



1. Dentro da pasta **app/pages** crie um arquivo chamado **CameraAccess.js**

Dentro de app/pages/CameraAccess.js

```
1 import React, { Component } from 'react';
2 import {
3   StyleSheet,
4   Text,
5   View,
6   Button,
7   CameraRoll,
8   Image
9 } from 'react-native';
```

Dentro de app/pages/CameraAccess.js




```
1 export default class CameraAccess extends Component {
2   static navigationOptions = ({ navigation }) => ({
3     drawerLabel: 'Acessar fotos',
4     title: 'Acessar fotos',
5     headerLeft:
6       <View style={{ marginLeft: 10 }}>
7         <Button title="menu" onPress={() => navigation.navigate('DrawerToggle')} />
8       </View>
9   });
10
11   constructor() {
12     super();
13
14     this.state = {
15       photos: []
16     };
17   }
18 }
```

Dentro de app/pages/CameraAccess.js, metodo render:



```
1 render() {
2   return (
3     <View style={styles.container}>
4       <View style={styles.btnContainer}>
5         <Button title="Buscar fotos" onPress={() => this.getPhotos()} />
6       </View>
7
8       <View style={styles.imagesContainer}>
9         {this.renderPhotos()}
10      </View>
11    </View>
12  );
13 }
```

Dentro de app/pages/CameraAccess.js, adicione os metodos:



```
getPhotos() {
  CameraRoll.getPhotos({
    first: 20,
    assetType: 'All',
    groupName: 'DCIM'
  })
  .then(photos => {
    console.log(photos);
    this.setState({ photos: photos.edges });
  });
}

renderPhotos() {
  return this.state.photos.map((photo, index) => {
    return <Image
      key={index}
      style={{
        width: 100,
        height: 100
      }}
      source={{ uri: photo.node.image.uri }}
      resizeMode="contain"
    />;
  });
}
```

Dentro de app/pages/CameraAccess.js:



```
1 const styles = StyleSheet.create({
2   container: {
3     flex: 1,
4     justifyContent: 'center',
5     alignItems: 'center',
6     backgroundColor: '#F5FCFF',
7   },
8   btnContainer: {
9     marginBottom: 10,
10    marginTop: 10
11  },
12  imagesContainer: {
13    flexDirection: 'row',
14    flexWrap: 'wrap',
15    justifyContent: 'space-around'
16  }
17 });
```

Dentro de app/pages/index.js, altere:



```
...  
  
import CameraAccess from './pages/CameraAccess';  
  
const Drawer = DrawerNavigator({  
  ...  
  
  CameraAccess: { screen: CameraAccess }  
}, {  
  contentComponent: props => <ScrollView><DrawerItems {...props} /></ScrollView>,  
});  
  
...
```




Veja o resultado!




1. Vamos usar o layout animation na prática

Dentro de app/pages/CameraAccess.js:



```
import React, { Component } from 'react';
import {
  StyleSheet,
  Text,
  View,
  Button,
  CameraRoll,
  Image,
  UIManager, // ← novo import
  LayoutAnimation // ← novo import
} from 'react-native';
import Share from 'react-native-share';
```

Dentro de app/pages/CameraAccess.js, no constructor altere:



```
constructor() {  
  super();  
  
  UIManager.setLayoutAnimationEnabledExperimental // <- nova linha  
  && UIManager.setLayoutAnimationEnabledExperimental(true); // <- nova linha  
  
  this.state = {  
    photos: []  
  };  
}
```

Dentro de app/pages/CameraAccess.js, no método getPhotos altere:



```
getPhotos() {  
  CameraRoll.getPhotos({  
    first: 20,  
    assetType: 'All',  
    groupName: 'DCIM'  
  })  
  .then(photos => {  
    console.log(photos);  
    LayoutAnimation.configureNext(LayoutAnimation.Presets.linear); // <- nova linha  
    this.setState({ photos: photos.edges });  
  });  
}
```



Veja o resultado!