

## Exercise (Instructions): Secure Store

### Objectives and Outcomes

In this exercise you will configure your app to make use of secure store from Expo SDK to enable storing and retrieving data in the form of key-value pairs. At the end of this lesson you will be able to:

- Configure your app with secure store
- Store and retrieve data using key-value pairs

### Using Secure Store

- Add a new file named LoginComponent.js in the components folder and add the following to it:

```

1  import React, { Component } from 'react';
2  import { View, Button, StyleSheet } from 'react-native';
3  import { Card, Icon, Input, CheckBox } from 'react-native-elements';
4  import { SecureStore } from 'expo';
5
6  class Login extends Component {
7
8    constructor(props) {
9      super(props);
10
11      this.state = {
12        username: '',
13        password: '',
14        remember: false
15      }
16    }
17
18    componentDidMount() {
19      SecureStore.getItemAsync('userinfo')
20        .then((value) => {
21          this.setState({username: value});
22        })
23        .catch((error) => console.log('Could not get user info', error));
24    };
25
26    handleLogin() {
27      console.log(JSON.stringify(this.state));
28      if (this.state.remember) {
29        SecureStore.setItemAsync('userinfo', JSON.stringify({username: this
30          .state.username, password: this.state.password}));
31        .catch((error) => console.log('Could not save user info', error
32          ));
33      }
34      else {
35        SecureStore.deleteItemAsync('userinfo')
36          .catch((error) => console.log('Could not delete user info',
37            error));
38      }
39    }
40
41    render() {
42      return (
43        <View style={styles.container}>
44          <Input
45            placeholder="Username"
46            leftIcon={{ type: 'font-awesome', name: 'user-o' }}
47            onChangeText={(username) => this.setState({username})}
48            value={this.state.username}
49            containerStyle={styles.formInput}
50          />
51          <Input
52            placeholder="Password"
53            leftIcon={{ type: 'font-awesome', name: 'key' }}
54            onChangeText={(password) => this.setState({password})}
55            value={this.state.password}
56            containerStyle={styles.formInput}
57          />
58          <CheckBox title="Remember Me"
59            checked={this.state.remember}
60          />
61        </View>
62      );
63    }
64  }
65
66  const styles = StyleSheet.create({
67    container: {
68      justifyContent: 'center',
69      margin: 20,
70    },
71    formInput: {
72      margin: 40,
73    },
74    formCheckbox: {
75      margin: 40,
76      backgroundColor: null
77    },
78    formButton: {
79      margin: 60,
80    }
81  });
82
83  export default Login;
```

- Then, update MainComponent.js to incorporate the LoginComponent as follows:

```

1  . . .
2
3  import Login from './LoginComponent';
4
5  . . .
6
18  headerTintColor: '#fff',
19  headerLeft: <Icon name="menu" size={24}
20    style={{color: 'white'}} />
```

```
20         iconStyle={{ color: white }}
21       onPress={() => navigation.toggleDrawer() } />
22     ))
23   );
24   . . .
25   . . .
26   . . .
27   Login:
28   { screen: LoginNavigator,
29     navigationOptions: {
30       title: 'Login',
31       type: 'font-awesome',
32       size: {24}
33       iconStyle={{ color: tintColor }}
34     } />
35   },
36   },
37   },
38   },
39   },
40   },
41   },
42   },
43   },
44   },
45   }, {
46     initialRouteName: 'Home',
47     drawerBackgroundColor: '#D1C4E9',
48     contentComponent: CustomDrawerContentComponent
49   }
50   . . .
51   . . .
```

- Save all the changes and do a Git commit with the message "Secure Store".

Conclusions

In this exercise you will learn to use the secure store from Expo SDK for secure storage of key-value pair based data

Mark as completed