FancyPackage.kt — wdi
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

```kotlin
package pl.wp.wdi

import com.facebook.react.ReactPackage
import com.facebook.react.bridge.NativeModule
import com.facebook.react.bridge.ReactApplicationContext
import com.facebook.react.uimanager.ViewManager
import java.util.*

class FancyPackage : ReactPackage {
    override fun createNativeModules(reactContext: ReactApplicationContext): List<NativeModule> {
        val modules = ArrayList<NativeModule>()
        modules.add(FancyManager(reactContext))
        return modules
    }

    override fun createViewManagers(reactContext: ReactApplicationContext): List<ViewManager<*, *>> {
        val modules = ArrayList<ViewManager<*, *>>()
        modules.add(FancyViewManager() as ViewManager<*, *>)
        return modules
    }

}
```

App.js — wdi
EXPLORER
OPEN EDITORS
WDI
__tests__
.vscode
android
fancy-native-module
ios
node_modules
.babelrc
.buckconfig
.flowconfig
.gitattributes
.gitignore
.watchmanconfig
App.js
app.json
index.js
package.json
yarn-error.log
yarn.lock
DOCKER
CODE OUTLINE
MAVEN PROJECTS
App.js

```
/**
 * Sample React Native App
 * https://github.com/facebook/react-native
 * @flow
 */

import React, { Component } from "react";
import { Platform, StyleSheet, Text, View } from "react-native";

import { requireNativeComponent, NativeModules } from "react-native";

class FancyViewManager extends React.Component {
  render() {
    return <RNFancyViewManager {...this.props} />;
  }
}

FancyViewManager.propTypes = {};

const RNFancyViewManager = requireNativeComponent("FancyViewManager", FancyViewManager, {
  nativeOnly: {
    nativeID: true,
    accessibilityComponentType: true,
    onLayout: true,
    testID: true,
    importantForAccessibility: true,
    accessibilityLiveRegion: true,
    accessibilityLabel: true,
    renderToHardwareTextureAndroid: true,
  },
});

const instructions = Platform.select({
  ios: "Press Cmd+R to reload,\n" + "Cmd+D or shake for dev menu",
```