**Platform specific code**

A big benefit of React Native is that we don't need to worry about whether the application is run on a Android or iOS device. However, there might be cases where we need to execute *platform specific code*. Such case could be for example using a different implementation of a component on a different platform.

We can access the user's platform through the Platform.OS constant:

import { React } from 'react';

import { Platform, Text, StyleSheet } from 'react-native';

const styles = StyleSheet.create({

text: {

color: Platform.OS === 'android' ? 'green' : 'blue',

},

});

const WhatIsMyPlatform = () => {

return <Text style={styles.text}>Your platform is: {Platform.OS}</Text>;

};

Possible values for the Platform.OS constant are android and ios. Another useful way to define platform specific code branches is to use the Platform.select method. Given an object where keys are one of ios, android, native and default, the Platform.select method returns the most fitting value for the platform the user is currently running on. We can rewrite the styles variable in the previous example using the Platform.select method like this:

const styles = StyleSheet.create({

text: {

color: Platform.select({

android: 'green',

ios: 'blue',

default: 'black',

}),

},

});

We can even use the Platform.select method to require a platform specific component:

const MyComponent = Platform.select({

ios: () => require('./MyIOSComponent'),

android: () => require('./MyAndroidComponent'),

})();

<MyComponent />;

However, a more sophisticated method for implementing and importing platform specific components (or any other piece of code) is to use the *.ios.jsx* and *.android.jsx* file extensions. Note that the *.jsx* extension can as well be any extensions recognized by the bundler, such as *.js*. We can for example have files *Button.ios.jsx* and *Button.android.jsx* which we can import like this:

import Button from './Button';

const PlatformSpecificButton = () => {

return <Button />;

};

Now, the Android bundle of the application will have the component defined in the *Button.android.jsx* whereas the iOS bundle the one defined in the *Button.ios.jsx* file.