

Exercise (Instructions): Forms

Objectives and Outcomes

In this exercise we will add a form to our app, and examine how to handle form submission. At the end of this exercise you will be able to:

- Create forms in your React Native app
- Handle form submissions

Adding a Form

- First, we will add a Date Picker module to our app as follows:

```
1 yarn add react-native-datepicker@1.7.2
```

- We will then add a new file named ReservationComponent.js and update its contents as follows:

```
15     }
16   }
17
18   static navigationOptions = {
19     title: 'Reserve Table',
20   };
21
22   handleReservation() {
23     console.log(JSON.stringify(this.state));
24     this.setState({
25       guests: 1,
26       smoking: false,
27       date: ''
28     });
29   }
30
31   render() {
32     return(
33       <ScrollView>
34         <View style={styles.formRow}>
35           <Text style={styles.formLabel}>Number of Guests</Text>
36           <Picker
37             style={styles.formItem}
38             selectedValue={this.state.guests}
39             onChange={(itemValue, itemIndex) => this.setState(
40               ({guests: itemValue})
41             >
42               <Picker.Item label="1" value="1" />
43               <Picker.Item label="2" value="2" />
44               <Picker.Item label="3" value="3" />
45               <Picker.Item label="4" value="4" />
46               <Picker.Item label="5" value="5" />
47               <Picker.Item label="6" value="6" />
48             </Picker>
49             <DatePicker
50               format=""
51               mode="datetime"
52               placeholder="select date and Time"
53               minDate="2017-01-01"
54               confirmBtnText="Confirm"
55               cancelBtnText="Cancel"
56               customStyles={{
57                 dateIcon: {
58                   position: 'absolute',
59                   left: 0,
60                   top: 4,
61                   marginLeft: 0
62                 },
63                 dateInput: {
64                   marginLeft: 36
65                 }
66               }}
67               onChange={(date) => {this.setState({date: date})}}
68             </DatePicker>
69           </View>
70           <View style={styles.formRow}>
71             <Button
72               onPress={() => this.handleReservation()}
73               title="Reserve"
74               color="#512DA8"
75               accessibilityLabel="Learn more about this purple button"
76             </Button>
77           </View>
78         </ScrollView>
79       );
80     }
81   }
82   formItem: {
83     flex: 1
84   }
```

```

111     }
112   });
113
114   export default Reservation;

```

- Then, we update MainComponent.js to integrate the new component into our app as follows:

```

1   . . .
2
3   import Reservation from './ReservationComponent';
4
5   . . .
6
7
8   const ReservationNavigator = createStackNavigator({
9     Reservation: { screen: Reservation }
10  }, {
11    navigationOptions: ({ navigation }) => ({
12      // ...
23    })
24  })
25
26  . . .
27
28  Reservation:
29    { screen: ReservationNavigator,
30      navigationOptions: {
31        title: 'Reserve Table',
32        drawerLabel: 'Reserve Table',
33        drawerIcon: ({ tintColor, focused }) => (
34          <Icon
35            name='cutlery'
36            type='font-awesome'
37            size={24}
38            iconStyle={{ color: tintColor }}
39          />
40        ),
41      }
42    }
43  . . .

```

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

Conclusions

In this exercise we have learnt how to add forms to our app.