



## EXPLORER

## ▶ OPEN EDITORS

## ▲ WDI

## ▶ \_\_tests\_\_

## ▶ .vscode

## ▶ android

## ▲ fancy-native-module

## ▲ android

## ▶ .gradle

## ▶ build

## ▲ src

## ▲ main

## ▲ java

## ▲ pl

## ▲ wp

## ▲ wdi

≡ FancyManager.kt

≡ FancyPackage.kt

≡ FancyViewManager.kt

## ▲ res

## ▲ layout

📄 AndroidManifest.xml

🔥 .classpath

🔄 build.gradle

## ▶ ios

## ▶ DOCKER

## ▶ CODE OUTLINE

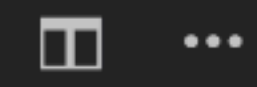
## ▶ MAVEN PROJECTS



≡ FancyPackage.kt x



```
1  package pl.wp.wdi
2
3  import com.facebook.react.ReactPackage
4  import com.facebook.react.bridge.NativeModule
5  import com.facebook.react.bridge.ReactApplicationContext
6  import com.facebook.react.uimanager.ViewManager
7  import java.util.*
8
9  class FancyPackage : ReactPackage {
10     ... override fun createNativeModules(reactContext: ReactApplicationContext): List<NativeModule> {
11         ...     val modules = ArrayList<NativeModule>()
12         ...     modules.add(FancyManager(reactContext))
13         ...     return modules
14     ... }
15
16     ... override fun createViewManagers(reactContext: ReactApplicationContext): List<ViewManager<*, *>> {
17         ...     val modules = ArrayList<ViewManager<*, *>>()
18         ...     modules.add(FancyViewManager() as ViewManager<*, *>)
19         ...     return modules
20     ... }
21
22 }
```



## EXPLORER

## ▶ OPEN EDITORS

## ◀ WDI

- ▶ \_\_tests\_\_
- ▶ .vscode
- ▶ android
- ▶ fancy-native-module
- ▶ ios
- ▶ node\_modules
- 6 .babelrc
- ≡ .buckconfig
- ≡ .flowconfig
- ◆ .gitattributes
- ◆ .gitignore
- { } .watchmanconfig

JS App.js

{ } app.json

JS index.js

{ } package.json

≡ yarn-error.log

👤 yarn.lock

## ▶ DOCKER

## ▶ CODE OUTLINE

## ▶ MAVEN PROJECTS



JS App.js x

```
1  /**
2   * Sample React Native App
3   * https://github.com/facebook/react-native
4   * @flow
5   */
6
7  import React, { Component } from "react";
8  import { Platform, StyleSheet, Text, View } from "react-native";
9
10 import { requireNativeComponent, NativeModules } from "react-native";
11
12 class FancyViewManager extends React.Component {
13   render() {
14     return <RNFancyViewManager {...this.props} />;
15   }
16 }
17
18 FancyViewManager.propTypes = {};
19
20 const RNFancyViewManager = requireNativeComponent("FancyViewManager", FancyViewManager, {
21   nativeOnly: {
22     nativeID: true,
23     accessibilityComponentType: true,
24     onLayout: true,
25     testID: true,
26     importantForAccessibility: true,
27     accessibilityLiveRegion: true,
28     accessibilityLabel: true,
29     renderToHardwareTextureAndroid: true,
30   },
31 });
32
33 const instructions = Platform.select({
34   ios: "Press Cmd+R to reload.\n" + "Cmd+D or shake for dev menu"
```