



Version: 2.6.0 – 2.12.0

# Migrating off RNGHEnabledRootView

## Update MainActivity.java

Update your `MainActivity.java` file (or wherever you create an instance of `ReactActivityDelegate`), so that it no longer overrides the method responsible for creating `ReactRootView` instance, or modify it so that it no longer uses `RNGestureHandlerEnabledRootView`. Do not forget to remove import for `RNGestureHandlerEnabledRootView`:

```
package com.swmansion.gesturehandler.react.example;

import com.facebook.react.ReactActivity;
- import com.swmansion.gesturehandler.react.RNGestureHandlerEnabledRootView;
public class MainActivity extends ReactActivity {

-   @Override
-   protected ReactActivityDelegate createReactActivityDelegate() {
-       return new ReactActivityDelegate(this, getMainComponentName()) {
-           @Override
-           protected ReactRootView createRootView() {
-               return new RNGestureHandlerEnabledRootView(MainActivity.this);
-           }
-       };
-   }
}
```

## Check if your app works correctly

Some libraries (for example React Navigation) already use `GestureHandlerRootView` as a wrapper to enable gesture interactions. In that case you don't have to add one yourself. If gestures in your app work as expected after removing `RNGestureHandlerEnabledRootView` you can skip the next step.

## Update your JS code

Instead of using `RNGestureHandlerEnabledRootView` wrap your entry point with `<GestureHandlerRootView>` or `gestureHandlerRootHOC`, for example:

```
export default function App() {  
  return <GestureHandlerRootView style={{ flex: 1 }}>{/* content */}  
</GestureHandlerRootView>;  
}
```

### ! INFO

Note that `GestureHandlerRootView` acts like a normal `View`. So if you want it to fill the screen, you will need to pass `{ flex: 1 }` like you'll need to do with a normal `View`. By default, it'll take the size of the content nested inside.