

# Header

```
1  import React, { PropTypes } from 'react';
2
3  const Header = ({ user }) => (
4    <nav className="navbar navbar-default">
5      <div className="container-fluid">
6        <div className="navbar-header">
7          <a className="navbar-brand" href="/">
8            Nitflex Movies
9          </a>
10        </div>
11
12        <div className="navbar-right">
13          <p className="navbar-text">
14            {user.name}
15          </p>
16        </div>
17      </div>
18    </nav>
19  );
20
21  Header.propTypes = {
22    user: PropTypes.shape({ name: React.PropTypes.string }).isRequired,
23  };
24
25  export default Header;
```



# Fetching AJAX data

# Movies Data

- Add [movies.json](#) to the public folder.



# AppContainer

```
4 class AppContainer extends Component {
5   constructor(props) {
6     super(props);
7
8     this.state = {
9       user: null,
10      movies: null,
11    };
12    this.loginAsUser = this.loginAsUser.bind(this);
13  }
14
15  componentWillMount() {
16    if (localStorage.user) {
17      try {
18        const user = JSON.parse(localStorage.user);
19        this.setState({ user });
20        this.fetchMovies();
21      } catch (e) {
22        // eslint-disable-next-line no-console
23        console.error(e);
24      }
25    }
26  }
27
28  fetchMovies() {
29    fetch('/movies.json')
30      .then(rsp => rsp.json())
31      .then(movies => this.setState({ movies }));
32  }
33
34  loginAsUser(user) {
35    if (user.rememberMe) {
36      localStorage.user = JSON.stringify(user);
37    }
38    this.setState({ user });
39    this.fetchMovies();
40  }
```

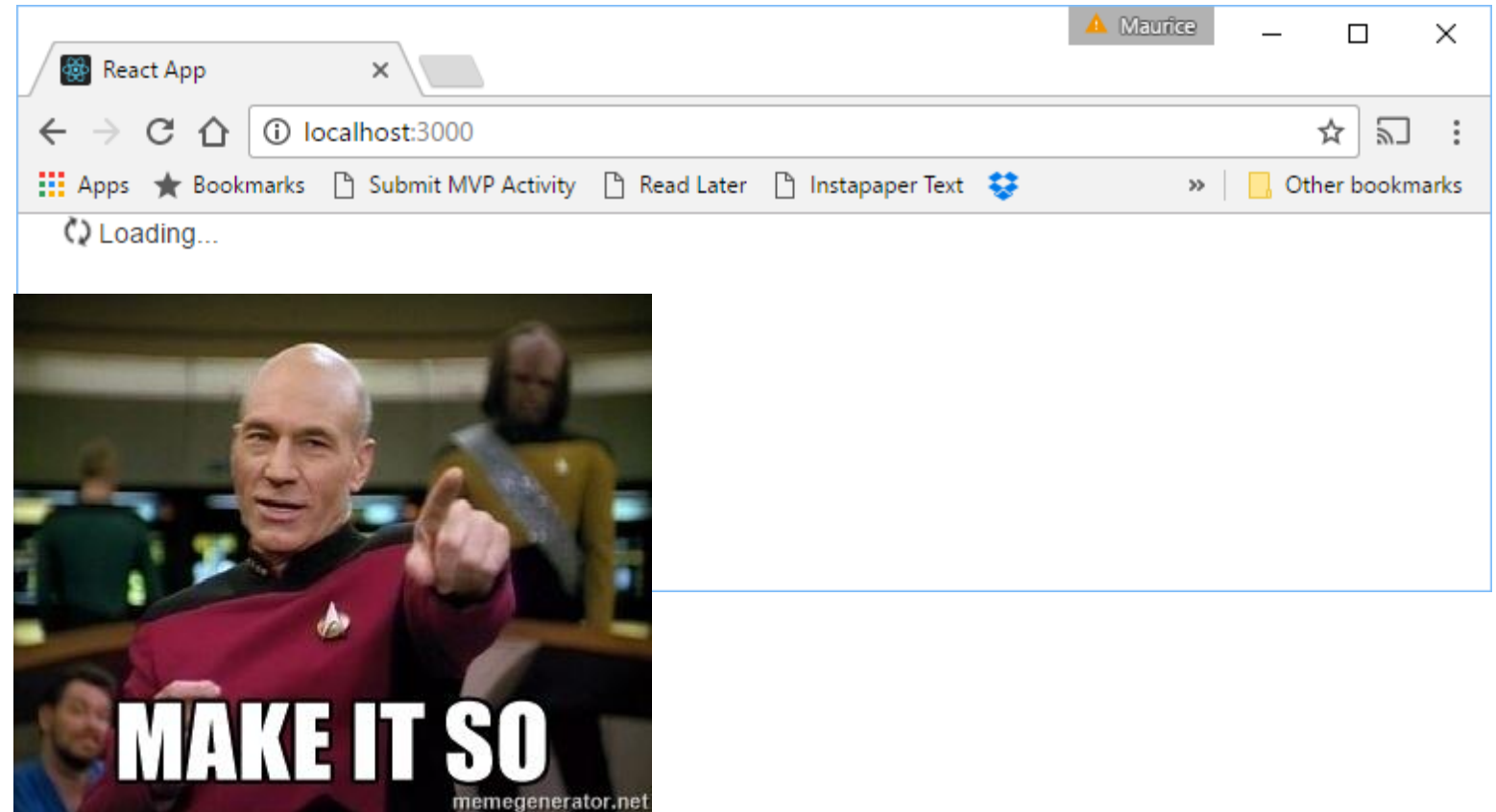
# AjaxLoading JSX

```
1  import React from 'react';  
2  import './ajax-loading.css';  
3  
4  const AjaxLoading = () => (  
5    <div>  
6      <i className="glyphicon glyphicon-refresh spinning" />  
7             Loading...  
8    </div>  
9  );  
10  
11 export default AjaxLoading;
```

# AjaxLoading CSS

```
1  .glyphicon.spinning {  
2      animation: spin 1s infinite linear;  
3  }  
4  
5  @keyframes spin {  
6      from { transform: scale(1) rotate(0deg); }  
7      to { transform: scale(1) rotate(360deg); }  
8  }
```

# AjaxLoading



# Adding the Billboard

# AppContainer

```
42   render() {  
43     const { user, movies } = this.state;  
44  
45     return (  
46       <AppPresentation  
47         user={user}  
48         loginAsUser={this.loginAsUser}  
49         movies={movies}  
50       />  
51     );  
52   }  
53 }
```

# MainPage

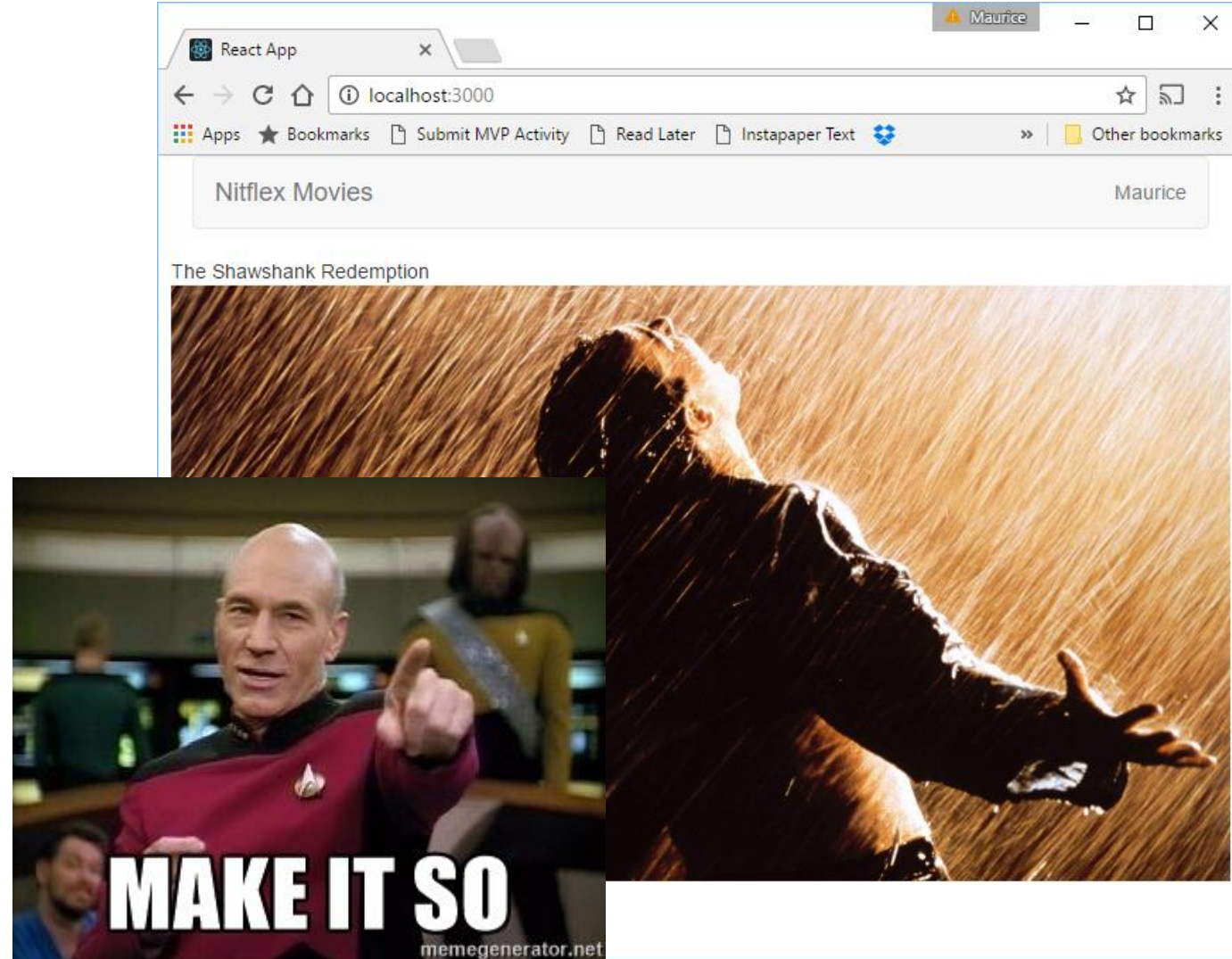
```
6  const MainPage = ({ user, movies, startPlaying }) => (  
7    <div>  
8      <Header user={user} />  
9      <Billboard movie={movies[0]} />  
10     <GenreList movies={movies} startPlaying={startPlaying} />  
11   </div>  
12 );  
13  
14 MainPage.propTypes = {  
15   user: PropTypes.object.isRequired,  
16   movies: PropTypes.array.isRequired,  
17   startPlaying: PropTypes.func.isRequired,  
18 };  
19  
20 export default MainPage;
```

# Billboard

```
1 | import React, { PropTypes } from 'react';
2 |
3 | - const Billboard = ({ movie }) => (
4 |   <div className="row">
5 |     <div className="title">
6 |       {movie.title}
7 |     </div>
8 |     <img
9 |       className="img-responsive"
10 |       alt={movie.title}
11 |       src={`http://image.tmdb.org/t/p/w1280/${movie.backdrop_path}`}
12 |     />
13 |   </div>
14 | );
15 |
16 | - Billboard.propTypes = {
17 |   movie: PropTypes.object.isRequired,
18 | };
19 |
20 | export default Billboard;
```



# Billboard



# A list of genres

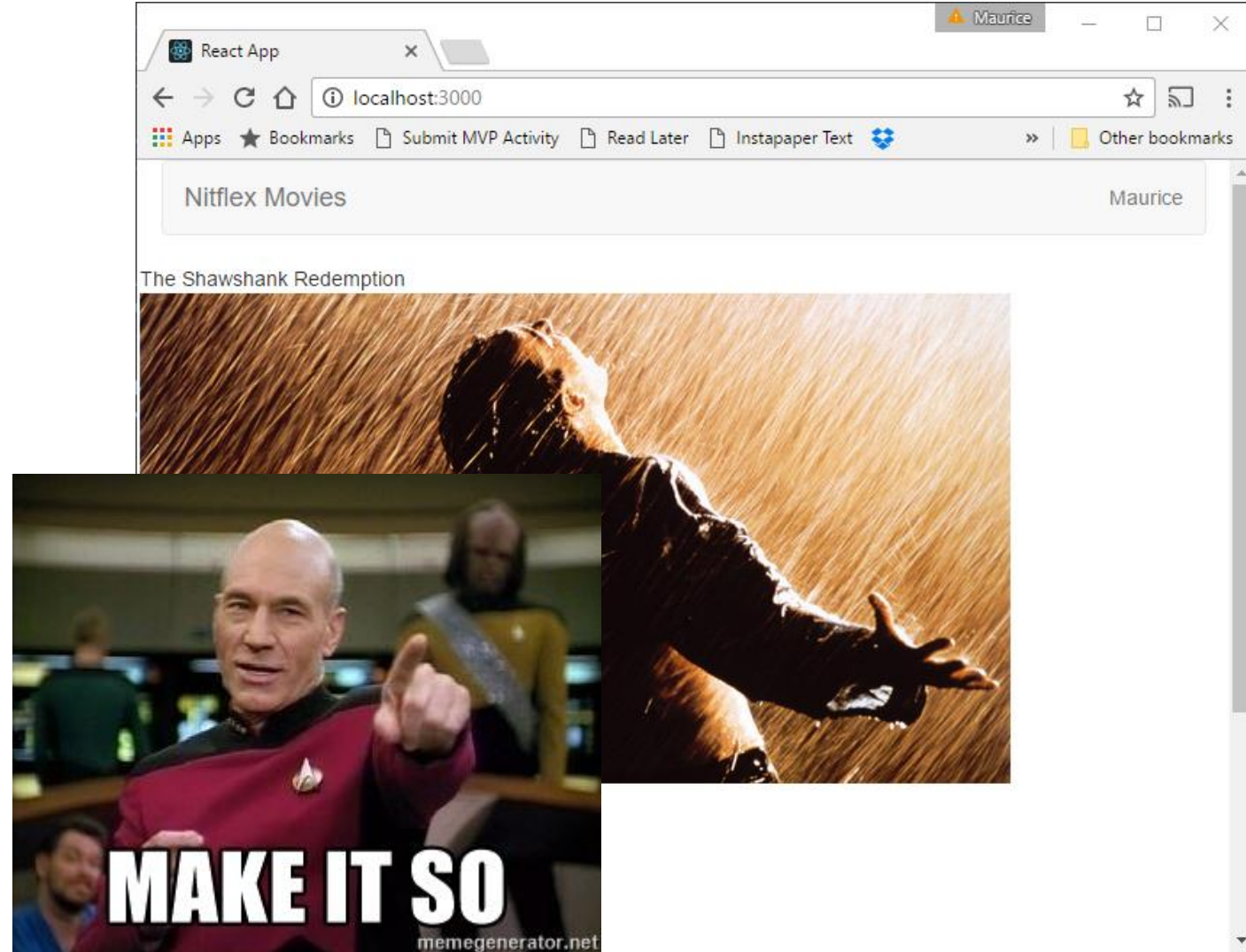
# GenreRow

- Add a GenreRow component to the main-page folder
  - Use the same component skeleton as before

# GenreList

```
1  import React, { PropTypes } from 'react';
2  import GenreRow from './genre-row';
3
4  const GenreList = ({ movies, startPlaying }) => {
5    const allGenres = {};
6
7    movies.forEach((movie) => {
8      movie.genre.forEach((genre) => {
9        if (allGenres[genre]) {
10          allGenres[genre].push(movie);
11        } else {
12          allGenres[genre] = [movie];
13        }
14      });
15    });
16
17    const genres = Object.keys(allGenres).sort();
18
19    return (
20      <div>
21        {genres.map(genre => <GenreRow
22          key={genre}
23          genre={genre}
24          movies={allGenres[genre]}
25          startPlaying={startPlaying}
26          />)}
27      </div>
28    );
29  };
30
31  GenreList.propTypes = {
32    movies: PropTypes.array.isRequired,
33    startPlaying: PropTypes.func.isRequired,
34  };
35
36  export default GenreList;
```

# GenreList





A genre is a list of  
movies

# GenreRowMovie

- Add a GenreRowMovie component to the main-page folder
  - Use the same component skeleton as before

# GenreRow JSX

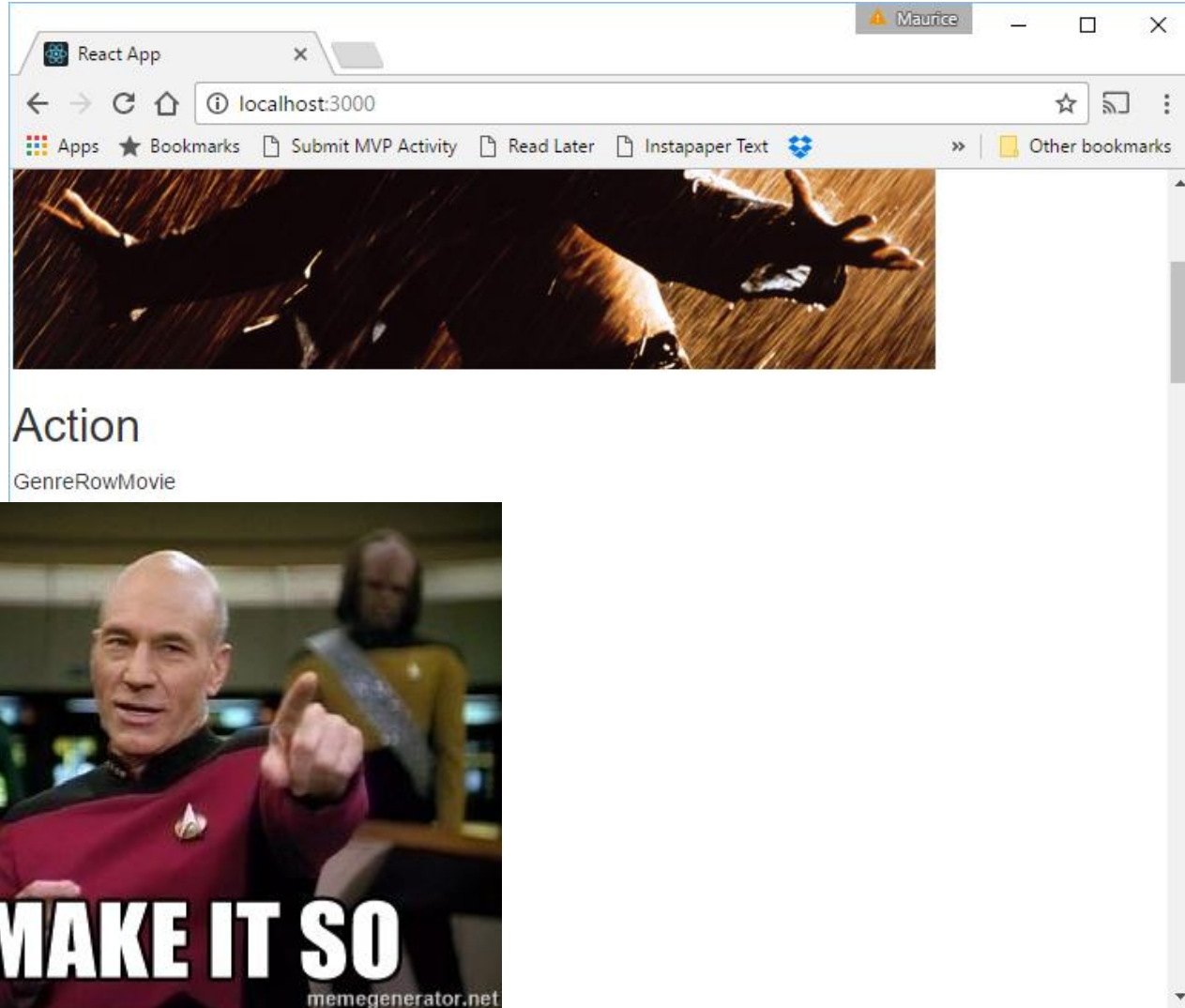
```
1  import React, { Component, PropTypes } from 'react';
2  import GenreRowMovie from './genre-row-movie';
3  import './genre-row.css';
4
5  class GenreRow extends Component {
6    render() {
7      const { genre, movies, startPlaying } = this.props;
8
9      return (
10       <div className="row genre-row">
11         <h2>{genre}</h2>
12
13         {movies.slice(0, 6).map(movie => (
14           <GenreRowMovie
15             key={movie.id}
16             movie={movie}
17             expandMovie={this.expandMovie}
18           />))}
19       </div>
20     );
21   }
22 }
23
24 GenreRow.propTypes = {
25   genre: PropTypes.string.isRequired,
26   movies: PropTypes.array.isRequired,
27   startPlaying: PropTypes.func.isRequired,
28 };
29
30 export default GenreRow;
```



# GenreRow CSS

```
1  .genre-row .genre-row-movie {  
2      padding: 0;  
3  }  
4  
5  .genre-row .genre-row-movie .title {  
6      overflow: hidden;  
7      text-overflow: ellipsis;  
8      white-space: nowrap;  
9  }
```

# GenreRow



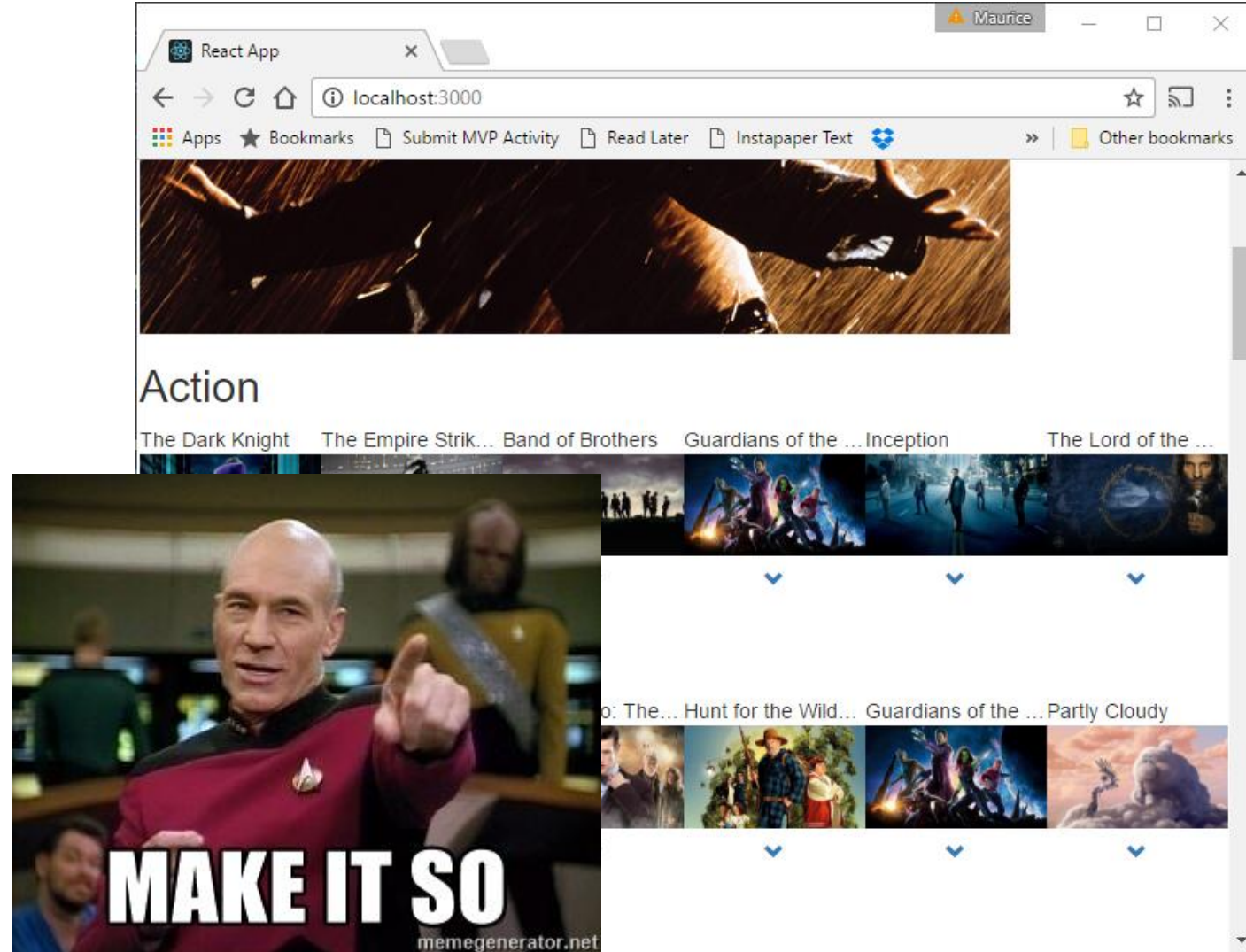


# Display each movie

# GenreRowMovie

```
1  import React, { Component, PropTypes } from 'react';
2
3  class GenreRowMovie extends Component {
4    constructor(props) {
5      super(props);
6    }
7
8    render() {
9      const { movie } = this.props;
10
11      return (
12        <div className="col-sm-2 genre-row-movie">
13          <div className="title">
14            {movie.title}
15          </div>
16          <img
17            className="img-responsive"
18            alt={movie.title}
19            src={`http://image.tmdb.org/t/p/w300/${movie.backdrop_path}`}
20          />
21          <button className="btn btn-link btn-block expand" onClick={this.expandMovie}>
22            <i className="glyphicon glyphicon-chevron-down" />
23          </button>
24        </div>
25      );
26    }
27  }
28
29  GenreRowMovie.propTypes = {
30    movie: PropTypes.object.isRequired,
31  };
32
33  export default GenreRowMovie;
```

# GenreRowMovie



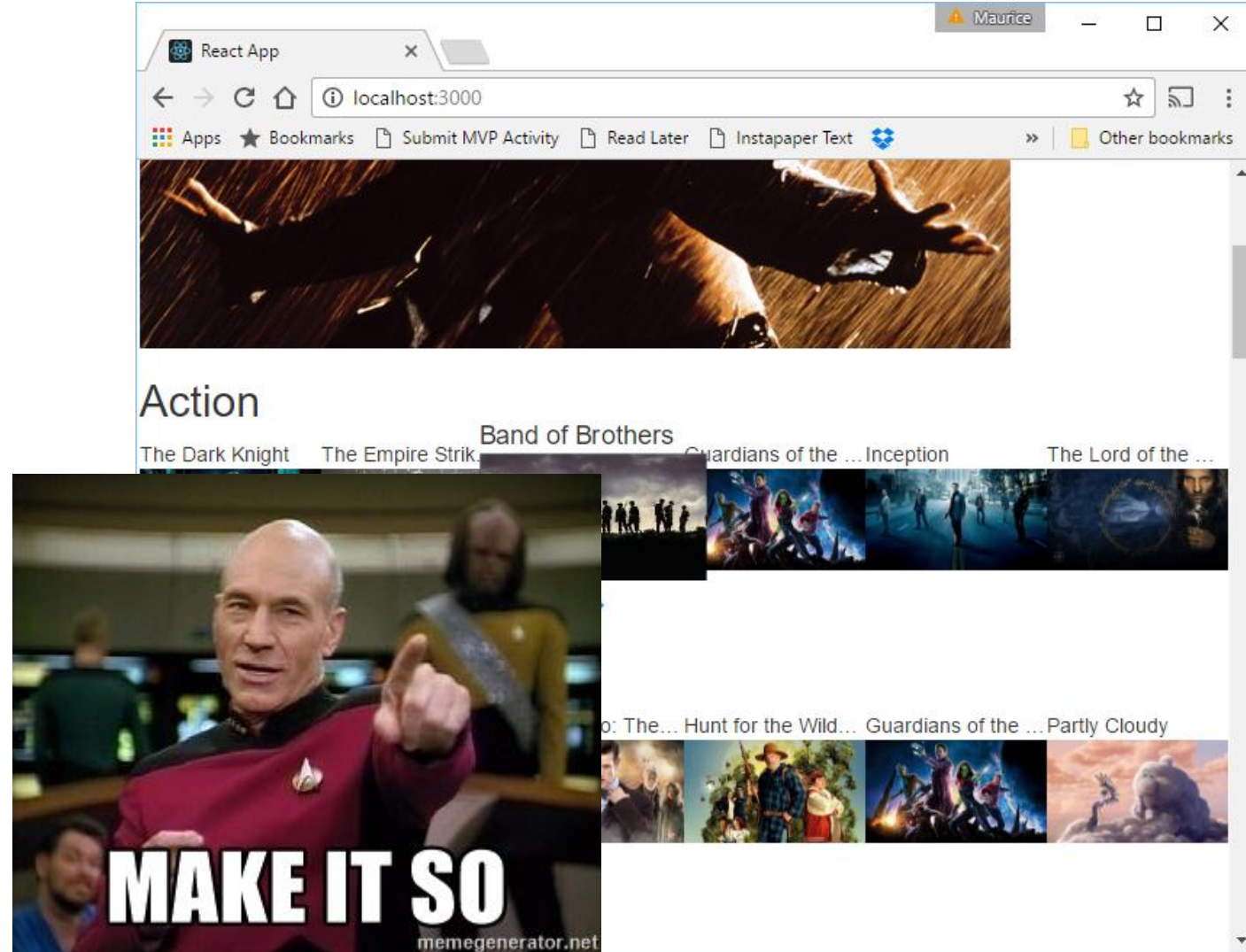


# Highlight a movie

# GenreRow CSS

```
1  .genre-row .genre-row-movie {
2      padding: 0;
3      transition: transform 0.3s;
4      transition-delay: 0s;
5  }
6
7  .genre-row .genre-row-movie:hover {
8      transform: scale(1.25);
9      transition-delay: 0.3s;
10     z-index: 1;
11 }
12
13 .genre-row .genre-row-movie .title {
14     overflow: hidden;
15     text-overflow: ellipsis;
16     white-space: nowrap;
17 }
18
19 .genre-row .genre-row-movie .expand {
20     transition: all 0.3s;
21     text-align: center;
22     opacity: 0;
23 }
24
25 .genre-row .genre-row-movie:hover .expand {
26     transition-delay: 0.3s;
27     opacity: 1;
28 }
29
30 .genre-row .genre-row-movie .btn:focus {
31     outline-style: none !important;
32 }
```

# Transitions





# Expand a movie

## ExpandedGenreRowMovie

- Add a ExpandedGenreRowMovie component to the main-page folder
  - Use the same component skeleton as before

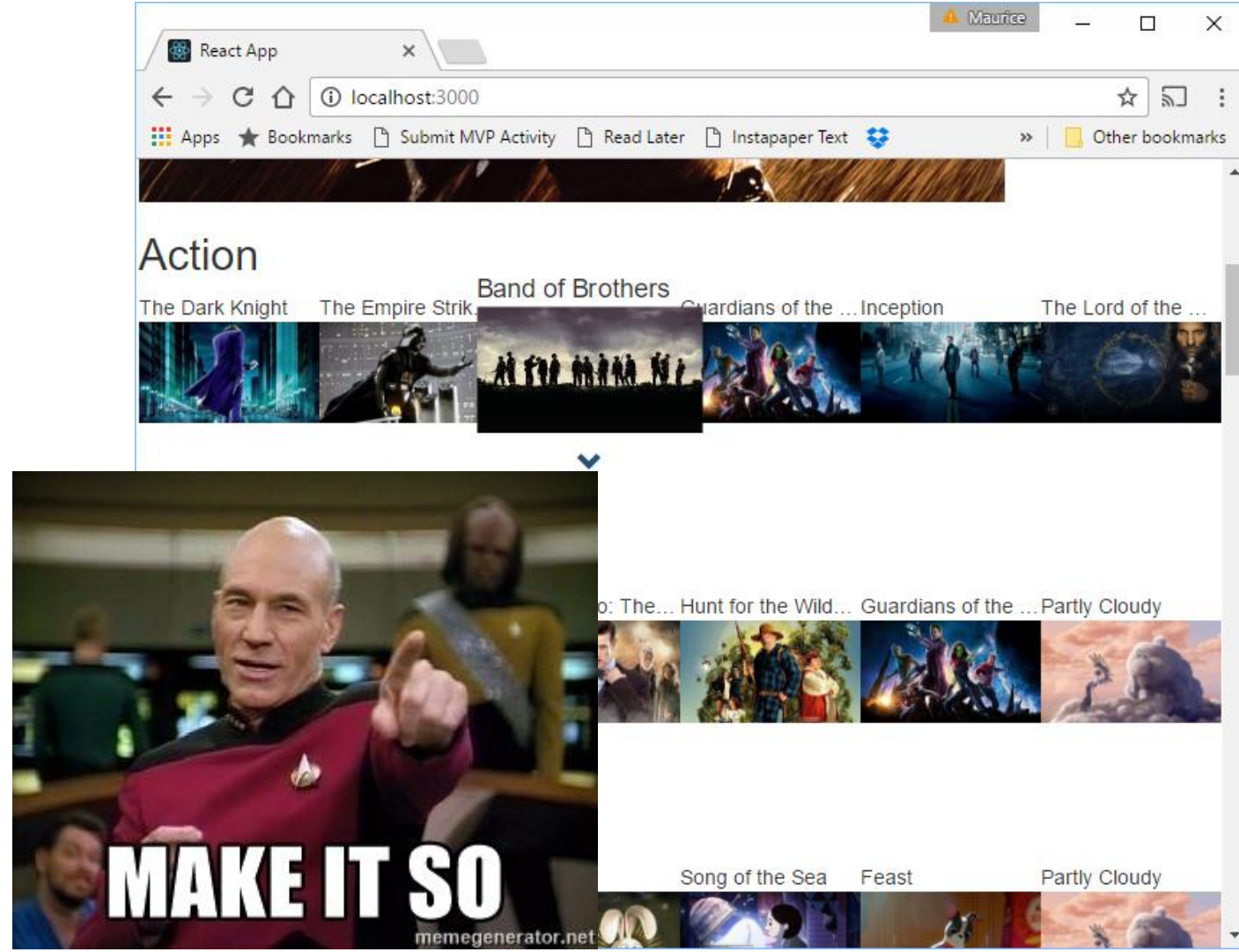
# GenreRow

```
5
6 class GenreRow extends Component {
7   constructor(props) {
8     super(props);
9
10    this.state = {
11      expandedMovie: null,
12    };
13
14    this.expandMovie = this.expandMovie.bind(this);
15  }
16
17  expandMovie(movie) {
18    this.setState({ expandedMovie: movie });
19  }
20
21  render() {
22    const { genre, movies, startPlaying } = this.props;
23    const { expandedMovie } = this.state;
24
25    return (
26      <div className="row genre-row">
27        <h2>{genre}</h2>
28
29        {movies.slice(0, 6).map(movie => (
30          <GenreRowMovie
31            key={movie.id}
32            movie={movie}
33            expandMovie={this.expandMovie}
34          />))}
35
36        {expandedMovie
37          ? <ExpandedGenreRowMovie movie={expandedMovie} startPlaying={startPlaying} />
38          : <div />}
39      </div>
40    );
41  }
42 }
```

# GenreRowMovie

```
1  import React, { Component, PropTypes } from 'react';
2
3  class GenreRowMovie extends Component {
4    constructor(props) {
5      super(props);
6
7      this.expandMovie = this.expandMovie.bind(this);
8    }
9
10   expandMovie() {
11     this.props.expandMovie(this.props.movie);
12   }
13
14   render() {
15     const { movie } = this.props;
16
17     return (
18       <div className="col-sm-2 genre-row-movie">
19         <div className="title">
20           {movie.title}
21         </div>
22         <img
23           className="img-responsive"
24           alt={movie.title}
25           src={`http://image.tmdb.org/t/p/w300/${movie.backdrop_path}`}
26         />
27         <button className="btn btn-link btn-block expand" onClick={this.expandMovie}>
28           <i className="glyphicon glyphicon-chevron-down" />
29         </button>
30       </div>
31     );
32   }
33 }
34
35 GenreRowMovie.propTypes = {
36   movie: PropTypes.object.isRequired,
37   expandMovie: PropTypes.func.isRequired,
38 };
39
40 export default GenreRowMovie;
```

## ExpandedGenreRowMovie



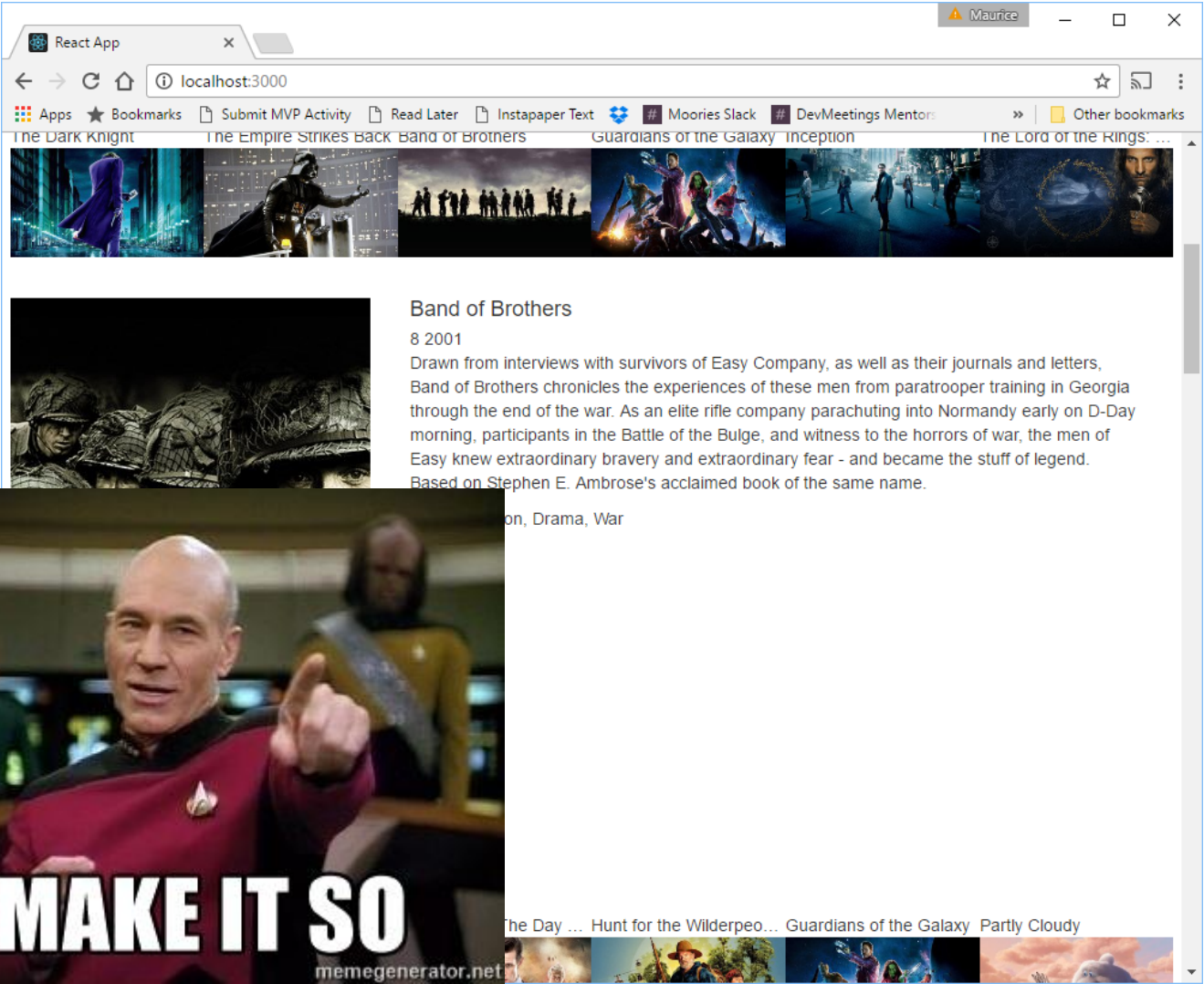


# Show movie details

## ExpandedGenreRowMovie

```
3 class ExpandedGenreRowMovie extends Component {
4   constructor(props) {
5     super(props);
6     this.startPlaying = this.startPlaying.bind(this);
7   }
8
9   startPlaying() {
10    const { movie, startPlaying } = this.props;
11    startPlaying(movie);
12  }
13
14  render() {
15    const { movie } = this.props;
16
17    return (
18      <div className="col-sm-12" style={{paddingLeft: 0}} >
19        <div className="col-sm-4" style={{paddingLeft: 0}} >
20          <img src={`http://image.tmdb.org/t/p/w300/${movie.poster_path}`} alt={movie.title} />
21        </div>
22        <div className="col-sm-8">
23          <h4 className="media-heading">{movie.title}</h4>
24          <div>
25            <span>
26              {movie.vote_average}
27              <span>&nbsp;</span>
28              {movie.release_date.slice(0, 4)}
29            </span>
30          </div>
31          <p>{movie.overview}</p>
32          <div>
33            <span>Genres:&nbsp;</span>
34            <span>{movie.genre.join(', ')}</span>
35          </div>
36          <button className="btn btn-default btn-lg" onClick={this.startPlaying}>
37            <i className="glyphicon glyphicon-play-circle" />
38          </button>
39        </div>
40      </div>
41    );
42  }
43 }
```

ExpandedGenreRowMovie



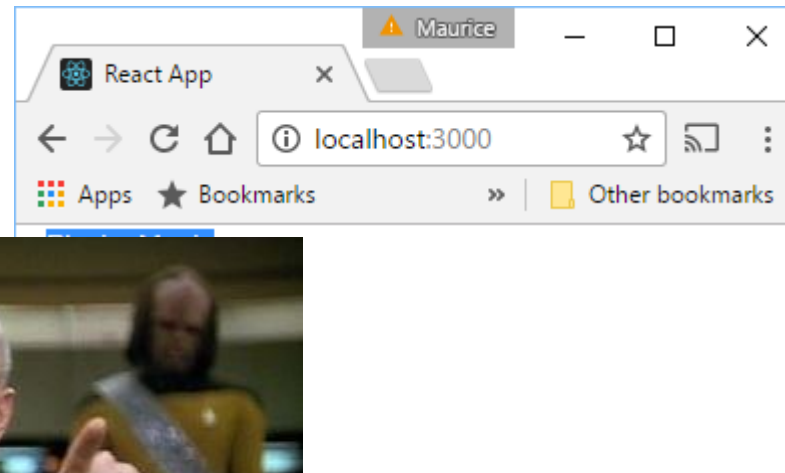


# Playing a movie

# AppContainer

```
29   fetchMovies() {
30     fetch('/movies.json')
31       .then(rsp => rsp.json())
32       .then(movies => this.setState({ movies }));
33   }
34
35   startPlaying(movie) {
36     this.setState({ playing: movie });
37   }
38
39   loginAsUser(user) {
40     if (user.rememberMe) {
41       localStorage.user = JSON.stringify(user);
42     }
43     this.setState({ user });
44     this.fetchMovies();
45   }
46
47   render() {
48     const { user, movies, playing } = this.state;
49
50     return (
51       <AppPresentation
52         user={user}
53         loginAsUser={this.loginAsUser}
54         movies={movies}
55         startPlaying={this.startPlaying}
56         playing={playing}
57       />
58     );
59   }
60 }
```

Playing  
a movie





# Play the movie

# PlayingMovie JSX

```
1  import React, { PropTypes } from 'react';
2  import './playing-movie.css';
3
4  const PlayingMovie = ({ movie, stopPlaying }) => (
5    <div className="row">
6      <div className="title">
7        <h1>{movie.title}</h1>
8        <button className="btn btn-link pull-right" onClick={stopPlaying}>
9          <i className="glyphicon glyphicon-remove" />
10        </button>
11      </div>
12      <iframe
13        src="https://www.youtube.com/embed/6hB3S9bIaco?autoplay=1"
14        frameborder="0"
15        allowFullScreen
16        className="video"
17      />
18    </div>
19  </div>
20  );
21
22  PlayingMovie.propTypes = {
23    movie: PropTypes.object.isRequired,
24    stopPlaying: PropTypes.func.isRequired,
25  };
26
27  export default PlayingMovie;
```

# PlayingMovie CSS

```
1  .video {  
2    position: absolute;  
3    top: 20%;  
4    left: 0;  
5    width: 100%;  
6    height: 80%;  
7 }
```

# PlayingMovie





# Stop a playing movie



# AppContainer

```
35
36   startPlaying(movie) {
37     this.setState({ playing: movie });
38   }
39
40   stopPlaying() {
41     this.setState({ playing: null });
42   }
43
44   loginAsUser(user) {
45     if (user.rememberMe) {
46       localStorage.user = JSON.stringify(user);
47     }
48     this.setState({ user });
49     this.fetchMovies();
50   }
51
52   render() {
53     const { user, movies, playing } = this.state;
54
55     return (
56       <AppPresentation
57         user={user}
58         loginAsUser={this.loginAsUser}
59         movies={movies}
60         startPlaying={this.startPlaying}
61         playing={playing}
62         stopPlaying={this.stopPlaying}
63       />
64     );
65   }
66 }
```

Stop playing



# Filter movies user interface

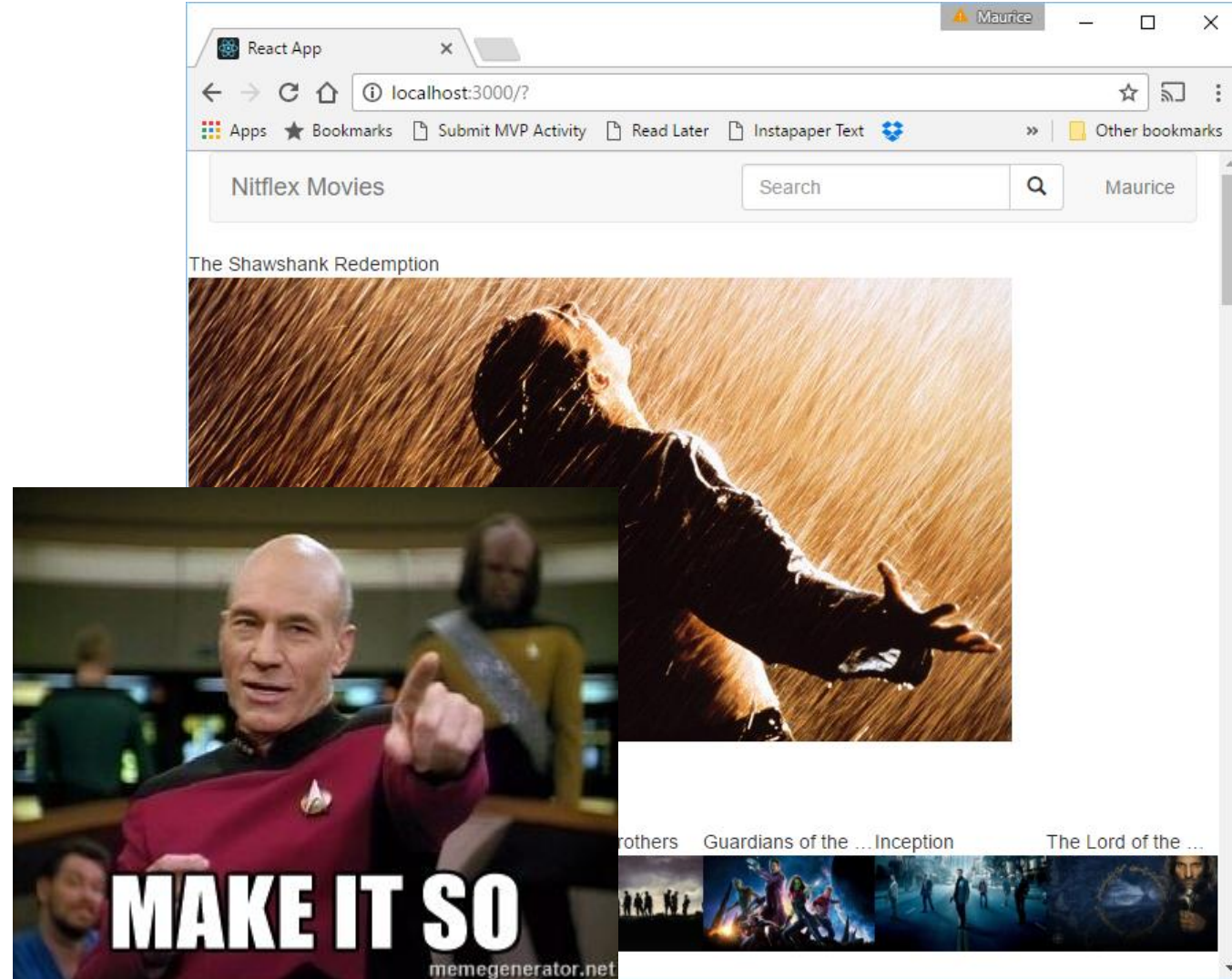
# FilterMovies

```
1  import React, { Component, PropTypes } from 'react';
2
3  class FilterMovies extends Component {
4    constructor(props) {
5      super(props);
6
7      this.state = {
8        search: '',
9      };
10
11      this.onSearchChange = this.onSearchChange.bind(this);
12    }
13
14    onSearchChange(e) {
15      const search = e.target.value;
16      this.setState({ search });
17    }
18
19    render() {
20      const { search } = this.state;
21
22      return (
23        <form className="navbar-form navbar-right">
24          <div className="input-group">
25            <input
26              type="text"
27              value={search}
28              onChange={this.onSearchChange}
29              className="form-control"
30              placeholder="Search"
31            />
32            <span className="input-group-btn">
33              <button className="btn btn-default">
34                <i className="glyphicon glyphicon-search" />
35              </button>
36            </span>
37          </div>
38        </form>
39      );
40    }
41  }
42
43  export default FilterMovies;
```

# Header

```
1  import React, { PropTypes } from 'react';
2  import FilterMovies from './filter-movies';
3
4  const Header = ({ user }) => (
5    <nav className="navbar navbar-default">
6      <div className="container-fluid">
7        <div className="navbar-header">
8          <a className="navbar-brand" href="/">
9            Nitflex Movies
10          </a>
11        </div>
12
13        <div className="navbar-right">
14          <p className="navbar-text">
15            {user.name}
16          </p>
17        </div>
18      <FilterMovies />
19    </div>
20  </nav>
21  );
22
23  Header.propTypes = {
24    user: PropTypes.shape({ name: React.PropTypes.string }).isRequired,
25  };
26
27  export default Header;
```

# Filter Movies UI



# Filter movies state

# FilterMovies

```
11   this.onSearchChange = this.onSearchChange.bind(this);
12   this.filterMovies = this.filterMovies.bind(this);
13 }
14
15 onSearchChange(e) {
16   const search = e.target.value;
17   this.setState({ search });
18 }
19
20 filterMovies(e) {
21   e.preventDefault();
22   this.props.filterMovies(this.state.search);
23 }
24
25 render() {
26   const { search } = this.state;
27
28   return (
29     <form className="navbar-form navbar-right">
30       <div className="input-group">
31         <input
32           type="text"
33           value={search}
34           onChange={this.onSearchChange}
35           className="form-control"
36           placeholder="Search"
37         />
38         <span className="input-group-btn">
39           <button className="btn btn-default" onClick={this.filterMovies}>
40             <i className="glyphicon glyphicon-search" />
41           </button>
42         </span>
43       </div>
44     </form>
45   );
46 }
47
48 FilterMovies.propTypes = {
49   filterMovies: PropTypes.func.isRequired,
50 };
51
52 export default FilterMovies;
```



# Header

```
1  import React, { PropTypes } from 'react';
2  import FilterMovies from './filter-movies';
3
4  const Header = ({ user, filterMovies }) => (
5    <nav className="navbar navbar-default">
6      <div className="container-fluid">
7        <div className="navbar-header">
8          <a className="navbar-brand" href="/">
9            Nitflex Movies
10          </a>
11        </div>
12
13        <div className="navbar-right">
14          <p className="navbar-text">
15            {user.name}
16          </p>
17        </div>
18        <FilterMovies filterMovies={filterMovies} />
19      </div>
20    </nav>
21  );
22
23  Header.propTypes = {
24    user: PropTypes.shape({ name: React.PropTypes.string }).isRequired,
25    filterMovies: PropTypes.func.isRequired,
26  };
27
28  export default Header;
```

# AppContainer

```
54
55   filterMovies(search) {
56     let filteredMovies = null;
57
58     if (search) {
59       const searchLower = search.toLowerCase();
60       const { allMovies } = this.state;
61       filteredMovies = allMovies.filter(m => m.title.toLowerCase().indexOf(searchLower) !== -1);
62
63       if (!filteredMovies.length) {
64         filteredMovies = null;
65       }
66     }
67
68     this.setState({ filteredMovies });
69   }
70
71   render() {
72     const { user, allMovies, filteredMovies, playing } = this.state;
73
74     return (
75       <AppPresentation
76         user={user}
77         loginAsUser={this.loginAsUser}
78         movies={filteredMovies || allMovies}
79         startPlaying={this.startPlaying}
80         playing={playing}
81         stopPlaying={this.stopPlaying}
82         filterMovies={this.filterMovies}
83       />
84     );
85   }
86 }
87
88 export default AppContainer;
```

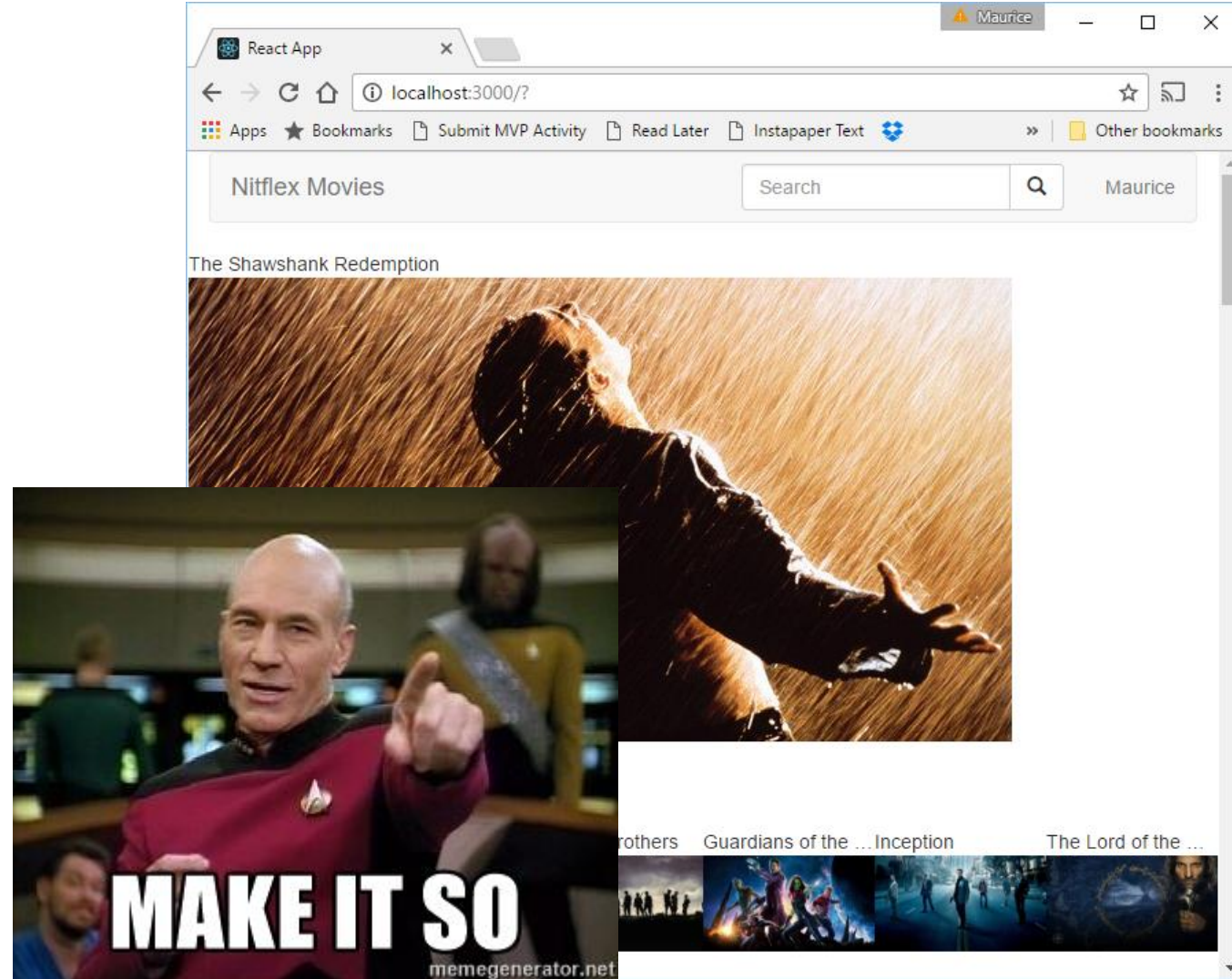
# AppPresentation

```
1 import React, { PropTypes } from 'react';
2 import LoginPage from './login-page';
3 import PlayingMovie from './playing-movie';
4 import MainPage from './main-page';
5 import AjaxLoading from './utils/ajax-loading';
6
7 const AppPresentation = ({ user, movies, playing, loginAsUser, startPlaying, stopPlaying, filterMovies }) => {
8   let component = null;
9
10  if (!user) {
11    component = <LoginPage loginAsUser={loginAsUser} />;
12  } else if (!movies) {
13    component = <AjaxLoading />;
14  } else if (playing) {
15    component = <PlayingMovie movie={playing} stopPlaying={stopPlaying} />;
16  } else {
17    component = <MainPage user={user} movies={movies} startPlaying={startPlaying} filterMovies={filterMovies} />;
18  }
19
20  return (
21    <div className="container">
22      {component}
23    </div>
24  );
25 };
26
27 AppPresentation.propTypes = {
28   user: PropTypes.object,
29   movies: PropTypes.array,
30   playing: PropTypes.object,
31   loginAsUser: PropTypes.func.isRequired,
32   startPlaying: PropTypes.func.isRequired,
33   stopPlaying: PropTypes.func.isRequired,
34   filterMovies: PropTypes.func.isRequired,
35 };
36
37 export default AppPresentation;
```

# MainPage

```
1  import React, { PropTypes } from 'react';
2  import Header from './header';
3  import Billboard from './billboard';
4  import GenreList from './genre-list';
5
6  const MainPage = ({ user, movies, startPlaying, filterMovies }) => (
7    <div>
8      <Header user={user} filterMovies={filterMovies} />
9      <Billboard movie={movies[0]} />
10     <GenreList movies={movies} startPlaying={startPlaying} />
11   </div>
12 );
13
14 MainPage.propTypes = {
15   user: PropTypes.object.isRequired,
16   movies: PropTypes.array.isRequired,
17   startPlaying: PropTypes.func.isRequired,
18   filterMovies: PropTypes.func.isRequired,
19 };
20
21 export default MainPage;
```

# Filter Movies State



# Bonus Exercises

- Add a **logout** button to the **Header**.
- Add a **play** button to the **Billboard** that appears on mouse hover.
- Extract **form-group** from **LoginPage**.
- Split **LoginPage** into container and presentation components.
- Add horizontal scrolling to the **GenreRow**.
- Use **React-Router** and different paths for the main components.

Maurice de Beijer

@mauricedb

maurice.de.beijer  
@gmail.com

