



Pulling It All Together

Previously, you have covered many discrete Android-to-iOS mapping topics. They are purposely self-contained with very few classes in the individual Xcode projects so they can serve as independent sample projects. In the real programming world, it is the combination of features and use cases that makes your app useful and entertaining. You will definitely need to apply more than one mapping guideline to complete a meaningful app.

To walk you through the whole porting process, you will port a fully functioning Android app from start to finish by applying the mapping topics from Part II:

1. Create the runnable storyboard using the counterpart live Android app as the wireframe.
2. Port the Android classes and classes' member signatures to Swift first. Preserve any signatures if possible.
3. Fill in the blanks one method at a time. The caller, receivers, and “dots” mostly will just connect to each other without any glitch by blindly translating the Java statements or expressions to Swift at the method level.

Nothing will really be new here; you will systematically repeat the same top-down development approach you have been taking, and implement one piece at a time.