Opening a PDF in React Native

Julian Grosshauser

React Native lets you create mobile apps in JavaScript, but instead of building a “hybrid app,” you’re using the same UI building blocks as regular iOS and Android apps. In this post, we’re going to use React Native to build an app that opens a PDF with the press of a button. First we’re going to use an npm package to present the PDF. Then we’ll see how easy it is to integrate PSPDFKit to add even more PDF features to your app.

Below you’ll see the steps for how to open a PDF in React Native with `react-native-pdf`.

Step 1: Installing the Prerequisites

We’re going to use `yarn` to install packages. Please follow the Yarn installation guide to set it up on your system if you haven’t yet installed it.

In order to create React Native apps from the command line, we need to install `react-native-cli`:

```
1  yarn global add react-native-cli
```

Step 2: Creating a New React Native App

We can use `react-native` to create a new React Native app from the command line (we chose the name “OpeningaPDF” for our app):

```
1  react-native init OpeningaPDF
```

For the rest of the tutorial, we’re going to work in “OpeningaPDF”:

```
1  cd OpeningaPDF
```

Step 3: Installing Dependencies

We're going to use `react-navigation` so that we can switch from one view to another in our app:

```
yarn add react-navigation
```

Next, we'll add `react-native-pdf`, which includes a `Pdf` component for us to use:

```
yarn add react-native-pdf
yarn add react-native-fetch-blob
react-native link react-native-pdf
react-native link react-native-fetch-blob
```

Step 4: Writing the App

Now we can start implementing our app. First, we're going to import all the required packages and initialize our navigation stack in `App.js`:

JavaScript

```
import React, { Component } from 'react';
import {
  StyleSheet,
  Dimensions,
  Button,
  View
} from 'react-native';
import { StackNavigator } from 'react-navigation';
import Pdf from 'react-native-pdf';

// Simple screen containing an "Open PDF" button
class HomeScreen extends Component {
  static navigationOptions = {
    title: 'Home'
  };
  render() {
    const { navigate } = this.props.navigation;
    return (
      <View style={styles.container}>
        <Button
          onPress={() => navigate('Pdf')}
          title='Open PDF'
        />
      </View>
    );
  }
}

// Screen that shows the contents of a PDF
class PdfScreen extends Component {
  static navigationOptions = {
    title: 'PDF'
  };

  render() {
    const source = require('./document.pdf');

    return <Pdf
              source={source}
              style={styles.pdf}
            />;
  }
}

const PdfApp = StackNavigator({
  Home: { screen: HomeScreen },
  Pdf: { screen: PdfScreen }
});
```

`HomeScreen` contains an “Open PDF” button that navigates to `PdfScreen`. We need to put a `document.pdf` file into the same path as `App.js` so that `PdfScreen` can show it.

Next, we’ll define our `App`. It simply renders our navigation stack:

JavaScript

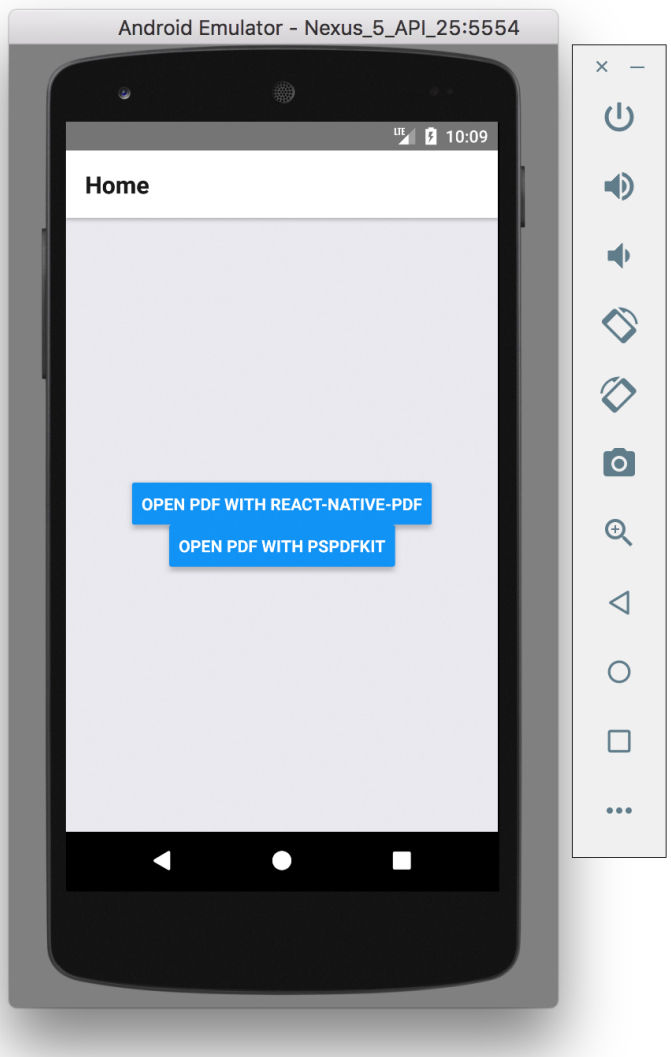
```
export default class App extends Component<{}> {
  render() {
    return <PdfApp />;
  }
}
```

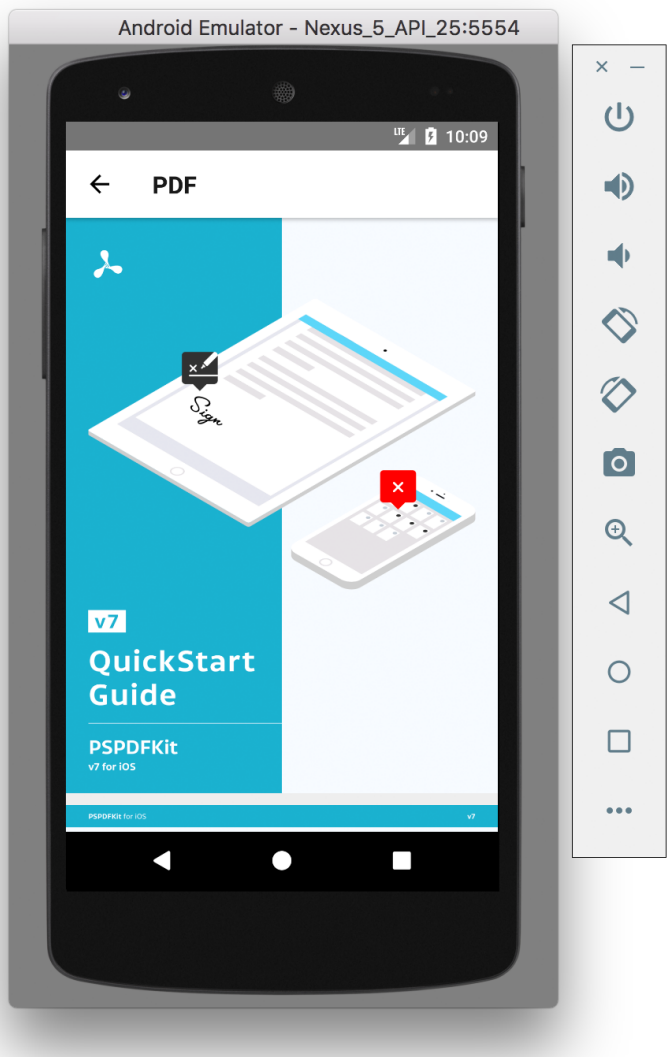
At the end of `App.js`, we will define our styles:

JavaScript

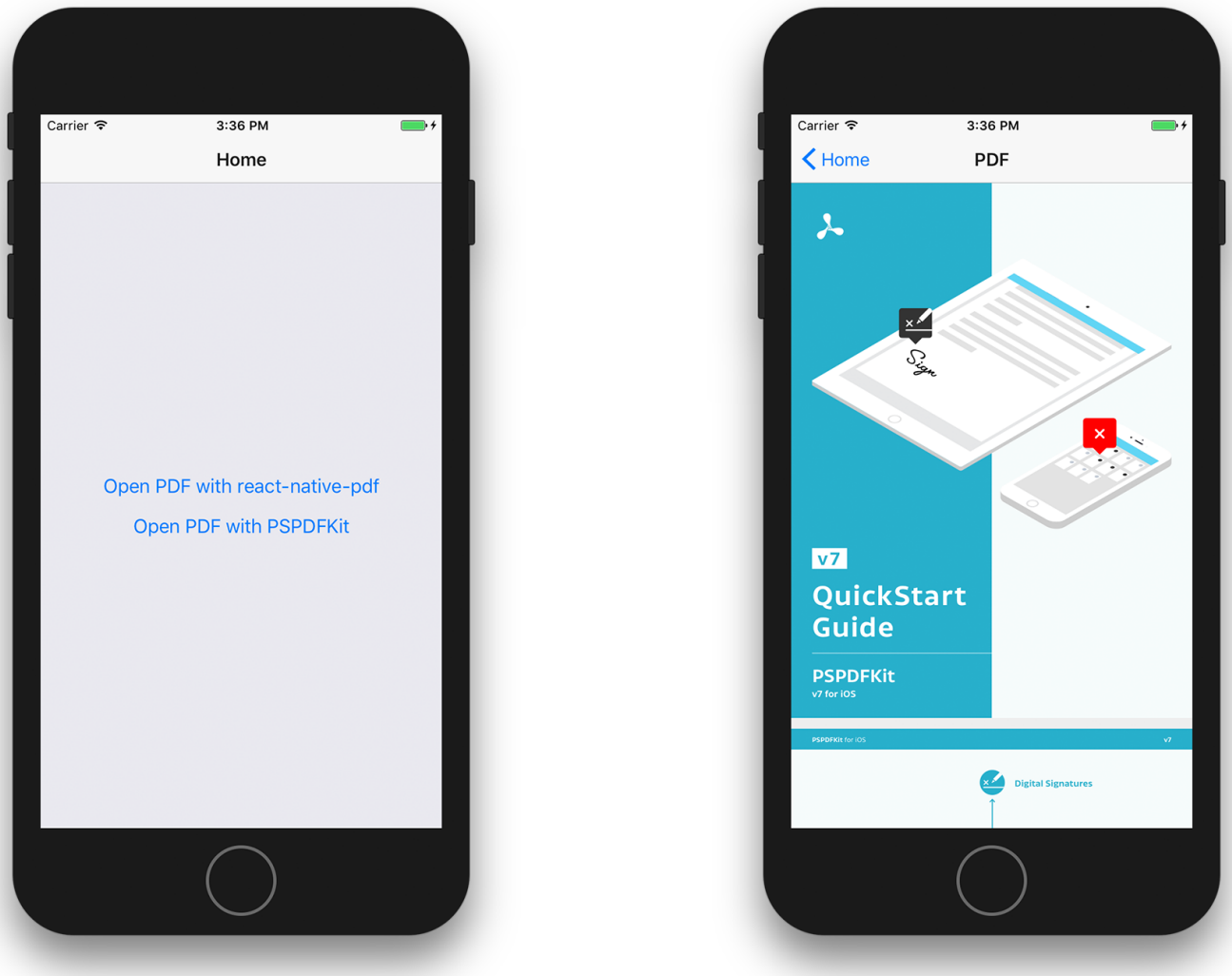
```
const styles = StyleSheet.create({
    container: {
        flex: 1,
        justifyContent: 'center',
        alignItems: 'center'
    },
    pdf: {
        flex: 1,
        width: Dimensions.get('window').width
    }
});
```

We center the “Open PDF” button and allow the PDF to fill the entire screen. Here’s how it looks on Android:





And here is how the app looks on iOS:



You can find the complete content of `App.js` on GitHub.

After these few steps, we can now tap on a button and scroll through a PDF document. However, we can't do anything else; there is no zooming and there are no annotations. We only have the scrolling view mode.

But that is where PSPDFKit comes in: It includes all of these features and more without you having to configure anything.

Opening a PDF with PSPDFKit

First, download a demo version of PSPDFKit and click on the "Download Trial" button. Then follow the integration guides for iOS and Android.

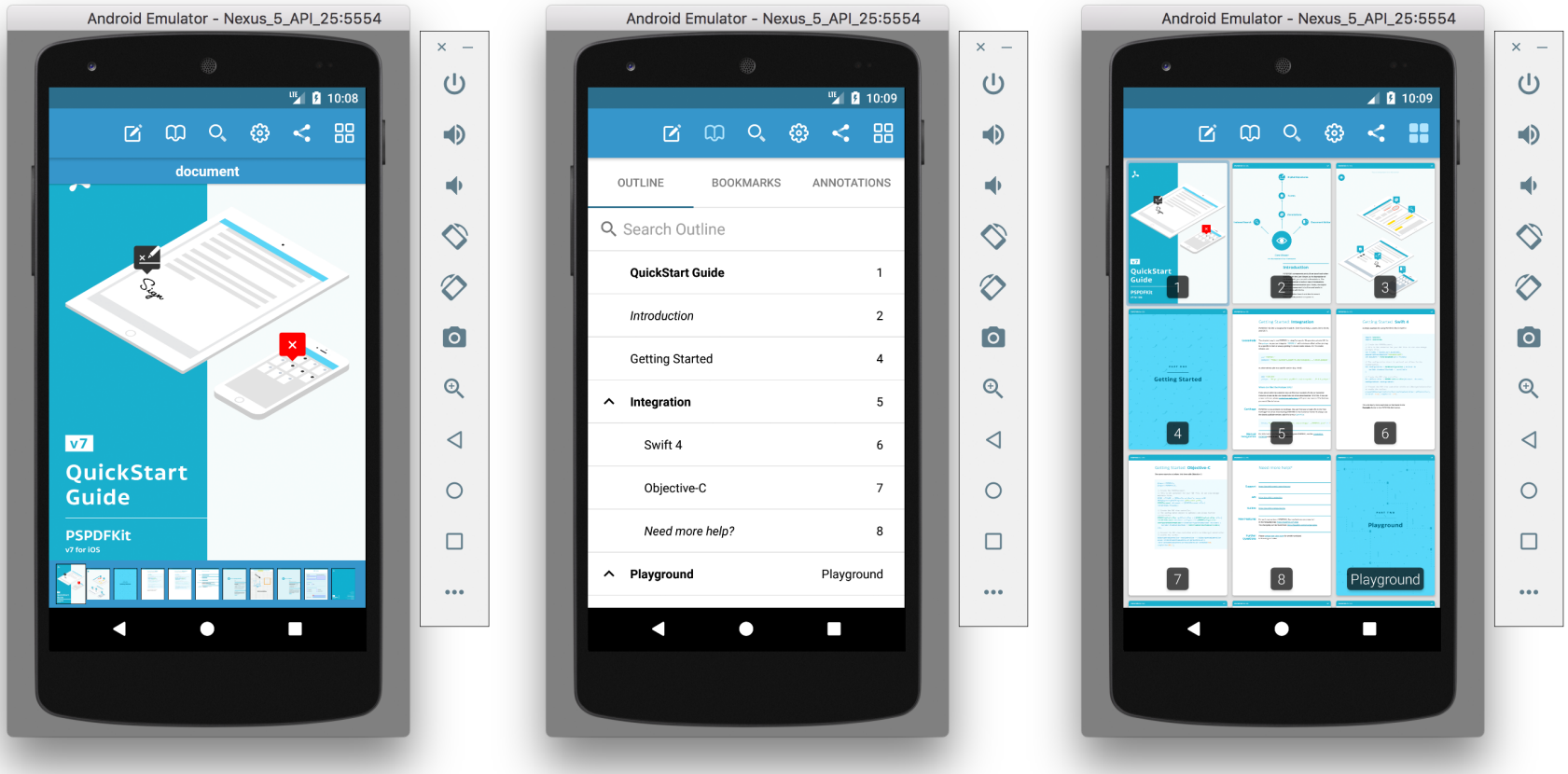
Once this is done, add a second button to `HomeScreen` that opens a PDF with PSPDFKit:

JavaScript

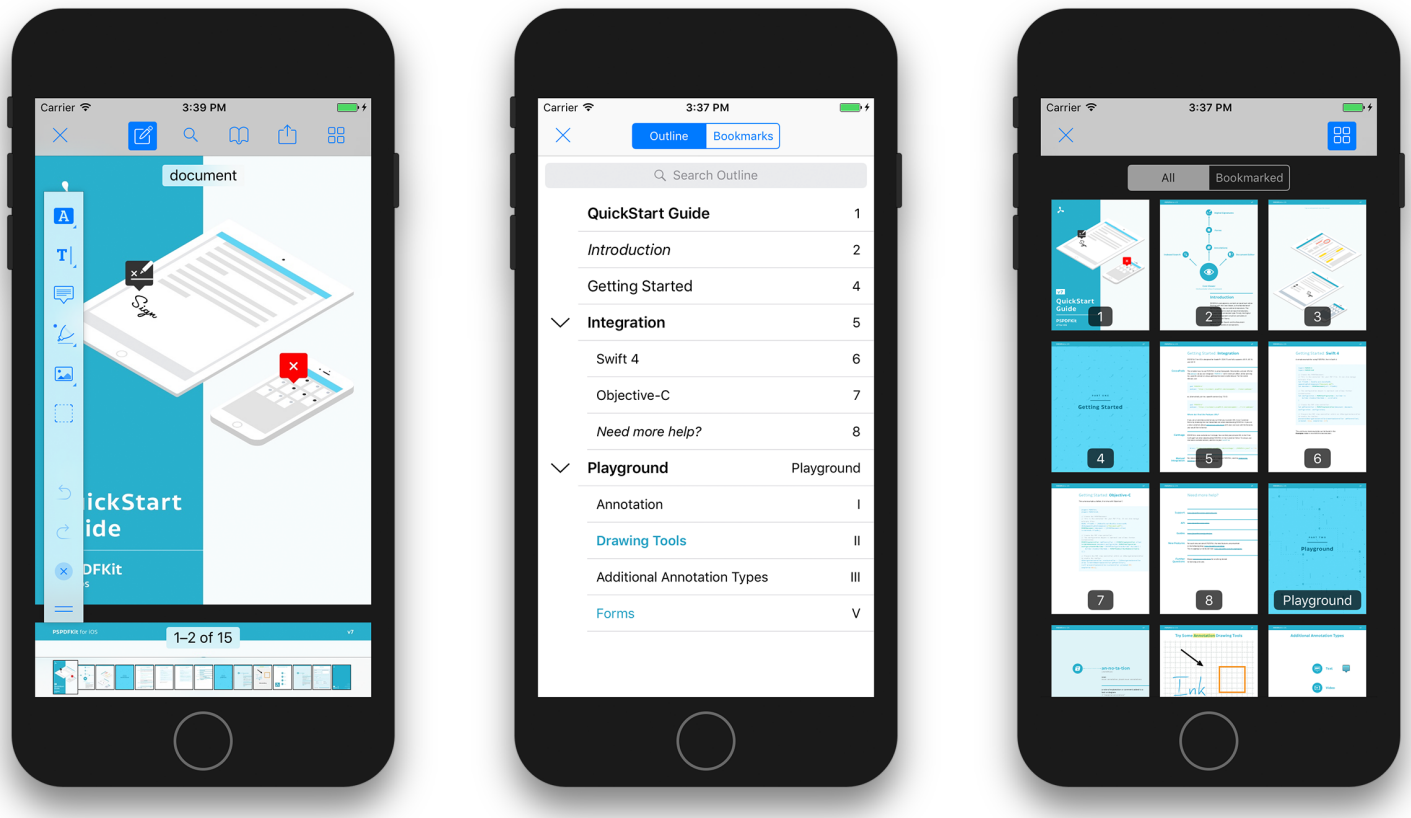
```
var PSPDFKit = NativeModules.PSPDFKit;

if (Platform.OS === 'ios') {
  PSPDFKit.setLicenseKey('INSERT_YOUR_LICENSE_KEY_HERE');
}

const DOCUMENT = Platform.OS === 'ios' ? 'document.pdf' : "file:///sdcard/document.pdf";

// Simple screen containing an "Open PDF" button
class HomeScreen extends Component {
  _presentPSPDFKit() {
    PSPDFKit.present(DOCUMENT, {
        pageTransition: 'scrollContinuous',
        scrollDirection: 'vertical'
    })
  }

  static navigationOptions = {
    title: 'Home'
  };

  render() {
    const { navigate } = this.props.navigation;
    return (
      <View style={styles.container}>
        <Button
          onPress={() => navigate('Pdf')}
          title='Open PDF with react-native-pdf'
        />
        <Button
          onPress={this._presentPSPDFKit}
          title='Open PDF with PSPDFKit'
        />
      </View>
    );
  }
}
```

All we need is `PSPDFKit.present('document.pdf')` and we can view a PDF document in PSPDFKit. Not only that, but we can also zoom, create annotations, look at the document's outline, and lots of other things. We can also customize the PDF viewer by passing a dictionary to `PSPDFKit.present`.

Our React Native app powered by PSPDFKit, as seen on Android:



And the same again on iOS:



Conclusion

As you saw, adding PDF support to your app with the `react-native-pdf` package isn't difficult, but in doing so, you're missing out on a lot of functionality. Meanwhile, PSPDFKit ships with many features out of the box, providing your users with a better user experience. PSPDFKit also comes with great customer support, so please reach out to us if you have any questions about our React Native integration.

Finished Implementation

You can find the source code for the entire project at GitHub.

PSPDFKit Newsletter

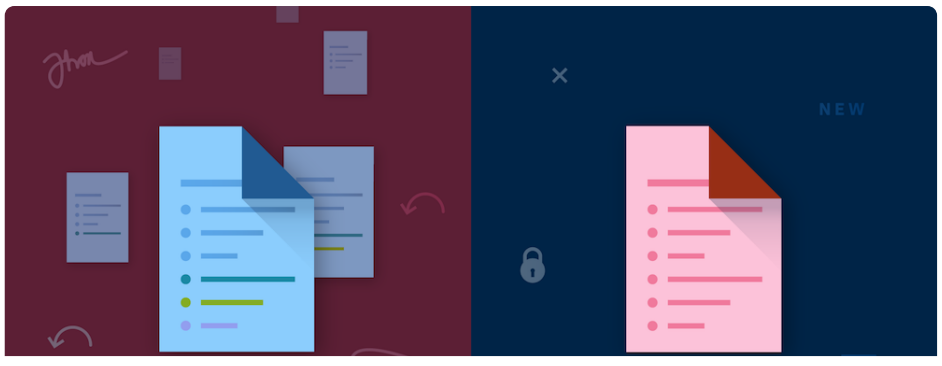
Subscribe to our newsletter for more articles like this.

name@company.com Subscribe

Related Articles

Open a PDF on the Universal Windows Platform

How Spaced Repetition Helps You Learn New Things



Incremental and Full Save in PDFs

Try Our Showcase App