React Native - Experience Documentation

Created	@Sep 14, 2020 4:20 PM
Experience Documentation	

Setting up Development environment

-Ensure you are running node version 12 or higher

to check - in command:

```
node -v
v12.18.3
```

Install Expo CLI globally

```
npm i -g expo-cli
...
+ expo-cli@3.27.4
added 1887 packages from 840 contributors in 126.288s
```

Added VS Code Extensions

- React Native Tools - for debugging react native app inside VS Code
- React-Native/React/Redux snippets for es6/es7 - for code snipped to code faster
- Prettier - for formatting code
- Material Icon Theme - for getting icon depending on their type

VS Code Setting

Code > Preferences > Settings

Search formatonsave

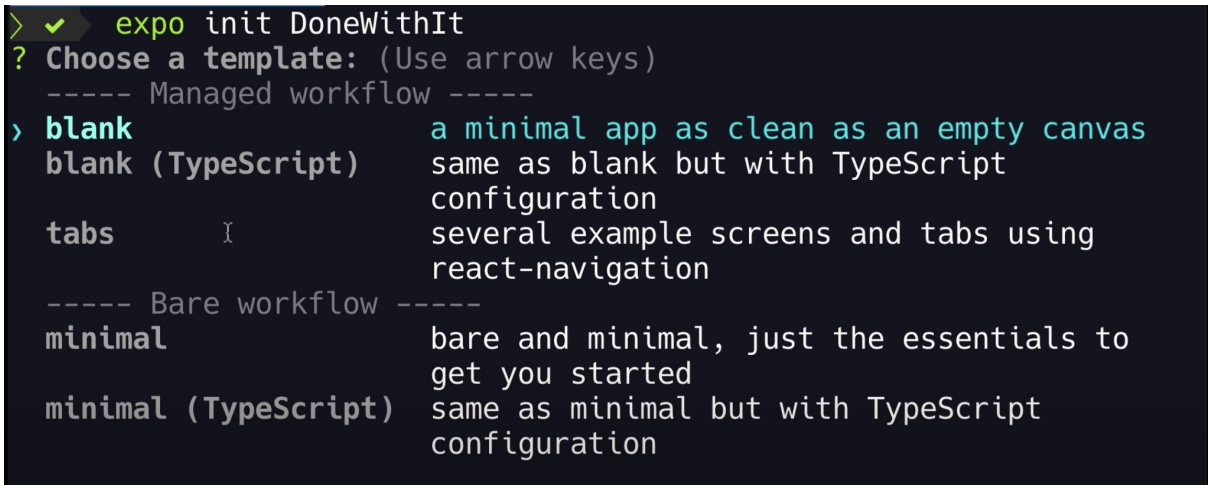
Ensure this is enabled - will enable your code to be formatted when file is saved

Development Environment now set up!

Create First Expo Project

In command line: (DoneWithIt - is just the name of the app from Youtube you can call it anything)

```
expo init DoneWithIt
```



Chose a workflow:

Managed workflow - will take care of all the complexity behind the scene therefore will not give you Android and iOS projects/folders (pure Javascript)

Bare workflow - bare-bone react native workflow so you will have the Android and iOS projects/folders

File setup now complete!

Starting Expo Server to Serve our App

In Terminal

```
npm start
```

This is open up your browser pointing to an address.

It will open up to Metro Bundler which is the JS bundler for React Native

Code Walkthrough starts from this section...

SafeAreaView

```
import { StyleSheet, Text, View, SafeAreaView } from 'react-native';

    <SafeAreaView style={styles.container}>
      <Text>The App is RUNNING!</Text>
      <StatusBar style="auto" />
    </SafeAreaView>
```

-import

-wrap around components

Text - numberOfLines

```
<SafeAreaView style={styles.container}>
      <Text numberOfLines={1}>
        The App is RUNNING! But this is a really really really long long line but its great
        </Text>
      <StatusBar style="auto" />
    </SafeAreaView>
```

-in opening text component

-can choose number of lines

Adding the press

```
export default function App() {
  const handlePress = () => console.log('text press')
  return (
    <SafeAreaView style={styles.container}>
      <Text numberOfLines={1} onPress={handlePress}>
        The App is RUNNING! But this is a really really really long long line but its great
        </Text>
      <StatusBar style="auto" />
    </SafeAreaView>
```

-will console.log when the text is pressed/clicked

Adding an Image

Locally

-image best placed in assets folder

```
import { StyleSheet, Text, View, SafeAreaView, Image } from 'react-native';

export default function App() {
  const handlePress = () => console.log('text press')
  return (
```

```
    <SafeAreaView style={styles.container}>
      <Text numberOfLines={1} onPress={handlePress}>
        The App is RUNNING! But this is a really really really long long line but its great
        </Text>
        <Image
        style={styles.profileImage}
        source={require('./assets/profile-photo.jpeg')}/>
      <StatusBar style="auto" />
    </SafeAreaView>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: 'aqua',
    alignItems: 'center',
    justifyContent: 'center',
  },
  profileImage: {
    width: 200,
    height: 300
  }
});
```

-NOTE: require function returns a number in reference to our image. To see this write the below console.log after the export default but before the return. Your console will tell you the number reference to this image.

```
console.log(require('./assets/profile-photo.jpeg'))
```

Network Images

-pass an object

-property call 'uri

-set to a string of the url

-need specificity of size either within the tag or in styles as illustrated below or the image will not show

```
import { StyleSheet, Text, View, SafeAreaView, Image } from 'react-native';

export default function App() {
  const handlePress = () => console.log('text press')
  return (
    <SafeAreaView style={styles.container}>
      <Text numberOfLines={1} onPress={handlePress}>
        The App is RUNNING! But this is a really really really long long line but its great
        </Text>
        <Image
        style={styles.profileImage}
        source={{uri:'https://picsum.photos/200/300'}}/>
      <StatusBar style="auto" />
    </SafeAreaView>
  );
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: 'aqua',
    alignItems: 'center',
    justifyContent: 'center',
  },
  profileImage: {
    width: 200,
    height: 300
  }
});
```

Buttons

-can type button in as component and it will auto import

-self closing

-need a title

-onPress to console if button is pressed

```
import { StyleSheet, Text, View, SafeAreaView, Image, Button, Alert } from 'react-native';

export default function App() {
  const handlePress = () => console.log('text press')
  return (
    <SafeAreaView style={styles.container}>
      <Text numberOfLines={1} onPress={handlePress}>
        The App is RUNNING! But this is a really really really long long line but its great
        </Text>
        <Image
        blurRadius={1}
        style={styles.profileImage}
        source={{uri:'https://picsum.photos/200/300'}}/>
        <Button title='Click me' onPress={() => console.log('button pressed')}/>
      <StatusBar style="auto" />
    </SafeAreaView>
  );
}
```

NOTE: to add a simple alert into the button replace

```
<Button title='Click me' onPress={() => console.log('button pressed')}/>
to
<Button title='Click me' onPress={() => alert('button pressed')}/>
```