



EXPLORER

▶ OPEN EDITORS

▲ WDI

▶ __tests__

▶ .vscode

▶ android

▲ fancy-native-module

▲ android

▶ .gradle

▶ build

▲ src

▲ main

▲ java

▲ pl

▲ wp

▲ wdi

≡ FancyManager.kt

≡ FancyPackage.kt

▲ res

📄 AndroidManifest.xml

🔥 .classpath

🔄 build.gradle

▶ ios

▶ ios

▶ node_modules

🔗 babelrc

▶ DOCKER

▶ CODE OUTLINE

▶ MAVEN PROJECTS



≡ FancyManager.kt x



```
1  package pl.wp.wdi
2
3  import com.facebook.react.bridge.ReactApplicationContext
4  import com.facebook.react.bridge.ReactContextBaseJavaModule
5  import com.facebook.react.bridge.ReactMethod
6
7  import android.util.Log
8
9  class FancyManager(reactContext: ReactApplicationContext): ReactContextBaseJavaModule(reactContext) {
10     ... override fun getName(): String {
11     ...         return "FancyNativeModule"
12     ...     }
13
14     ... /*
15     ...  GW: Manager class must have at least one method to be visible as
16     ...  NativeModule!
17     ...  */
18     ... @ReactMethod
19     ... fun saySomething(message: String) {
20     ...         Log.d("TAG", message)
21     ...     }
22 }
```



EXPLORER

▶ OPEN EDITORS

◀ WDI



main

◀ java

◀ pl

◀ wp

◀ wdi

≡ FancyManager.kt

≡ FancyPackage.kt

◀ res

AndroidManifest.xml

.classpath

build.gradle

▶ ios

▶ ios

▶ node_modules

.babelrc

.buckconfig

.flowconfig

.gitattributes

.gitignore

{ } .watchmanconfig

JS App.js

{ } app.json

JS index.js

▶ DOCKER

▶ CODE OUTLINE

▶ MAVEN PROJECTS



JS App.js

≡ FancyManager.kt x

```
1  package pl.wp.wdi
2
3  import com.facebook.react.bridge.ReactApplicationContext
4  import com.facebook.react.bridge.ReactContextBaseJavaModule
5  import com.facebook.react.bridge.ReactMethod
6  import com.facebook.react.bridge.Promise;
7
8  import android.util.Log
9
10 class FancyManager(reactContext: ReactApplicationContext): ReactContextBaseJavaModule(reactContext) {
11     ... override fun getName(): String {
12     ...         return "FancyNativeModule"
13     ...     }
14
15     ... /*
16     ...  * GW: Manager class must have at least one method to be visible as
17     ...  * NativeModule!
18     ...  */
19     ... @ReactMethod
20     ... fun addNumbers(first: Int, second: Int, promise: Promise) {
21     ...         promise.resolve(first + second);
22     ...     }
23 }
```