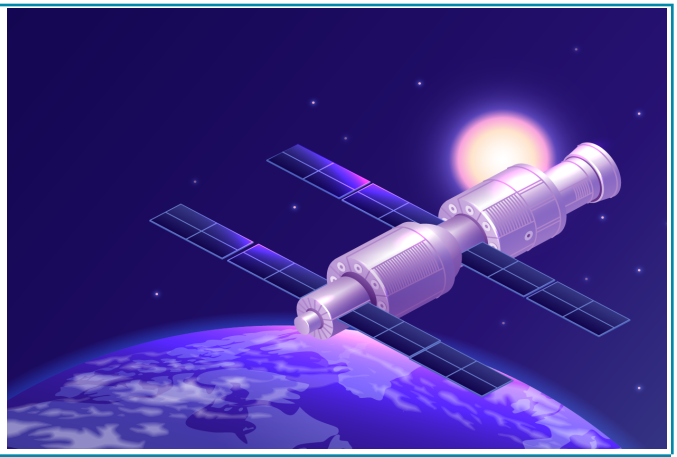
ISS LOCATION



What is our GOAL for this MODULE?

In this class, we learned to use the React Native maps library to add the world map to the app. And display the location of the ISS on the map.

What did we ACHIEVE in the class TODAY?

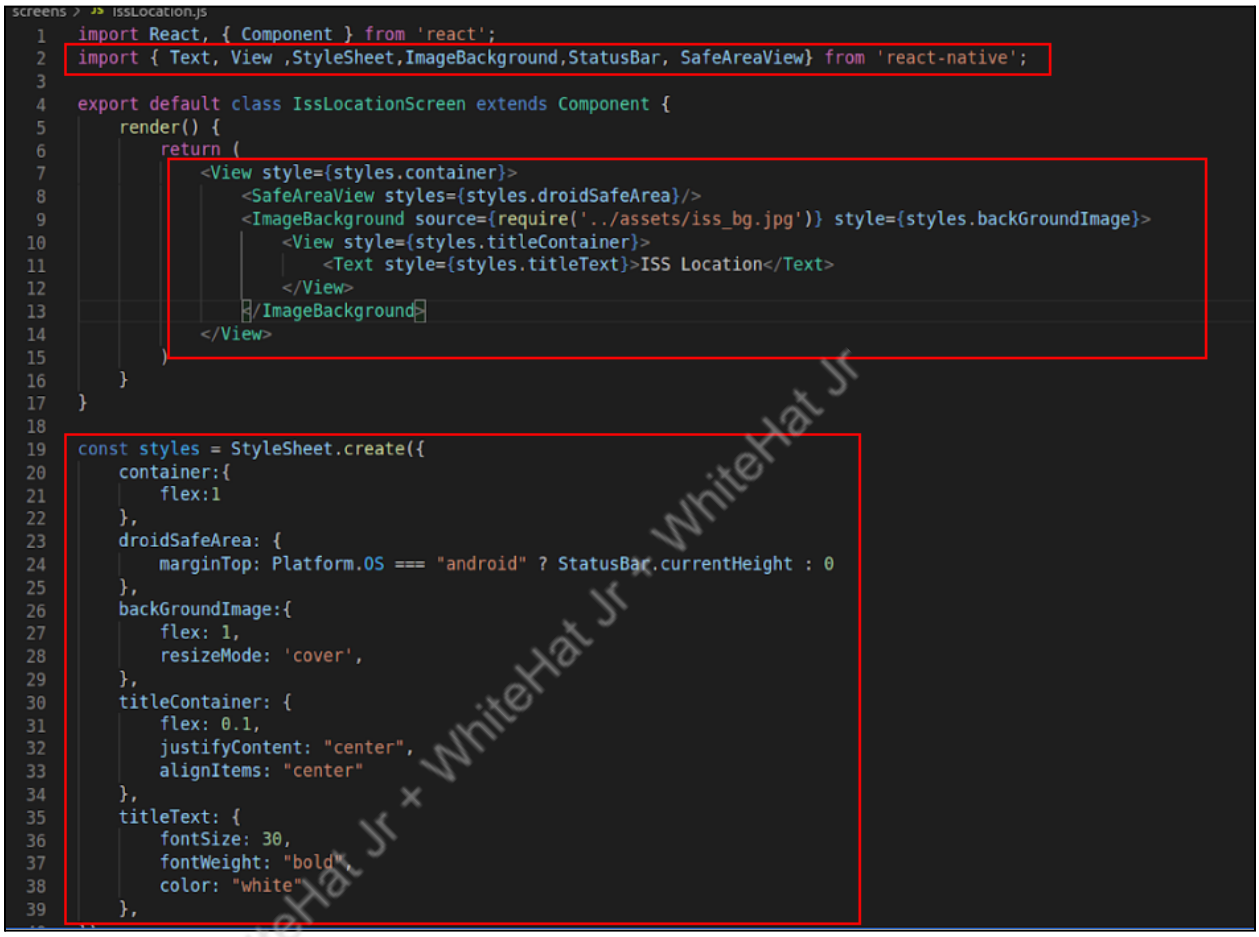
- We used the react native maps library to add maps to the app.
- Displayed the latitude, longitude of the ISS.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Usage of React Native library.

How did we DO the activities?

1. Add the image background, and styles to the title for the ISS location screen.



```
screens > IssLocation.js
1  import React, { Component } from 'react';
2  import { Text, View ,StyleSheet,ImageBackground,StatusBar, SafeAreaView} from 'react-native';
3
4  export default class IssLocationScreen extends Component {
5      render() {
6          return (
7              <View style={styles.container}>
8                  <SafeAreaView styles={styles.droidSafeArea}/>
9                  <ImageBackground source={require('../assets/iss_bg.jpg')} style={styles.backGroundImage}>
10                     <View style={styles.titleContainer}>
11                         <Text style={styles.titleText}>ISS Location</Text>
12                     </View>
13                 </ImageBackground>
14             </View>
15         )
16     }
17 }
18
19 const styles = StyleSheet.create({
20     container:{
21         flex:1
22     },
23     droidSafeArea: {
24         marginTop: Platform.OS === "android" ? StatusBar.currentHeight : 0
25     },
26     backGroundImage:{
27         flex: 1,
28         resizeMode: 'cover',
29     },
30     titleContainer: {
31         flex: 0.1,
32         justifyContent: "center",
33         alignItems: "center"
34     },
35     titleText: {
36         fontSize: 30,
37         fontWeight: "bold",
38         color: "white"
39     },
```

2. Install the **react native maps** library using the **expo** command.

```
ISS-Tracker$ expo install react-native-maps
```

3. Check the data on the satellites API.

```
{"name":"iss","id":25544,"latitude":-37.264086390933,"longitude":-47.601405349805,"altitude"
:431.75316730831,"velocity":27548.608651074,"visibility":"eclipsed","footprint":4566.7813851
089,"timestamp":1612153955,"daynum":2459246.689294,"solar_lat":-17.041142488018,"solar_lon":
115.247552949,"units":"kilometers"}
```

4. Install **axios** using **npm**.

```
ISS-Tracker$ npm add axios
```

5. Use **axios.get** to request data from the API.

```
export default class IssLocationScreen extends Component {
    constructor(props) {
        super(props);
        this.state = {
            location: {},
        };
    }

    componentDidMount() {
        this.getIssLocation()
    }

    getIssLocation = () => {
        axios
            .get("https://api.whereistheiss.at/v1/satellites/25544")
            .then(response => {
                this.setState({ location: response.data })
            })
            .catch(error => {
                Alert.alert(error.message)
            })
    }
```

6. Import the **MapView** and **Marker** from the **react-native-maps** to render the map on the screen.

```
import MapView, { Marker } from 'react-native-maps';
```

7. Use map view in render() to display the map to the screen.

```
        return (
            <View style={styles.container}>
                <SafeAreaView style={styles.droidSafeArea} />
                <ImageBackground source={require('../assets/iss_bg.jpg')} style={styles.backgroundImage}>
                    <View style={styles.titleContainer}>
                        <Text style={styles.titleText}>ISS Location</Text>
                    </View>
                    <View style={styles.mapContainer}>
                        <MapView
                            style={styles.map}
                            region={{
                                latitude: this.state.location.latitude,
                                longitude: this.state.location.longitude,
                                latitudeDelta: 100,
                                longitudeDelta: 100
                            }}
                        >
                        </MapView>
                    </View>
                </ImageBackground>
            </View>
        );
    }
  }
}
```

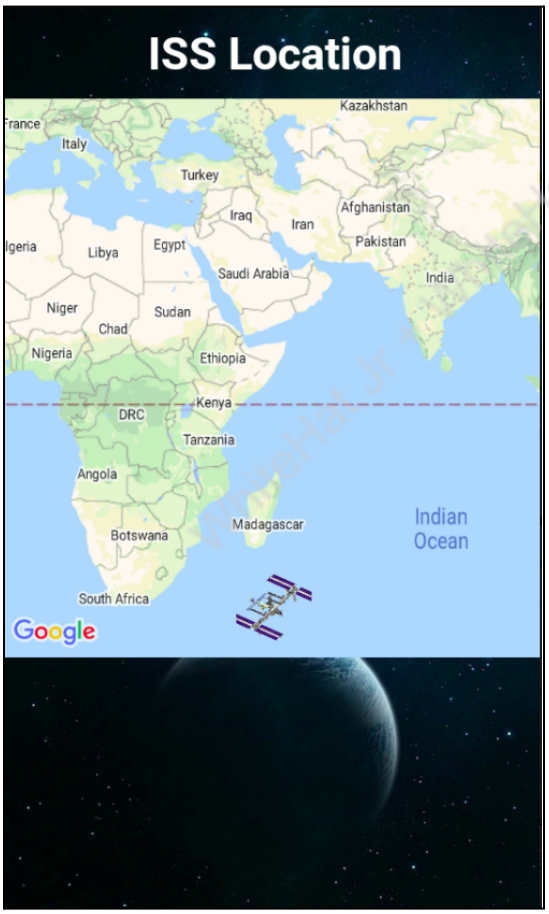
8. Add styles to the **MapView** component.

```
const styles = StyleSheet.create({
    container: {
        flex: 1
    },
    droidSafeArea: {
        marginTop: Platform.OS === "android" ? StatusBar.currentHeight : 0
    },
    backgroundImage: {
        flex: 1,
        resizeMode: 'cover',
    },
    titleContainer: {
        flex: 0.1,
        justifyContent: "center",
        alignItems: "center"
    },
    titleText: {
        fontSize: 30,
        fontWeight: "bold",
        color: "white"
    },
    refeshContainer: {
        flex: 0.1,
        justifyContent: "center",
        alignItems: "center"
    },
    mapContainer: {
        flex: 0.6
    },
    map: {
        width: "100%",
        height: "100%"
    }
});
```

9. Use the **Marker** to add the ISS location on the map.

```
<View style={styles.mapContainer}>
    <MapView
        style={styles.map}
        region={{
            latitude: this.state.location.latitude,
            longitude: this.state.location.longitude,
            latitudeDelta: 100,
            longitudeDelta: 100
        }}
    >
        <Marker
            coordinate={{ latitude: this.state.location.latitude, longitude: this.state.location.longitude }}
        >
            <Image source={require('../assets/iss_icon.png')} style={{ height: 50, width: 50 }} />
        </Marker>
    </MapView>
</View>
```

10. Run the code to check the output.



11. Display the Map Information on the screen as follows:

```
<View style={styles.infoContainer}>
    <Text style={styles.infoText}>Latitude: {this.state.location.latitude}</Text>
    <Text style={styles.infoText}>Longitude: {this.state.location.longitude}</Text>
    <Text style={styles.infoText}>Altitude (KM): {this.state.location.altitude}</Text>
    <Text style={styles.infoText}>Velocity (KM/H): {this.state.location.velocity}</Text>
</View>
```

And Add Styles for it -

```
infoContainer: {
    flex: 0.2,
    backgroundColor: 'white',
    marginTop: -10,
    borderTopLeftRadius: 30,
    borderTopRightRadius: 30,
    padding: 30
},
infoText: {
    fontSize: 15,
    color: "black",
    fontWeight: "bold"
}
```


Output:

What's NEXT?

In the next class, we will be working on creating the meteor screen.

EXTEND YOUR KNOWLEDGE:

1. Bookmark the following link to know more about MapView: https://docs.expo.io/versions/latest/sdk/map-view/