REACT NATIVE

React Navigation, Flatlist

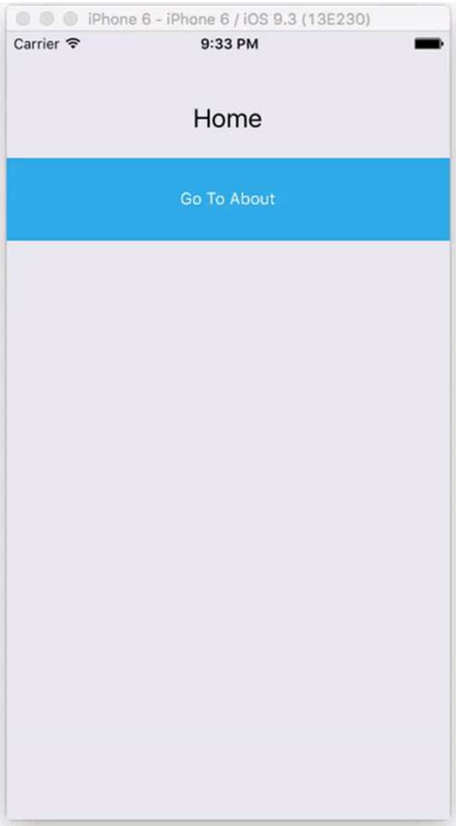
React Navigation

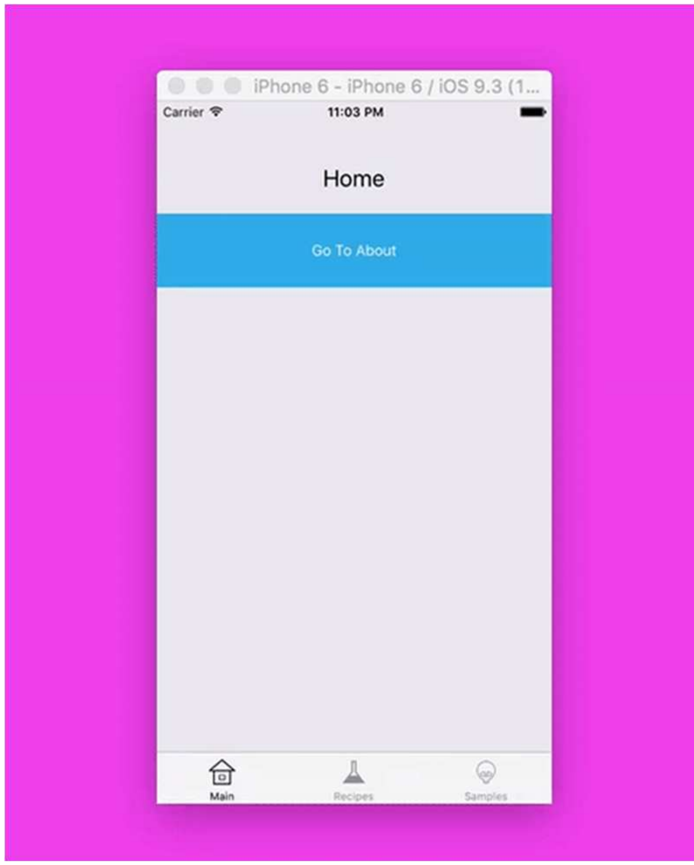
- https://reactnative.dev/docs/navigation
- https://docs.expo.dev/guides/routing-and-navigation

- 3 types:
  - Stack
  - Tab
  - Drawer

```
"@react-navigation/native": "6.0.8",
"@react-navigation/native-stack": "6.5.0",
"@react-navigation/bottom-tabs": "6.2.0",
"@react-navigation/drawer": "6.3.1",
```

HIENLTH





Stack Navigation

- import { NavigationContainer } from '@react-navigation/native';
- import { createNativeStackNavigator } from '@react-navigation/native-stack';

```
const Stack = createNativeStackNavigator();
  <NavigationContainer>
   <Stack.Navigator>
    <Stack.Screen
     name="Home"
     component={HomeScreen}
     options={{ title: 'Welcome HOME' }}
    />
    <Stack.Screen name="Profile" component={ProfileScreen} />
   </Stack.Navigator>
  </NavigationContainer>
```

HIENLTH

Tab Navigation

- import { NavigationContainer } from '@react-navigation/native';
- import {createBottomTabNavigator} from '@react-navigation/bottom-tabs';

```
const Tab = createBottomTabNavigator();
  <NavigationContainer>
   <Tab.Navigator>
    <Tab.Screen
     name="Home"
     component={HomeScreen}
     options={{ title: 'Welcome HOME' }}
    />
    <Tab.Screen name="Profile" component={ProfileScreen} />
   </Tab.Navigator>
  </NavigationContainer>
```

HIENLTH

Drawer Navigation

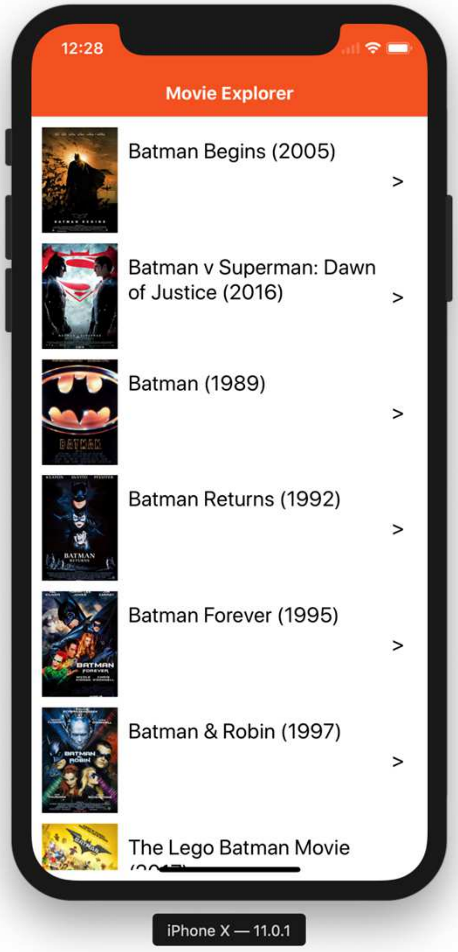
- import { NavigationContainer } from '@react-navigation/native';
- import {createDrawerNavigator} from '@react-navigation/drawer';
- const Drawer = createDrawerNavigator();

```
<NavigationContainer>
  <Drawer.Navigator initialRouteName="Home">
    <Drawer.Screen name="Home" component={HomeScreen} />
    <Drawer.Screen name="Profile" component={ProfileScreen} />
    <Drawer.Screen name="Search" component={Search} />
  </Drawer.Navigator>
</NavigationContainer>
```

HIENLTH

Movie Explorer App

```
react-native init lesson_6
```


12:28

Movie Explorer

Batman Begins (2005) >

Batman v Superman: Dawn of Justice (2016) >

Batman (1989) >

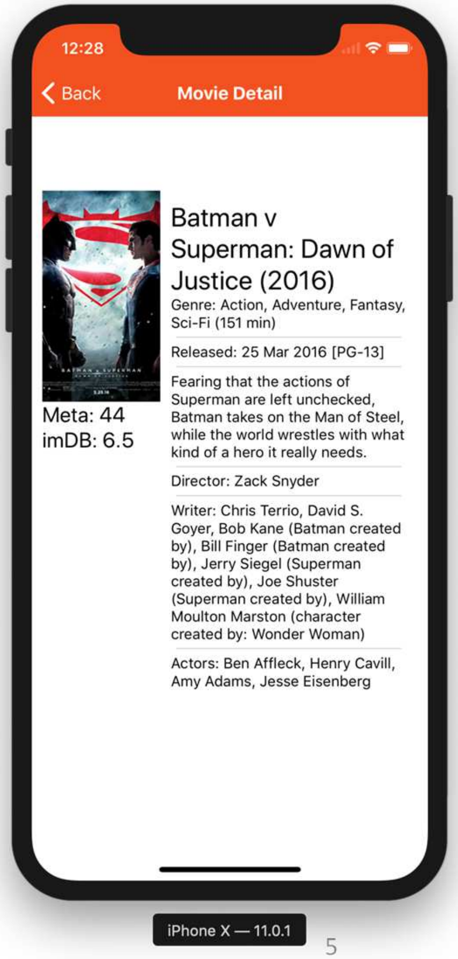
Batman Returns (1992) >

Batman Forever (1995) >

Batman & Robin (1997) >

The Lego Batman Movie

iPhone X — 11.0.1


12:28

< Back

Movie Detail

Meta: 44
imDB: 6.5

Batman v Superman: Dawn of Justice (2016)

Genre: Action, Adventure, Fantasy, Sci-Fi (151 min)

Released: 25 Mar 2016 [PG-13]

Fearing that the actions of Superman are left unchecked, Batman takes on the Man of Steel, while the world wrestles with what kind of a hero it really needs.

Director: Zack Snyder

Writer: Chris Terrio, David S. Goyer, Bob Kane (Batman created by), Bill Finger (Batman created by), Jerry Siegel (Superman created by), Joe Shuster (Superman created by), William Moulton Marston (character created by: Wonder Woman)

Actors: Ben Affleck, Henry Cavill, Amy Adams, Jesse Eisenberg

iPhone X — 11.0.1

React Navigation

- React Navigation’s stack provide a wait for app to transition between screens and manage navigation history
- If app uses only one stack navigator then it is conceptually similar to how a web browser handles navigation state
- React Navigation's stack navigator provides the gestures and animations that you would expect on Android and iOS when navigating between routes in the stack
- To setup react native navigation, you must use createStackNavigator which takes a route configuration object and, optionally, an options object
- For more information: https://reactnavigation.org/en/

HIENLTH

Install React Navigation

- Install the react-navigation package in your React Native project.

```
yarn add react-navigation
# or with npm
# npm install react-navigation
```

- Install react-native-gesture-handler. If you’re using the Expo managed workflow then you don’t need to do anything here, it’s included in the SDK

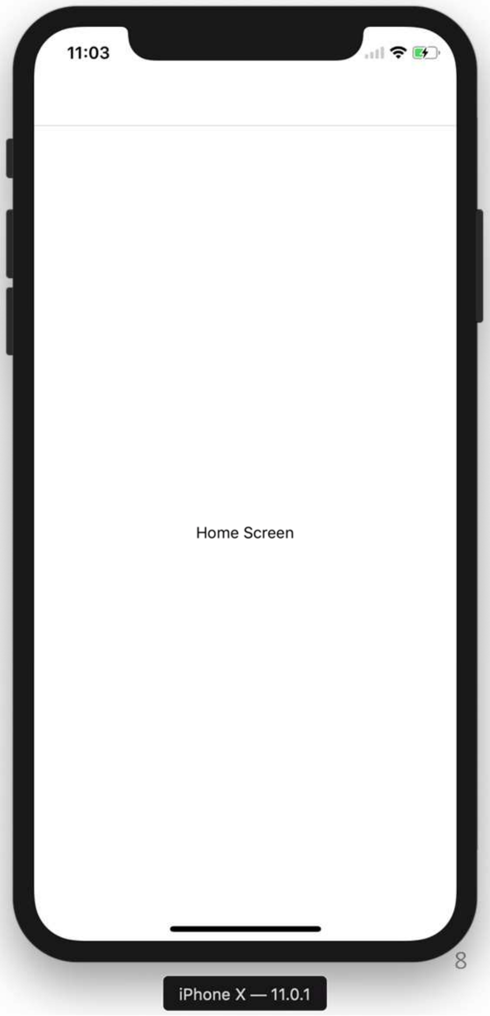
```
yarn add react-native-gesture-handler
# or with npm
# npm install --save react-native-gesture-handler
```

- Link all native dependencies

```
react-native link react-native-gesture-handler
```

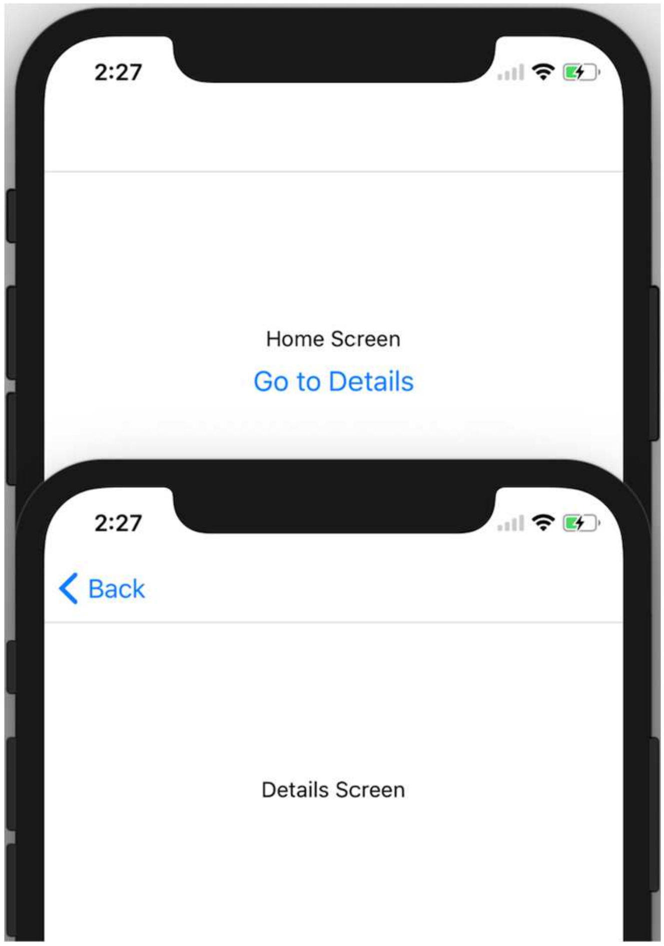
Basic React Navigation

```
import React, {Component} from 'react';
import { View, Text } from 'react-native';
import { createAppContainer } from 'react-navigation';
import { createStackNavigator } from 'react-navigation-stack';

export default class HomeScreen extends Component {
    render() {
        return (
            <View style={{flex: 1, alignItems: 'center', justifyContent: 'center'}}>
                <Text>Home Screen</Text>
            </View>
        );
    }
}

const AppNavigator = createStackNavigator({
    Home: {
        screen: HomeScreen
    }
});

const AppContainer = createAppContainer(AppNavigator);
```



Navigate To Other Screen

```
import React, {Component} from 'react';
import { View, Text, Button } from 'react-native';
import { createAppContainer } from 'react-navigation';
import { createStackNavigator } from 'react-navigation-stack';

export default class HomeScreen extends Component {
    render() {
        return (
            <View style={{flex: 1, alignItems: 'center', justifyContent: 'center'}}>
                <Text>Home Screen</Text>
                <Button
                    title="Go to Details"
                    onPress={() => this.props.navigation.navigate('Details')}
                />
            </View>
        );
    }
}
```



HIENLTH

Navigate To Other Screen

- To navigate to other screen, we use this.props.navigation.navigate(‘Details’)
  - this.props.navigation: the navigation prop is passed in to every screen component in stack navigator
  - navigate('Details'): we call the navigate function with the name of the route that we'd like to move the user to.
- If we call this.props.navigation.navigate with a route name haven’t been defined on a stack navigator, nothing happen

Navigate To A Route Multiple Times

```
class DetailScreen extends Component {
    render() {
        return (
            <View style={{flex: 1, alignItems: 'center', justifyContent: 'center'}}>
                <Text>Details Screen</Text>
                <Button
                    title="Go to Details... again"
                    onPress={() => this.props.navigation.navigate('Details')}
                />
            </View>
        );
    }
}
```

```
<Button
    title="Go to Details... again"
    onPress={() => this.props.navigation.push('Details')}
/>
```

Going Back

- To go back to previous screen, we use this.props.navigation.goBack()
- Or if you want to go back multiples screen, you can use navigate(‘Home’)
- navigation.popToTop will go back to the first screen in the stack

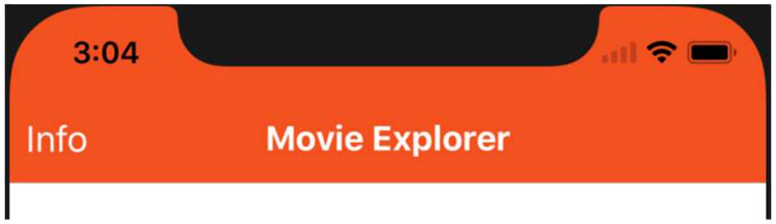
HIENLTH

Going Back

```
class DetailScreen extends Component {
    render() {
        return (
            <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
                <Text>Details Screen</Text>
                <Button
                    title="Go to Details... again"
                    onPress={() => this.props.navigation.push('Details')}
                />
                <Button
                    title="Go to Home"
                    onPress={() => this.props.navigation.navigate('Home')}
                />
                <Button
                    title="Go back"
                    onPress={() => this.props.navigation.goBack()}
                />

            </View>
        );
    }
}
```

Configuring Header Bar

```
static navigationOptions = {
  title: 'Movie Explorer',
  headerStyle: {
    backgroundColor: '#f4511e'
  },
  headerTintColor: '#fff',
  headerTitleStyle: {
    fontWeight: 'bold'
  },
  headerLeft: (
    <Button
      onPress={() => alert('This is a button!')}
      title='Info'
      color='#fff'
    />
  )
};
```



Configuring Header Bar

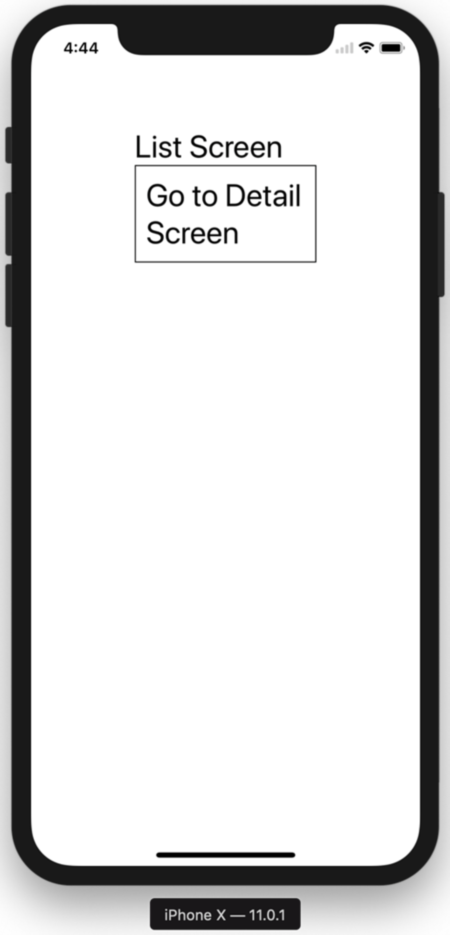
- Using static property called **navigationOptions.** It has some configuration options to configure header
  - title: title of header bar
  - headerStyle: a style object that will be applied to the View that wraps the header
  - headerTintColor: set color for back button and title
  - headerTitleStyle: use to customize the fontFamily, fontWeight and other Text style properties for the title
  - headerLeft: configure left view for header bar
  - headerRight: configure right view for header bar

Splitting Code

- We will create a Movie App with 2 screens, ListScreen and DetailScreen
- Each screen should belong to a component (having its own file)
- We need to create three new files. Each file having a scene component
  - ListScreen component in ListScreen.js
  - DetailScreen component in DetailScreen.js
  - Movie component in Movie.js

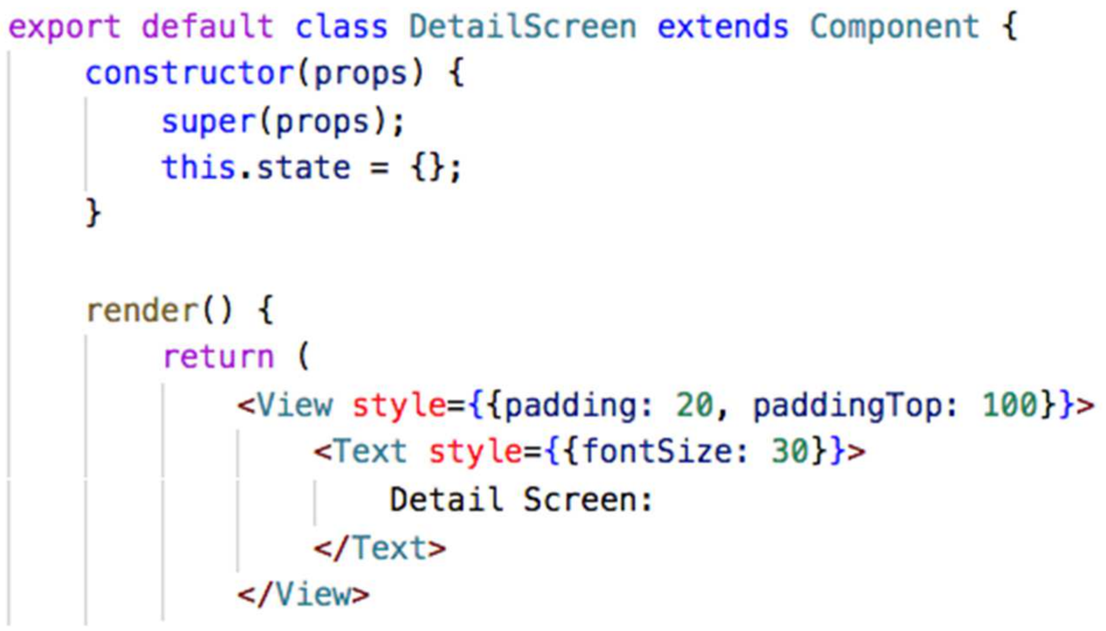
ListScreen.js

```
export default class ListScreen extends Component {

  constructor(props) {
    super(props);
  }

  render() {
    return (
      <View style={{ padding: 100 }}>
        <Text style={{ fontSize: 30 }}>
          Lits Screen
        </Text>
        <TouchableOpacity onPress={() => {
                      this.props.navigation.navigate('Details', {
                        movieName: 'Frozen'
                      })
                    }}>
          <Text style={{ fontSize: 30, padding: 10, borderWidth: 1 }}>
            Go to Detail Screen
          </Text>
        </TouchableOpacity>
      </View>
    );
  }
```

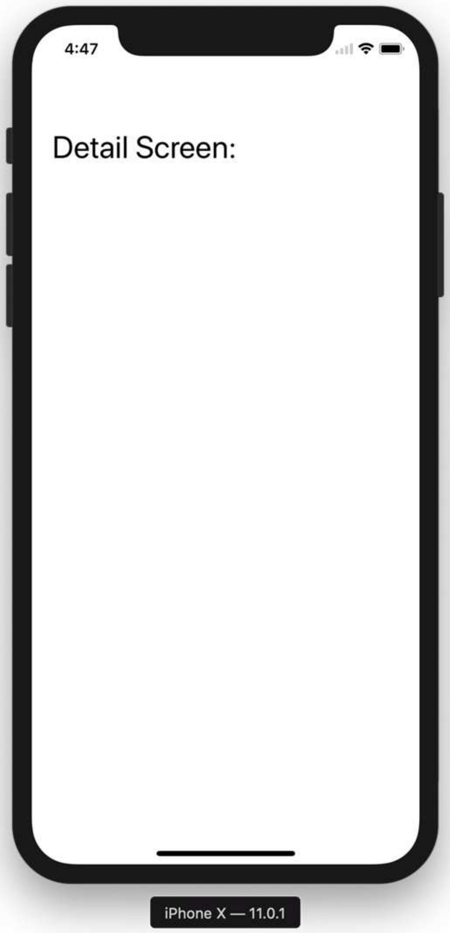


HIENLTH

DetailScreen.js

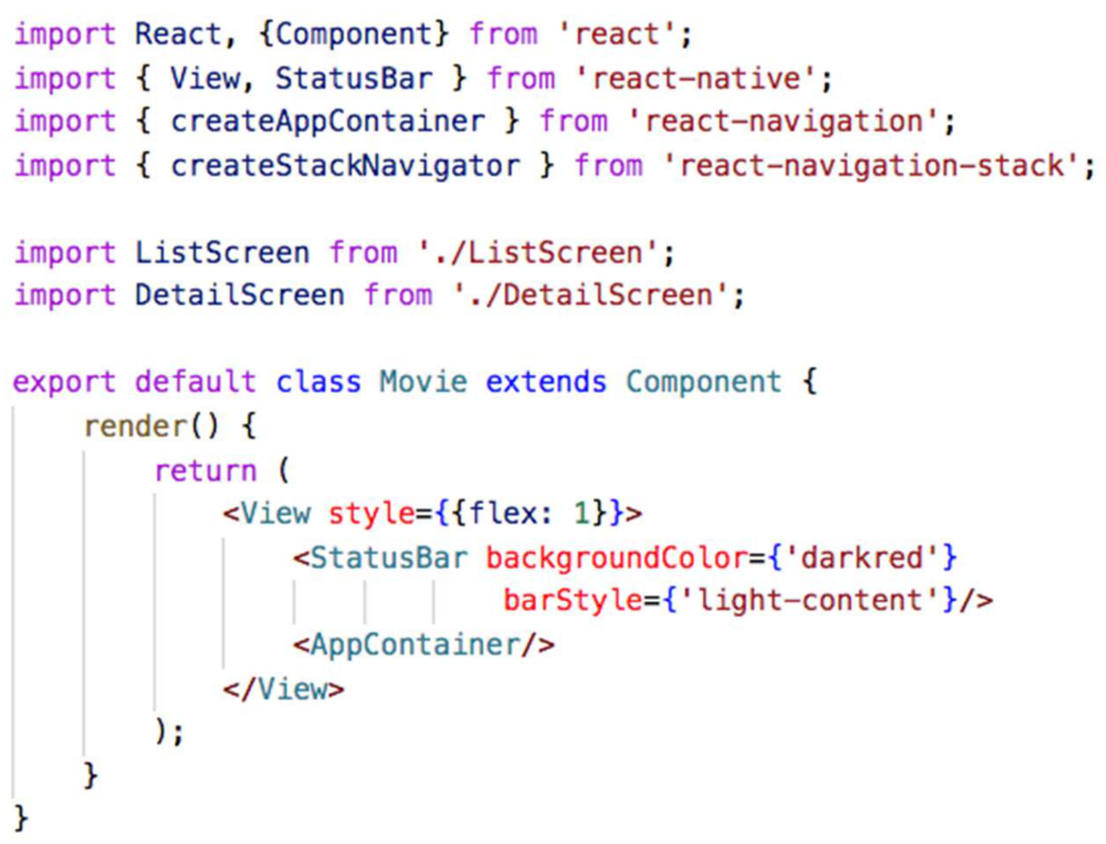


```
export default class DetailScreen extends Component {
    constructor(props) {
        super(props);
        this.state = {};
    }

    render() {
        return (
            <View style={{padding: 20, paddingTop: 100}}>
                <Text style={{fontSize: 30}}>
                    Detail Screen:
                </Text>
            </View>
```

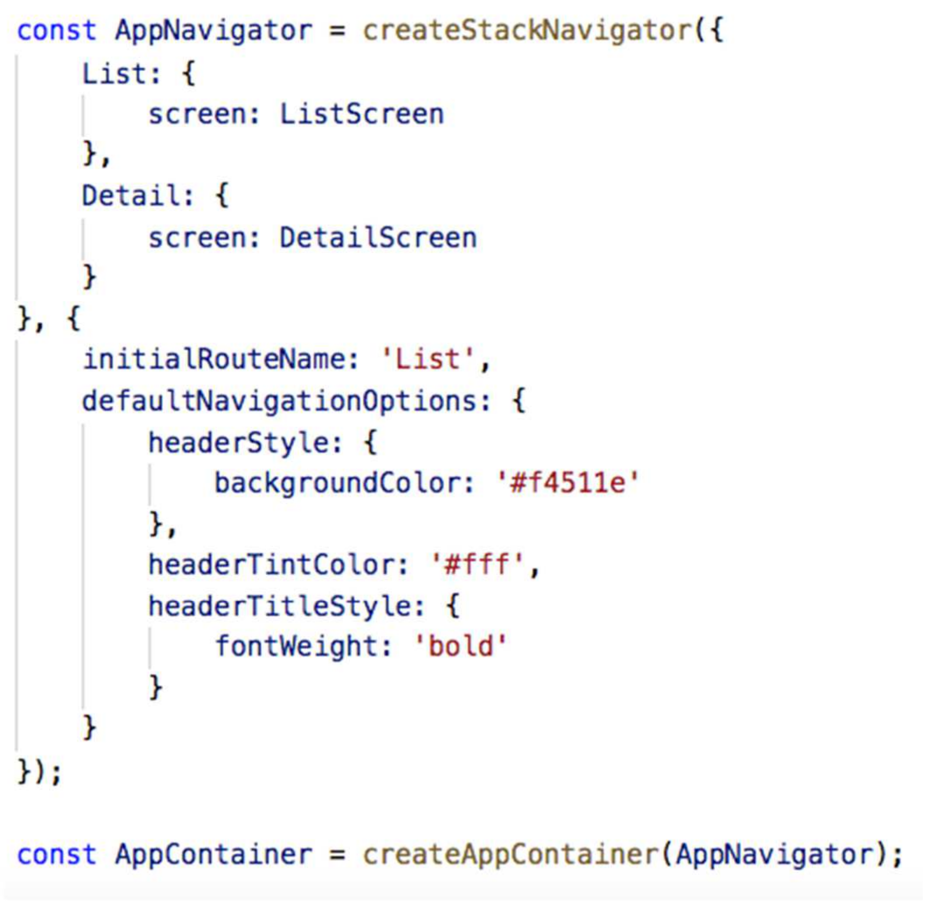


HIENLTH

Movie.js



```
import React, {Component} from 'react';
import { View, StatusBar } from 'react-native';
import { createAppContainer } from 'react-navigation';
import { createStackNavigator } from 'react-navigation-stack';

import ListScreen from './ListScreen';
import DetailScreen from './DetailScreen';

export default class Movie extends Component {
    render() {
        return (
            <View style={{flex: 1}}>
                <StatusBar backgroundColor={'darkred'}
                            barStyle={'light-content'}/>
                <AppContainer/>
            </View>
        );
    }
}
```



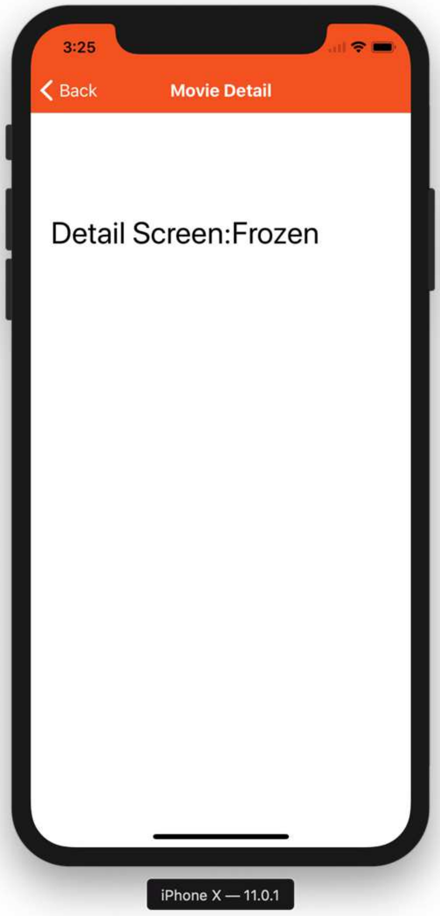
```
const AppNavigator = createStackNavigator({
    List: {
        screen: ListScreen
    },
    Detail: {
        screen: DetailScreen
    }
}, {
    initialRouteName: 'List',
    defaultNavigationOptions: {
        headerStyle: {
            backgroundColor: '#f4511e'
        },
        headerTintColor: '#fff',
        headerTitleStyle: {
            fontWeight: 'bold'
        }
    }
});

const AppContainer = createAppContainer(AppNavigator);
```

HIENLTH

Passing Value To New Scene

```
render() {
    return (
        <TouchableOpacity onPress={() =>
                        this.props.navigation.navigate('Details',
                              {movieName: 'frozen'})}>
        </TouchableOpacity>
    );
}
```

```
render() {
    return (
        <View style={{padding: 20, paddingTop: 100}}>
            <Text style={{fontSize: 30}}>
                Detail Screen:
                {this.props.navigation.getParam('movieName')}
            </Text>
        </View>
    );
}
```



Detail Screen:Frozen

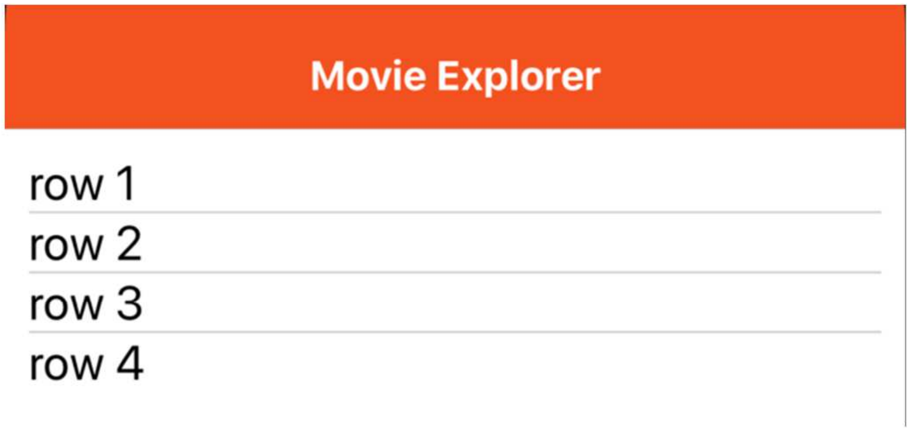
HIENLTH

FlatList

- A core component designed for efficient display of vertically scrolling lists of changing data.
- FlatList requires a FlatList.data to manipulate the display data. Its simplest form is a simple array of data.
- FlatList render each item of data according to FlatList.renderItem callback.

FlatList

```
    this.state = {
      dataSource: ['row 1', 'row 2', 'row 3', 'row 4']
    };
  }
  render() {
    return (
      <FlatList style={styles.container}
        data={this.state.dataSource}
        enableEmptySections={true}
        renderItem={(itemData) => {
          return (
            <Text style={styles.title}>{itemData.item}</Text>
          )
        }

        }
        keyExtractor={(item, index) => index.toString()}
        ItemSeperatorComponent={(sectionID, rowID, adjacentRowHighlighted) =>
          <View key={rowID} style={{ height: 1, backgroundColor: 'lightgray' }} />
        }
```



FlatList Benefits over Custom-made UI

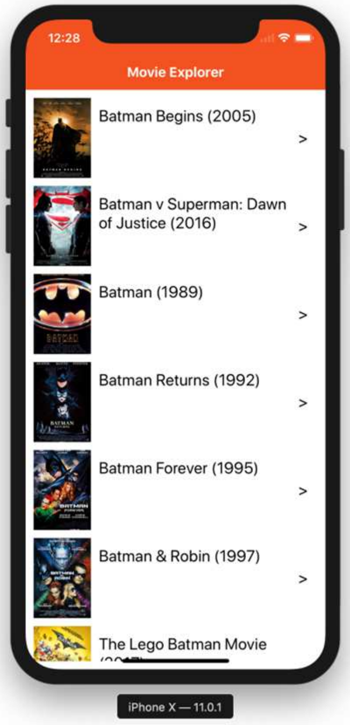
- FlatList supports advanced features such as:
  - Scrolling
  - Section Segmentation
  - Sticky Section Headers
  - Header and Footer Supports
  - Callback when reaching the end of the available data (onEndReached), which can enable infinite browsing.

Exercise

- DesignMovie App
- Skills to be achieved:
  - Fetching data from API (need register account to get key: http://www.omdbapi.com)
  - FlatList
  - React Navigation
- Some suggestion screens:
- Duration 40 minutes

http://www.omdbapi.com/?apikey=**<key>**&s=**batman**&plot=full

http://www.omdbapi.com/?apikey=<key>&i=<**imdbID**>



Batman Begins (2005) >

Batman v Superman: Dawn of Justice (2016) >

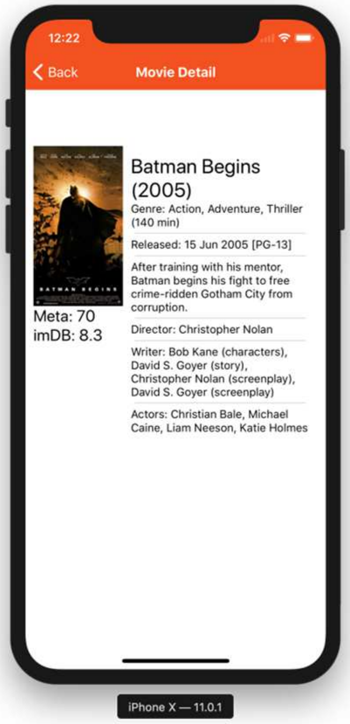
Batman (1989) >

Batman Returns (1992) >

Batman Forever (1995) >

Batman & Robin (1997) >

The Lego Batman Movie



Batman Begins (2005)

Genre: Action, Adventure, Thriller (140 min)

Released: 15 Jun 2005 [PG-13]

After training with his mentor, Batman begins his fight to free crime-ridden Gotham City from corruption.

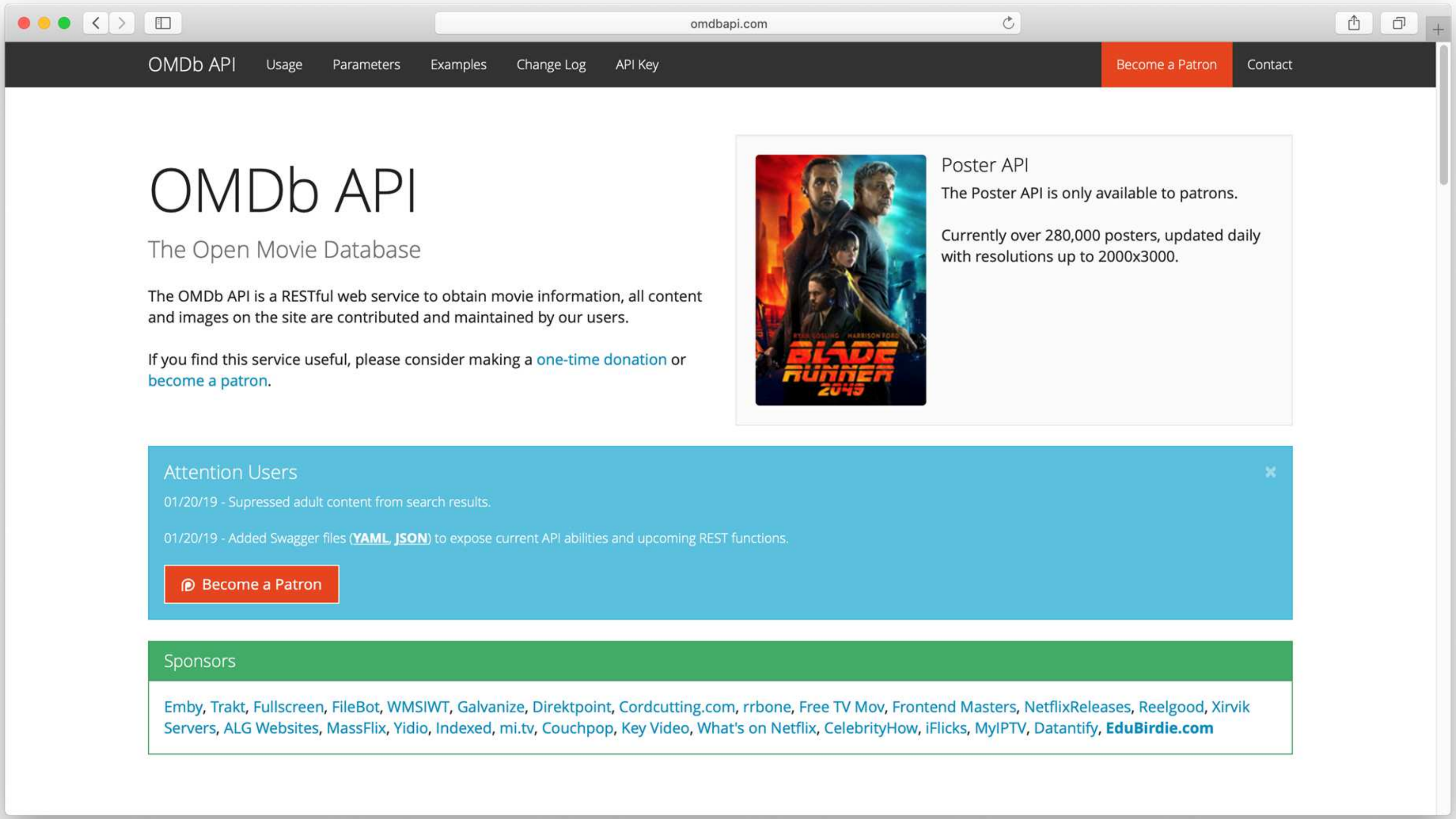
Director: Christopher Nolan

Writer: Bob Kane (characters), David S. Goyer (story), Christopher Nolan (screenplay), David S. Goyer (screenplay)

Actors: Christian Bale, Michael Caine, Liam Neeson, Katie Holmes

Meta: 70
imDB: 8.3

OMDb API



OMDb API

The Open Movie Database

The OMDb API is a RESTful web service to obtain movie information, all content and images on the site are contributed and maintained by our users.

If you find this service useful, please consider making a one-time donation or become a patron.

Poster API

The Poster API is only available to patrons.

Currently over 280,000 posters, updated daily with resolutions up to 2000x3000.

Attention Users

01/20/19 - Supressed adult content from search results.

01/20/19 - Added Swagger files (**YAML**, **JSON**) to expose current API abilities and upcoming REST functions.

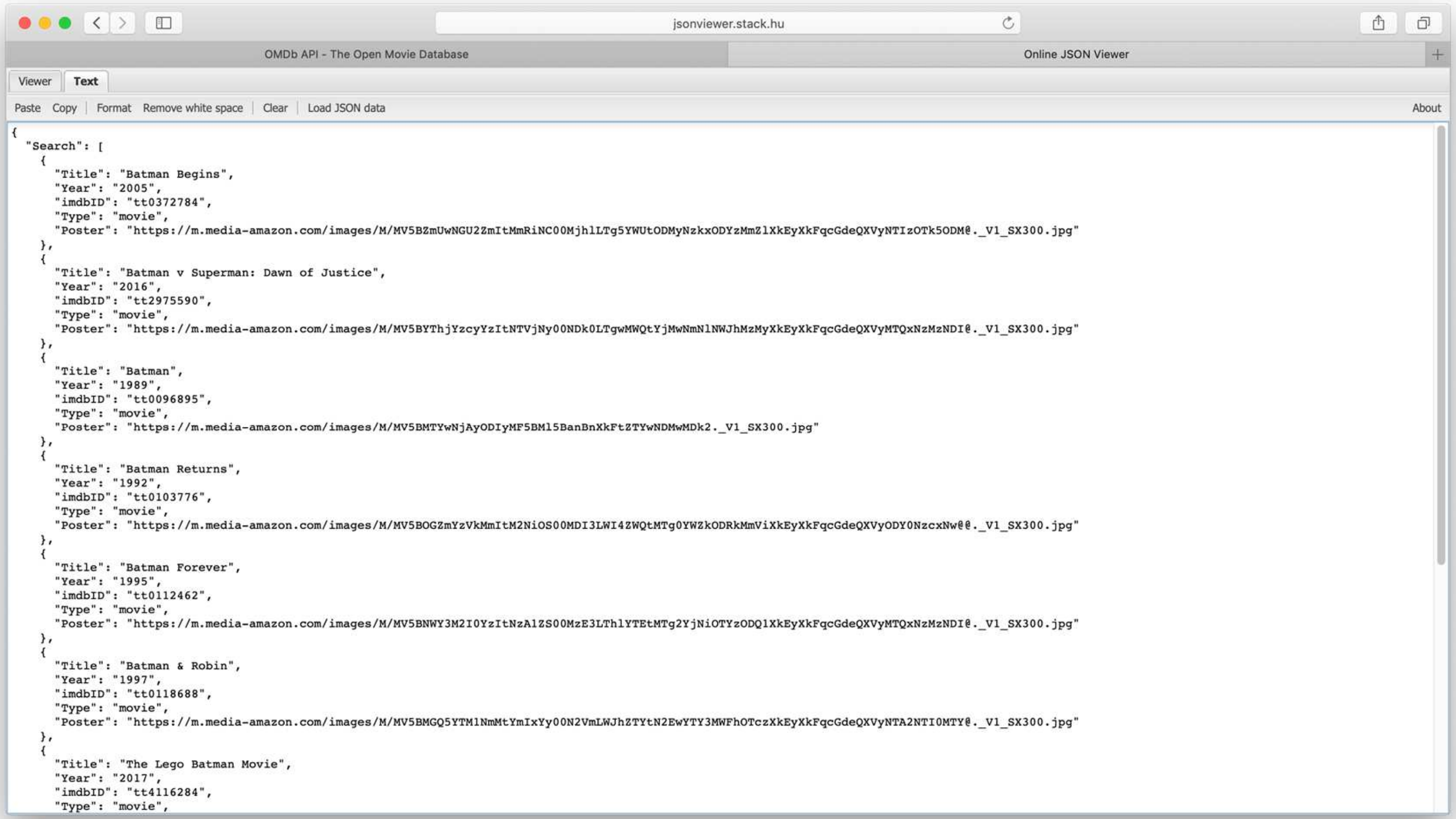
Become a Patron

Sponsors

Emby, Trakt, Fullscreen, FileBot, WMSIWT, Galvanize, Direktpoint, Cordcutting.com, rrbone, Free TV Mov, Frontend Masters, NetflixReleases, Reelgood, Xirvik Servers, ALG Websites, MassFlix, Yidio, Indexed, mi.tv, Couchpop, Key Video, What's on Netflix, CelebrityHow, iFlicks, MyIPTV, Datantify, **EduBirdie.com**

Search Movie

http://www.omdbapi.com/?s=Batman



```
{
  "Search": [
    {
      "Title": "Batman Begins",
      "Year": "2005",
      "imdbID": "tt0372784",
      "Type": "movie",
      "Poster": "https://m.media-amazon.com/images/M/MV5BZmUwNGU2ZmItMmRiNC00MjhlLTg5YWUtODMyNzkxODYzMmZlXkEyXkFqcGdeQXVyNTIzOTk5ODM@._V1_SX300.jpg"
    },
    {
      "Title": "Batman v Superman: Dawn of Justice",
      "Year": "2016",
      "imdbID": "tt2975590",
      "Type": "movie",
      "Poster": "https://m.media-amazon.com/images/M/MV5BYThjYzcyYzItNTVjNy00NDk0LTgwMWQtYjMwNmNlNWJhMzMyXkEyXkFqcGdeQXVyMTQxNzMzNDI@._V1_SX300.jpg"
    },
    {
      "Title": "Batman",
      "Year": "1989",
      "imdbID": "tt0096895",
      "Type": "movie",
      "Poster": "https://m.media-amazon.com/images/M/MV5BMTYwNjAyODIyMF5BMl5BanBnXkFtZTYwNDMwMDk2._V1_SX300.jpg"
    },
    {
      "Title": "Batman Returns",
      "Year": "1992",
      "imdbID": "tt0103776",
      "Type": "movie",
      "Poster": "https://m.media-amazon.com/images/M/MV5BOGZmYzVkMmItM2NiOS00MDI3LWI4ZWQtMTg0YWZkODRkMmViXkEyXkFqcGdeQXVyODY0NzcxNw@@._V1_SX300.jpg"
    },
    {
      "Title": "Batman Forever",
      "Year": "1995",
      "imdbID": "tt0112462",
      "Type": "movie",
      "Poster": "https://m.media-amazon.com/images/M/MV5BNWY3M2I0YzItNzA1ZS00MzE3LThlYTEtMTg2YjNiOTYzODQ1XkEyXkFqcGdeQXVyMTQxNzMzNDI@._V1_SX300.jpg"
    },
    {
      "Title": "Batman & Robin",
      "Year": "1997",
      "imdbID": "tt0118688",
      "Type": "movie",
      "Poster": "https://m.media-amazon.com/images/M/MV5BMGQ5YTM1NmMtYmIxYy00N2VmLWJhZTYtN2EwYTY3MWFhOTczXkEyXkFqcGdeQXVyNTA2NTI0MTY@._V1_SX300.jpg"
    },
    {
      "Title": "The Lego Batman Movie",
      "Year": "2017",
      "imdbID": "tt4116284",
      "Type": "movie",
```

API JSON

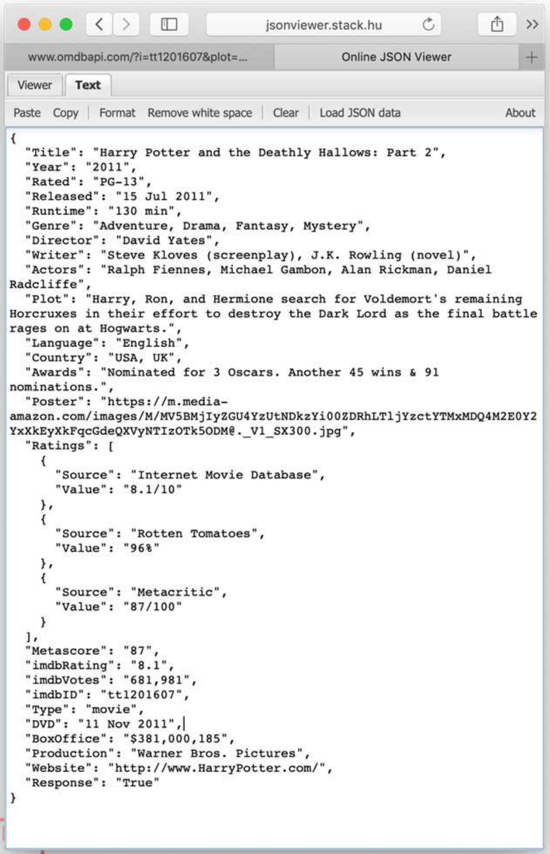
http://www.omdbapi.com/?s=Harry

```
{
 "Search": [
   {
     "Title": "Harry Potter and the Deathly Hallows: Part 2",
     "Year": "2011",
     "imdbID": "tt1201607",
     "Type": "movie",
     "Poster": "https://m.media-
amazon.com/images/M/MV5BMjIyZGU4YzUtNDkzYi00ZDRhLTljYzctYTMxMDQ4M2E0Y2YxXkEyXkFqc
GdeQXVyNTIzOTk5ODM@._V1_SX300.jpg"
   },
   ...
}
```

HIENLTH

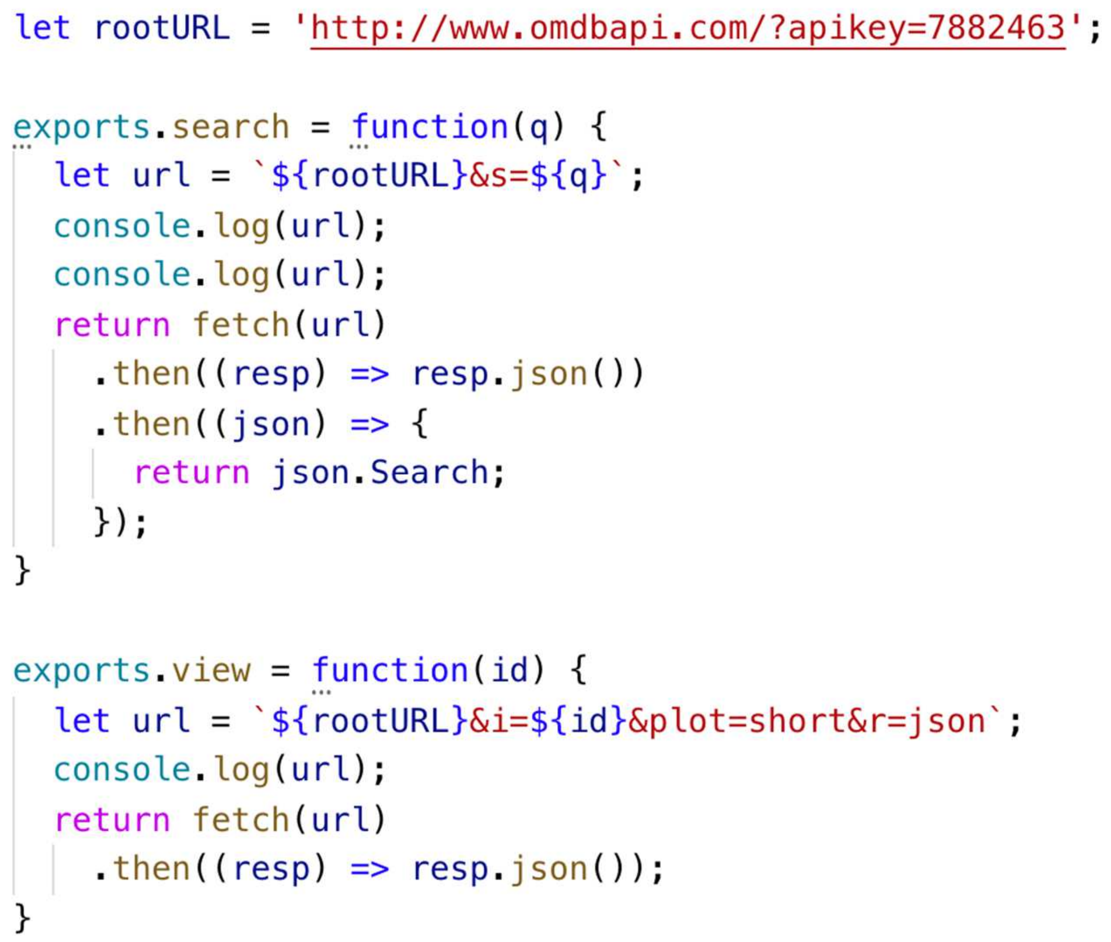
Get Movie Information

http://www.omdbapi.com/?i=tt1201607&plot=short&r=json



```
{
  "Title": "Harry Potter and the Deathly Hallows: Part 2",
  "Year": "2011",
  "Rated": "PG-13",
  "Released": "15 Jul 2011",
  "Runtime": "130 min",
  "Genre": "Adventure, Drama, Fantasy, Mystery",
  "Director": "David Yates",
  "Writer": "Steve Kloves (screenplay), J.K. Rowling (novel)",
  "Actors": "Ralph Fiennes, Michael Gambon, Alan Rickman, Daniel
Radcliffe",
  "Plot": "Harry, Ron, and Hermione search for Voldemort's remaining
Horcruxes in their effort to destroy the Dark Lord as the final battle
rages on at Hogwarts.",
  "Language": "English",
  "Country": "USA, UK",
  "Awards": "Nominated for 3 Oscars. Another 45 wins & 91
nominations.",
  "Poster": "https://m.media-
amazon.com/images/M/MV5BMjIyZGU4YzUtNDkzYi00ZDRhLTljYzctYTMxMDQ4M2E0Y2
YxXkEyXkFqcGdeQXVyNTIzOTk5ODM@._V1_SX300.jpg",
  "Ratings": [
    {
      "Source": "Internet Movie Database",
      "Value": "8.1/10"
    },
    {
      "Source": "Rotten Tomatoes",
      "Value": "96%"
    },
    {
      "Source": "Metacritic",
      "Value": "87/100"
    }
  ],
  "Metascore": "87",
  "imdbRating": "8.1",
  "imdbVotes": "681,981",
  "imdbID": "tt1201607",
  "Type": "movie",
  "DVD": "11 Nov 2011",
  "BoxOffice": "$381,000,185",
  "Production": "Warner Bros. Pictures",
  "Website": "http://www.HarryPotter.com/",
  "Response": "True"
}
```

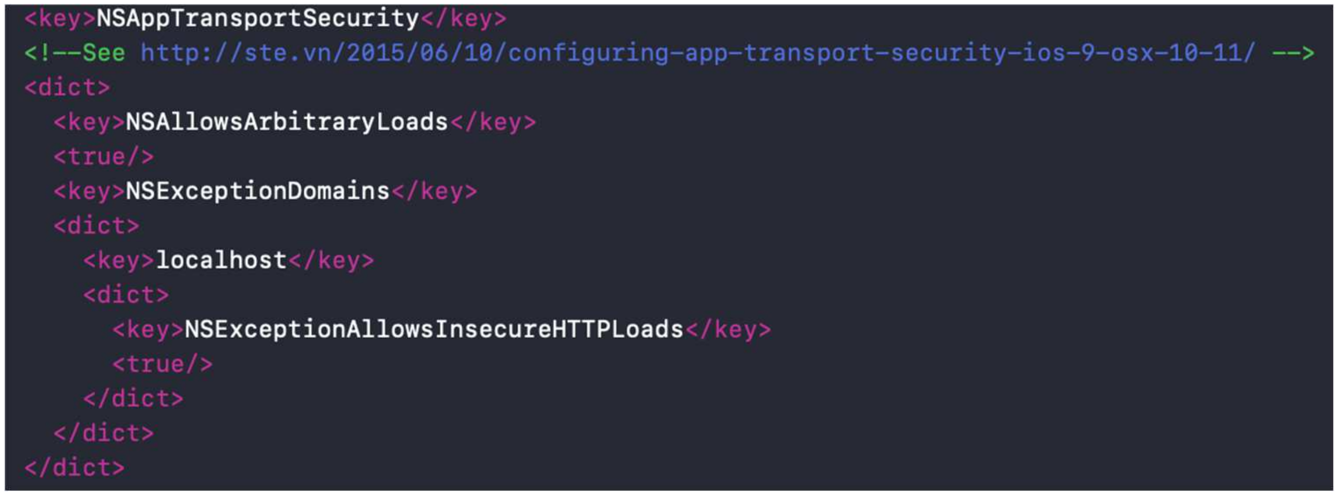
Making API



```
let rootURL = 'http://www.omdbapi.com/?apikey=7882463';

exports.search = function(q) {
  let url = `${rootURL}&s=${q}`;
  console.log(url);
  console.log(url);
  return fetch(url)
    .then((resp) => resp.json())
    .then((json) => {
      return json.Search;
    });
}

exports.view = function(id) {
  let url = `${rootURL}&i=${id}&plot=short&r=json`;
  console.log(url);
  return fetch(url)
    .then((resp) => resp.json());
}
```

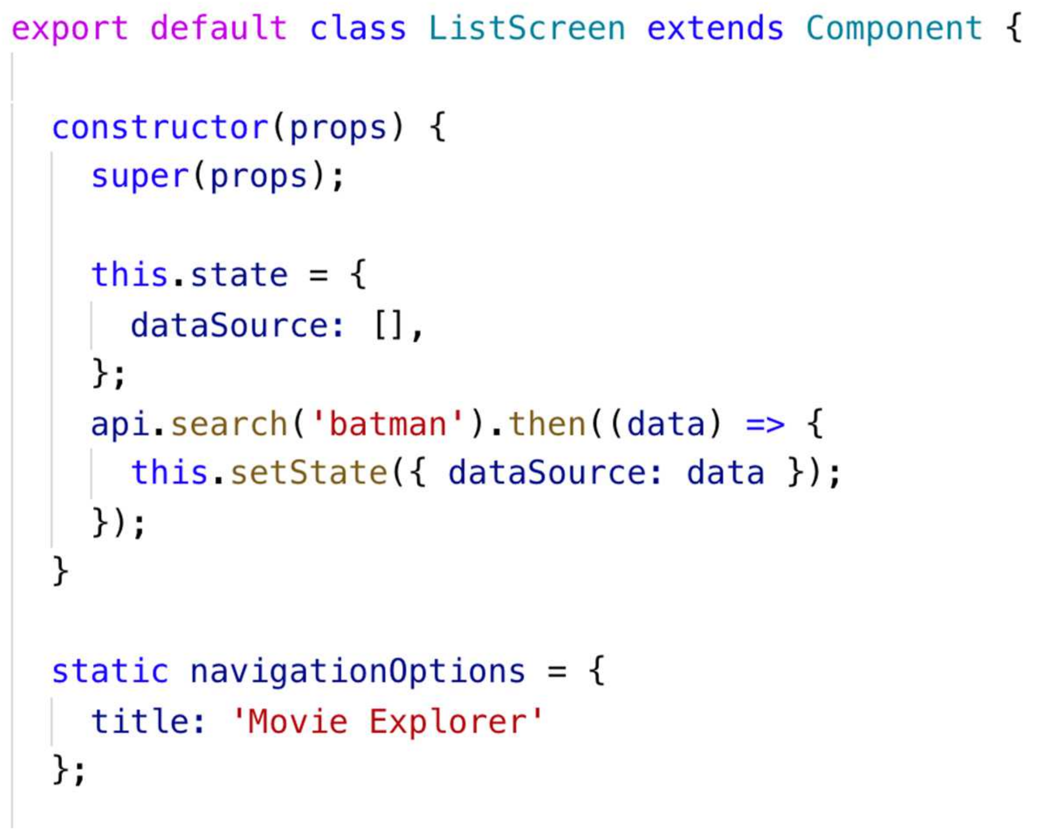
iOS Block HTTP connection

- By default, only HTTPS connection is allowed in iOS
- To remove this restriction, go to {project folder}/ios/l8_movie/info.plist
- At <Key>NSAppTransportSecurity<Key>
- Set value as true.



```
<key>NSAppTransportSecurity</key>
<!--See http://ste.vn/2015/06/10/configuring-app-transport-security-ios-9-osx-10-11/ -->
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
  <key>NSExceptionDomains</key>
  <dict>
    <key>localhost</key>
    <dict>
      <key>NSExceptionAllowsInsecureHTTPLoads</key>
      <true/>
    </dict>
  </dict>
</dict>
```

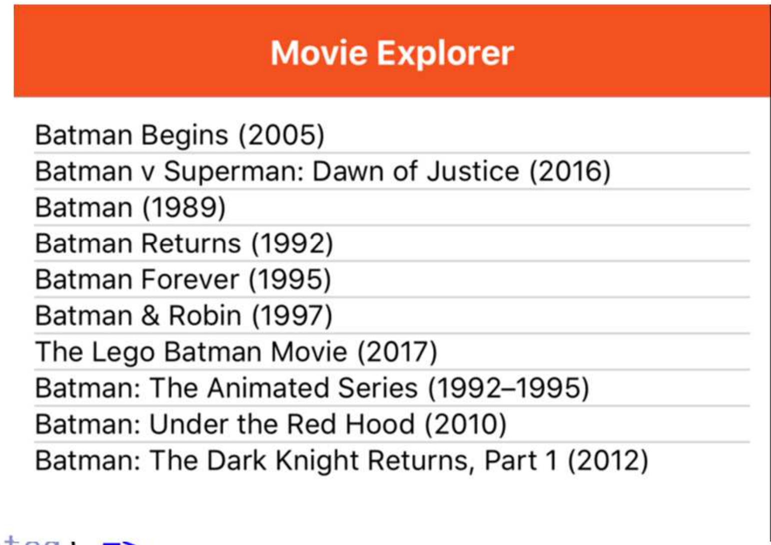
Getting Movies List by Search



```
export default class ListScreen extends Component {

  constructor(props) {
    super(props);

    this.state = {
      dataSource: [],
    };
    api.search('batman').then((data) => {
      this.setState({ dataSource: data });
    });
  }

  static navigationOptions = {
    title: 'Movie Explorer'
  };
```

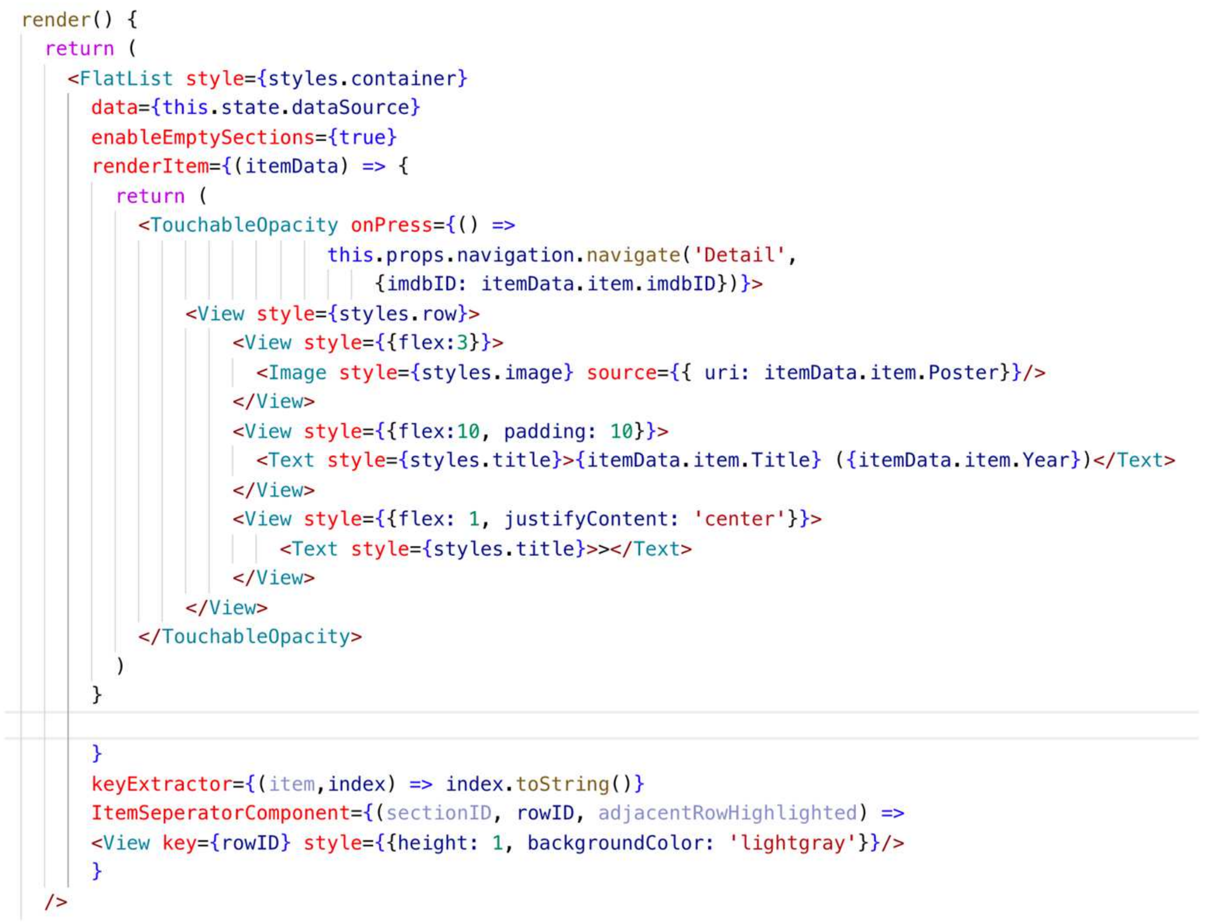
Getting Movies List by Search

```
<FlatList style={styles.container}
  data={this.state.dataSource}
  enableEmptySections={true}
  renderItem={(itemData) => {
    return (
      <Text>{itemData.item.Title} ({itemData.item.Year})</Text>
    )
  }

  }
  keyExtractor={(item,index) => index.toString()}
  ItemSeperatorComponent={(sectionID, rowID, adjacentRowHighlighted) =>
  <View key={rowID} style={{height: 1, backgroundColor: 'lightgray'}}/>
  }
/>
);
```


Movie Explorer

Batman Begins (2005)
Batman v Superman: Dawn of Justice (2016)
Batman (1989)
Batman Returns (1992)
Batman Forever (1995)
Batman & Robin (1997)
The Lego Batman Movie (2017)
Batman: The Animated Series (1992–1995)
Batman: Under the Red Hood (2010)
Batman: The Dark Knight Returns, Part 1 (2012)

Styling Row

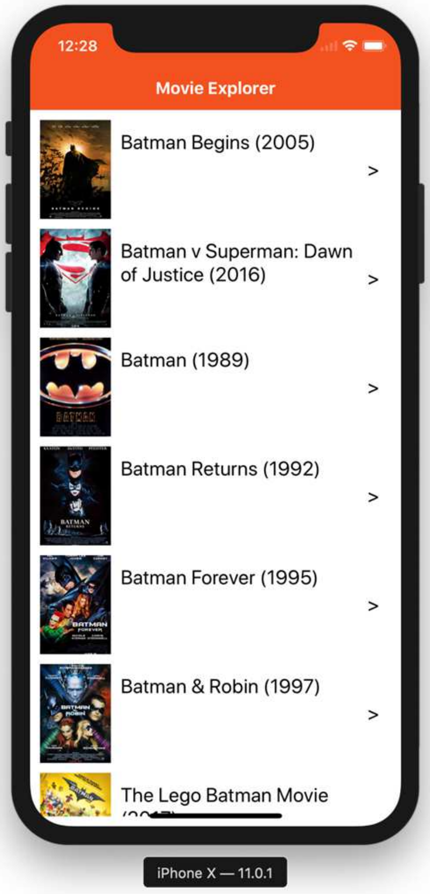


```
render() {
  return (
    <FlatList style={styles.container}
      data={this.state.dataSource}
      enableEmptySections={true}
      renderItem={(itemData) => {
        return (
          <TouchableOpacity onPress={() =>
                    this.props.navigation.navigate('Detail',
                      {imdbID: itemData.item.imdbID})}>
              <View style={styles.row}>
                <View style={{flex:3}}>
                  <Image style={styles.image} source={{ uri: itemData.item.Poster}}/>
                </View>
                <View style={{flex:10, padding: 10}}>
                  <Text style={styles.title}>{itemData.item.Title} ({itemData.item.Year})</Text>
                </View>
                <View style={{flex: 1, justifyContent: 'center'}}>
                    <Text style={styles.title}>></Text>
                </View>
              </View>
          </TouchableOpacity>
        )
      }

      }
      keyExtractor={(item,index) => index.toString()}
      ItemSeperatorComponent={(sectionID, rowID, adjacentRowHighlighted) =>
      <View key={rowID} style={{height: 1, backgroundColor: 'lightgray'}}/>
      }
  />
```

Styling Row

```
const styles = StyleSheet.create({
  container: {
    padding: 10,
    marginBottom: 10,
    flex: 1
  },
  row: {
    flexDirection: 'row',
    height: 100,
    marginBottom: 10,
    marginTop: 10
  },
  image: {
    height: 100
  },
  title: {
    fontSize: 20
  }
});
```

ListScreen.js

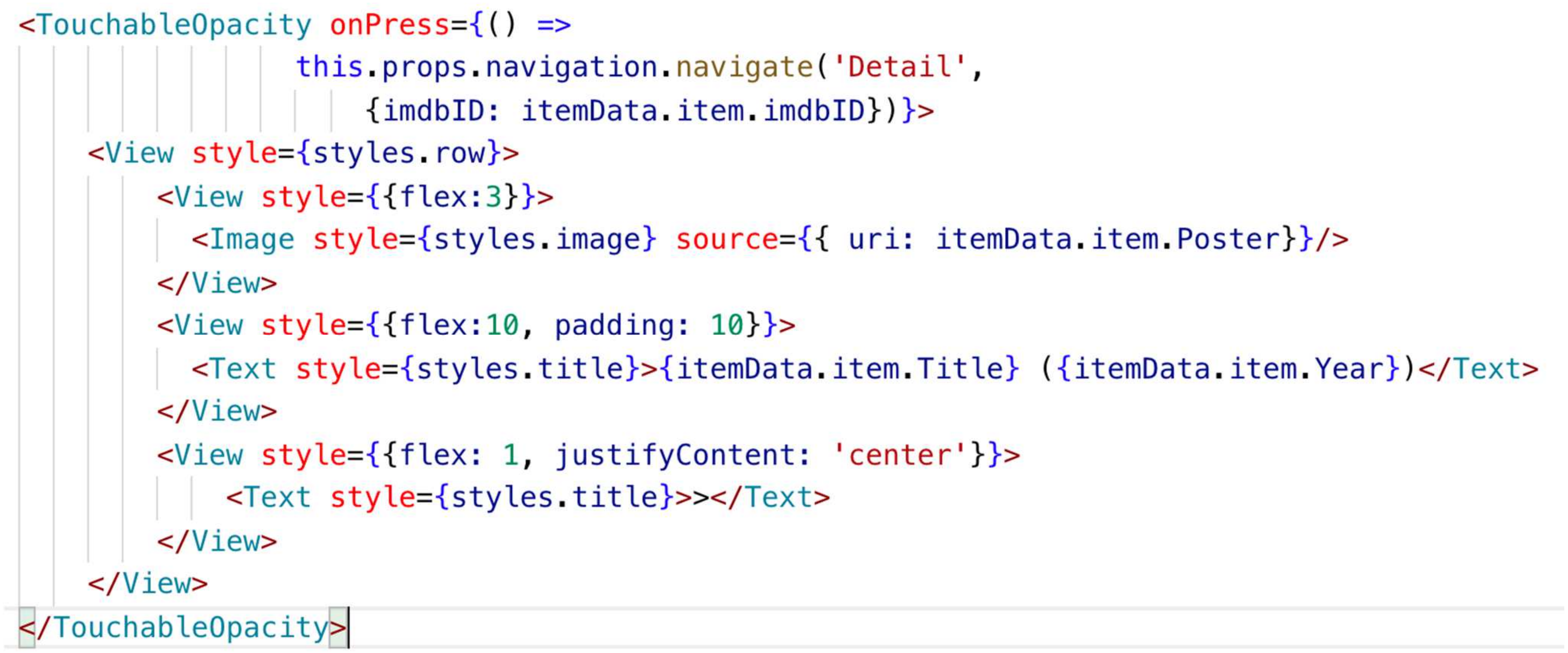


Batman Begins (2005) >
Batman v Superman: Dawn of Justice (2016) >
Batman (1989) >
Batman Returns (1992) >
Batman Forever (1995) >
Batman & Robin (1997) >
The Lego Batman Movie

HIENLTH

Adding Button to DetailScreen

- Since we need imdbID for querying movie information, we send it to detail screen instead of the movie name.

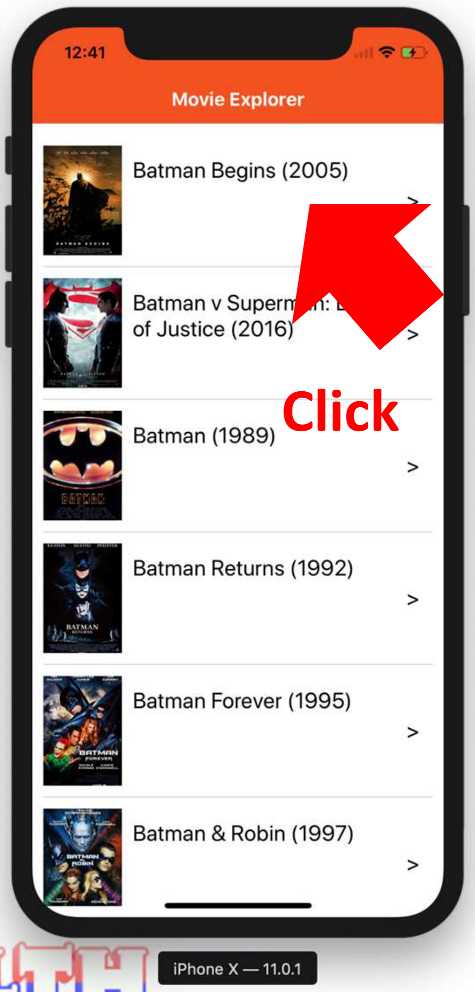


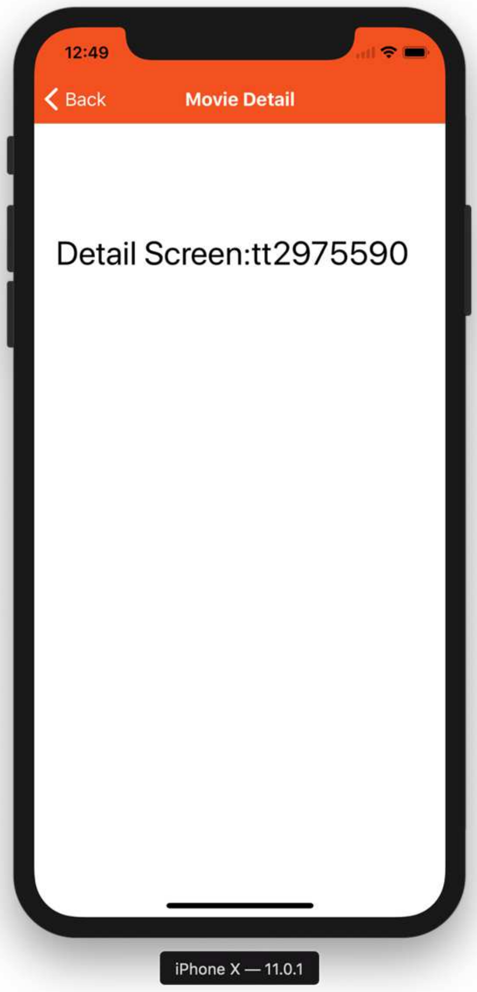
```
<TouchableOpacity onPress={() =>
                this.props.navigation.navigate('Detail',
                    {imdbID: itemData.item.imdbID})}>
    <View style={styles.row}>
        <View style={{flex:3}}>
          <Image style={styles.image} source={{ uri: itemData.item.Poster}}/>
        </View>
        <View style={{flex:10, padding: 10}}>
          <Text style={styles.title}>{itemData.item.Title} ({itemData.item.Year})</Text>
        </View>
        <View style={{flex: 1, justifyContent: 'center'}}>
            <Text style={styles.title}>></Text>
        </View>
    </View>
</TouchableOpacity>
```

Show imdbID in Detail Screen

```
<View style={{padding: 20, paddingTop: 100}}>
    <Text style={{fontSize: 30}}>
        Detail Screen:
        {this.props.navigation.getParam('imdbID')}
    </Text>
</View>
```

Show imdbID in Detail Screen



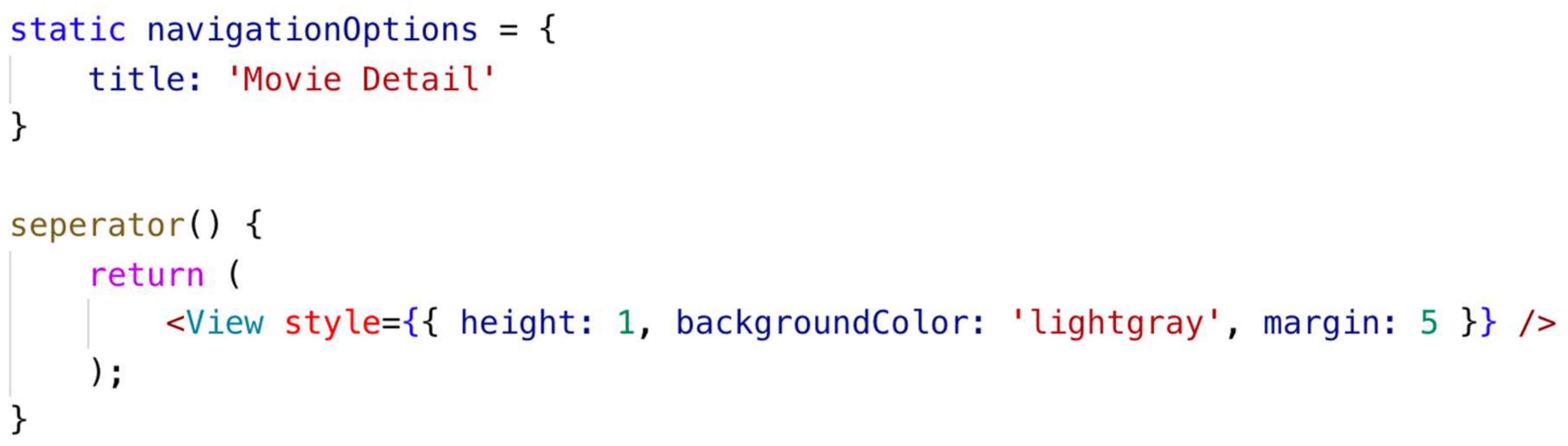


Detail Screen:tt2975590

DetailScreen: Getting Movie Information

```
export default class DetailScreen extends Component {
    constructor(props) {
        super(props);
        this.state = {};

        const { navigation } = this.props;
        const imdbID = navigation.getParam('imdbID');

        api.view(imdbID).then((data) => {
            this.setState(data);
        });
    }
```

DetailScreen: Getting Movie Information



```
static navigationOptions = {
    title: 'Movie Detail'
}

seperator() {
    return (
        <View style={{ height: 1, backgroundColor: 'lightgray', margin: 5 }} />
    );
}
```

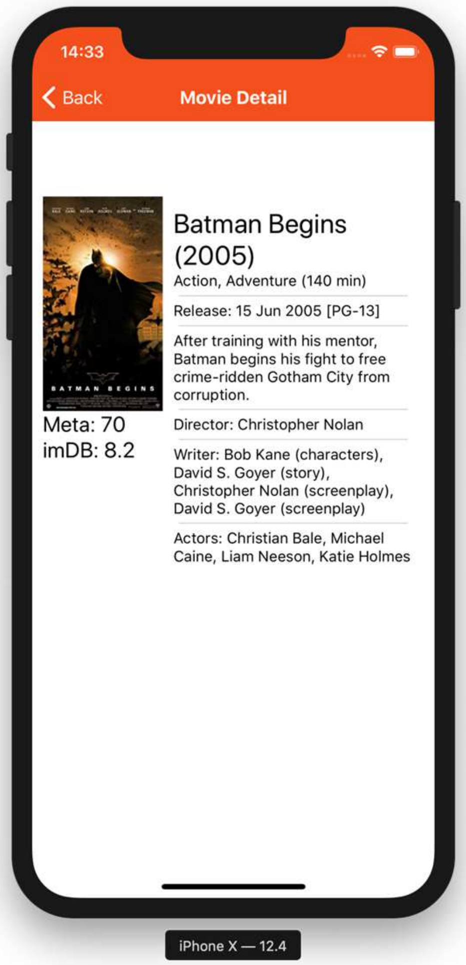
DetailScreen: Getting Movie Information

```
render() {
    return (
        <View style={styles.container}>
            <View style={{ flex: 1 }}>
                <Image style={styles.image} source={{ uri: this.state.Poster }} />
                <Text style={styles.subTitle}>Meta: {this.state.Metascore}</Text>
                <Text style={styles.subTitle}>imDB: {this.state.imdbRating}</Text>
            </View>
            <View style={{ flex: 2, padding: 10 }}>
                <Text style={styles.title}>{this.state.Title} ({this.state.Year})</Text>
                <Text>{this.state.Genre} ({this.state.Runtime})</Text>
                {this.seperator()}
                <Text>Release: {this.state.Released} [{this.state.Rated}]</Text>
                {this.seperator()}
                <Text>{this.state.Plot}</Text>
                {this.seperator()}
                <Text>Director: {this.state.Director}</Text>
                {this.seperator()}
                <Text>Writer: {this.state.Writer}</Text>
                {this.seperator()}
                <Text>Actors: {this.state.Actors}</Text>
            </View>
        </View>
    );
}
```

DetailScreen: Getting Movie Information

```
let styles = StyleSheet.create({
    container: {
        flex: 1,
        padding: 10,
        paddingTop: 70,
        flexDirection: 'row'
    },
    image: {
        height: 200
    },
    title: {
        fontSize: 25
    },
    subTitle: {
        fontSize: 20
    }
});
```

DetailScreen.js



Batman Begins (2005)

Action, Adventure (140 min)

Release: 15 Jun 2005 [PG-13]

After training with his mentor, Batman begins his fight to free crime-ridden Gotham City from corruption.

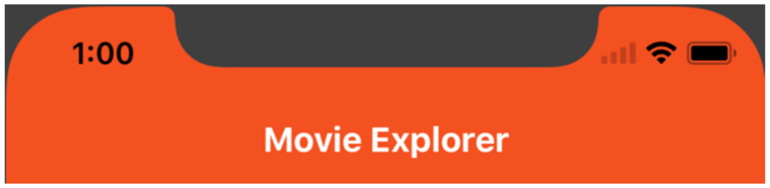
Director: Christopher Nolan

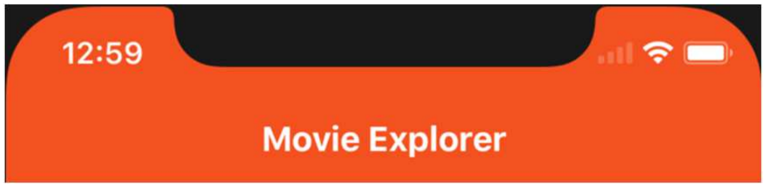
Writer: Bob Kane (characters), David S. Goyer (story), Christopher Nolan (screenplay), David S. Goyer (screenplay)

Actors: Christian Bale, Michael Caine, Liam Neeson, Katie Holmes

Meta: 70
imDB: 8.2

Fixing StatusBar Style



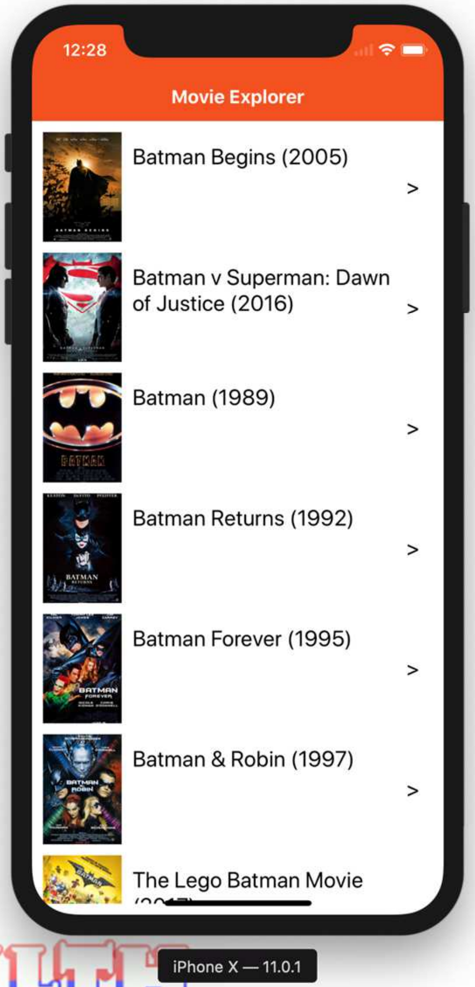


```
export default class Movie extends Component {
    render() {
        return (
            <View style={{flex: 1}}>
                <AppContainer/>
            </View>
        );
    }
}
```

```
export default class Movie extends Component {
    render() {
        return (
            <View style={{flex: 1}}>
                <StatusBar backgroundColor={'darkred'}
                           barStyle={'light-content'}/>
                <AppContainer/>
            </View>
        );
    }
}
```

HIENLTH

Final App


12:28

Movie Explorer

Batman Begins (2005) >

Batman v Superman: Dawn of Justice (2016) >

Batman (1989) >

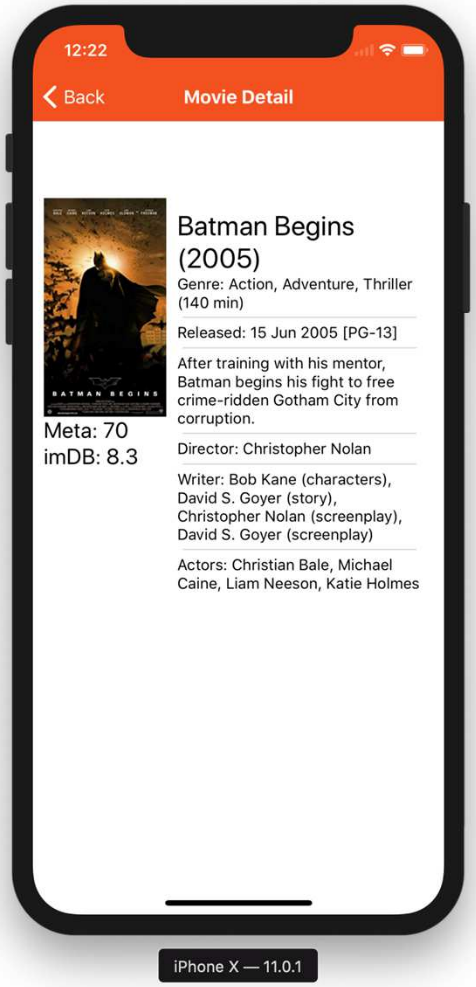
Batman Returns (1992) >

Batman Forever (1995) >

Batman & Robin (1997) >

The Lego Batman Movie (2017)

iPhone X — 11.0.1


12:22

< Back Movie Detail

Batman Begins (2005)

Genre: Action, Adventure, Thriller (140 min)

Released: 15 Jun 2005 [PG-13]

After training with his mentor, Batman begins his fight to free crime-ridden Gotham City from corruption.

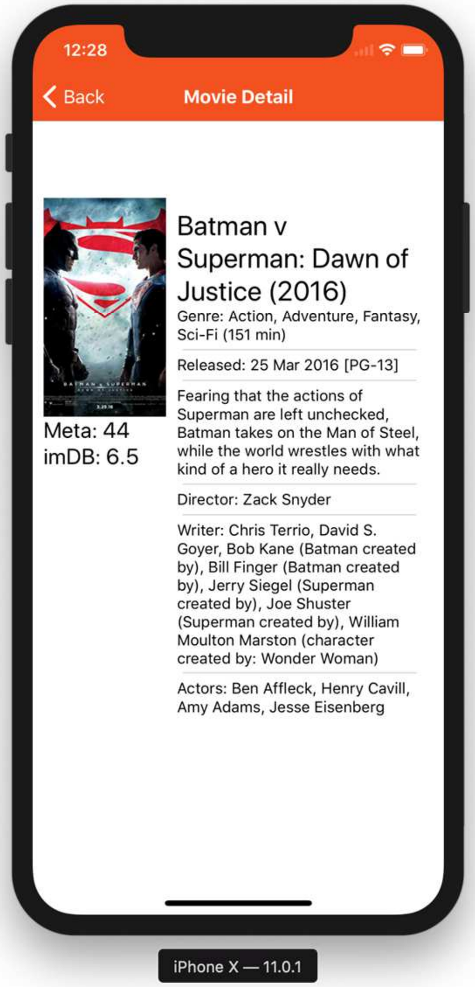
Director: Christopher Nolan

Writer: Bob Kane (characters), David S. Goyer (story), Christopher Nolan (screenplay), David S. Goyer (screenplay)

Actors: Christian Bale, Michael Caine, Liam Neeson, Katie Holmes

Meta: 70
imDB: 8.3

iPhone X — 11.0.1


12:28

< Back Movie Detail

Batman v Superman: Dawn of Justice (2016)

Genre: Action, Adventure, Fantasy, Sci-Fi (151 min)

Released: 25 Mar 2016 [PG-13]

Fearing that the actions of Superman are left unchecked, Batman takes on the Man of Steel, while the world wrestles with what kind of a hero it really needs.

Director: Zack Snyder

Writer: Chris Terrio, David S. Goyer, Bob Kane (Batman created by), Bill Finger (Batman created by), Jerry Siegel (Superman created by), Joe Shuster (Superman created by), William Moulton Marston (character created by: Wonder Woman)

Actors: Ben Affleck, Henry Cavill, Amy Adams, Jesse Eisenberg

Meta: 44
imDB: 6.5

iPhone X — 11.0.1

HomeWork

- Implementing Movie Explorer using theMovieDb api
- Why only “Batman” movie?
- Make a search box for any kind of movies you want
- Change animation when navigating from ListScreen to DetailScreen

Thank you!