

Step it up: Compose for Desktop



Nishant Srivastava

www.nisrulz.com

Origin

Jetpack Compose



Jetpack Compose

- ⦿ Modern Toolkit
- ⦿ API inspired by React and Flutter
- ⦿ Declarative and Reactive UI Framework
- ⦿ Simplify UI development
- ⦿ Built using Kotlin Language
- ⦿ Currently in **ALPHA** 🤪



Jetpack Compose



New Project

Templates

- Phone and Tablet
- Wear OS
- Android TV
- Automotive
- Android Things

No Activity

Basic Activity

Bottom Navigation Activity

Empty Compose Activity

Previous Next Cancel Finish

Jetpack Compose



```
@Composable
fun helloWorld() {
    Text(text = "Hello World!")
}
```

Jetpack Compose



```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContent {  
            helloWorld()  
        }  
    }  
}
```

Jetpack Compose



```
@Preview(showBackground = true)
@Composable
fun DefaultPreview() {
    MyApplicationTheme {
        helloWorld()
    }
}
```

Hello World!

Jetpack Compose



Compose Compiler Plugin



Jetpack Compose

Jetpack Compose



Compose Compiler Plugin



Jetpack Compose
(Android)

Compose



Compose Compiler Plugin



Compose for Desktop

Compose



Compose Compiler Plugin

+



Jetpack Compose
(Android)

Compose for Desktop

Compose for Desktop



Compose for Desktop

- ⦿ Everything Jetpack Compose for Android has
- ⦿ Simplified Desktop UI Development
- ⦿ Targets only JVM right now
- ⦿ First public milestone was in Nov 2020
- ⦿ Currently in **ALPHA** 🤪



Compose for Desktop



- ⦿ Everything Jetpack Compose for Android has
- ⦿ Simplified Desktop UI Development
- ⦿ Targets only JVM right now
- ⦿ First public milestone was in Nov 2020
- ⦿ Currently in **ALPHA** 😲
- ⦿ ..is also called "Jetpack Compose for Desktop"

Compose for Desktop



New Project

Name:

Location:

Project Template:

- Application
 - Library
 - Native Application
 - Full-Stack Web Application
- Kotlin/JS
 - Browser Application
 - React Application
 - Node.JS Application
- Jetpack Compose for Desktop (Experimental)
 - Desktop uses Kotlin 1.4.20**
 - Multiplatform uses Kotlin 1.4.20

A Compose application targeting the desktop JVM platform (Windows, Linux, macOS)

Build System: Gradle Kotlin Gradle Groovy IntelliJ Maven

Project JDK:

Project Structure

- Project
 - compose Module

Compose for Desktop



```
import androidx.compose.material.Text
import androidx.compose.runtime.Composable

@Composable
fun helloWorld() {
    Text(text = "Hello World!")
}
```

Compose for Desktop



```
import androidx.compose.material.Text
import androidx.compose.runtime.Composable

@Composable
fun helloWorld() {
    Text(text = "Hello World!")
}
```

Compose for Desktop



```
import androidx.compose.desktop.Window

fun main() = Window {
    helloWorld()
}
```

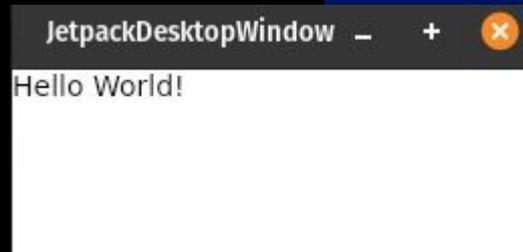
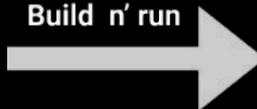
Compose for Desktop



```
import androidx.compose.desktop.Window

fun main() = Window {
    helloWorld()
}
```

Build n' run



Note: There is no setup right now to Preview, like in Android! 😞

Compose for Desktop



```
compose.desktop {
    application {
        mainClass = "MainKt"
        nativeDistributions {
            targetFormats(TargetFormat.Dmg, TargetFormat.Msi, TargetFormat.Deb)
            packageName = "ComposeDesktopExample"
        }
    }
}
```

Compose for Desktop



```
compose.desktop {  
    application {  
        mainClass = "MainKt"  
        nativeDistributions {  
            targetFormats(TargetFormat.Dmg, TargetFormat.Msi, TargetFormat.Deb)  
            packageName = "ComposeDesktopExample"  
        }  
    }  
}
```

Compose for Desktop



```
enum class TargetFormat(  
    internal val id: String, private vararg val compatibleOSs: OS) {  
    AppImage("app-image", *OS.values()),  
    Deb("deb", OS.Linux),  
    Rpm("rpm", OS.Linux),  
    Dmg("dmg", OS.MacOS),  
    Pkg("pkg", OS.MacOS),  
    Exe("exe", OS.Windows),  
    Msi("msi", OS.Windows);  
    ...  
}
```

Compose for Desktop

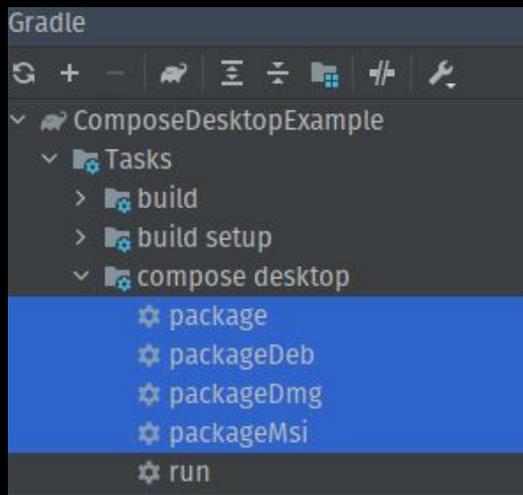


```
# Binaries are generated at  
# Path: {projectName}/build/compose/binaries/main/{packageType}  
  
./gradlew package
```

Compose for Desktop



```
# Binaries are generated at  
# Path: {projectName}/build/compose/binaries/main/{packageType}
```



Why?

Why?

- ⦿ Traditional Desktop UI frameworks
 - SWING
 - AWT
 - Callback driven
- ⦿ Electron
- ⦿ Qt, GTK, etc.



So what's the magic?

So what's the magic?

- ◎ Skia
 - 2D rendering library
 - Open sourced under BSD
 - Maintained by Google
 - Compose is using Skija (Java bindings for Skia)



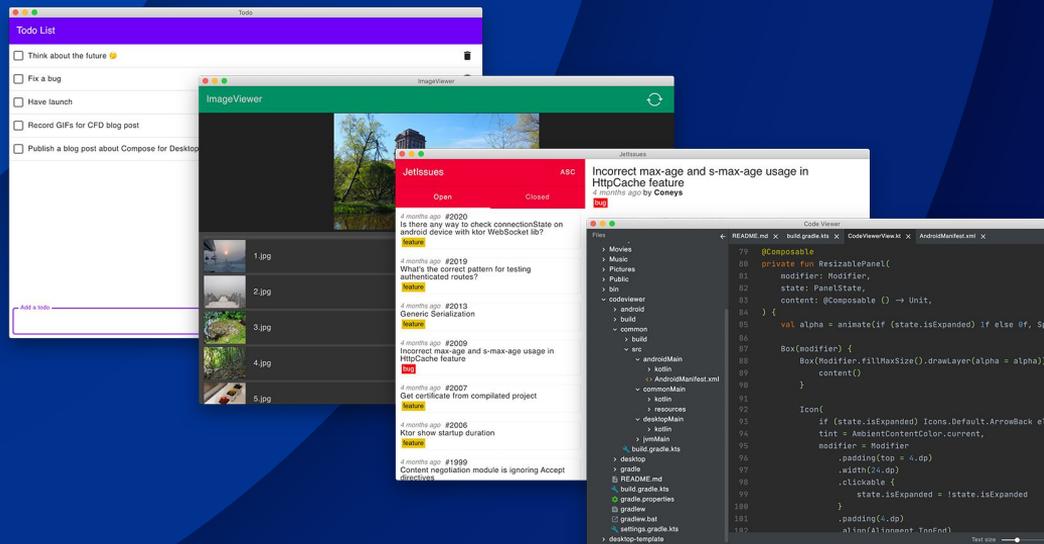
What's next?

What's Next?

- ⦿ First Jetpack Compose for Android, then Compose for Desktop
- ⦿ Testing setup
 - Possible: Screenshot Testing, Mocking
- ⦿ Should work in theory on GraalVM
- ⦿ Target native platforms



What's Next?



<https://github.com/JetBrains/compose-jb/tree/master/examples>

Links

<https://skia.org>

<https://github.com/JetBrains/skija>

<https://github.com/jetbrains/compose-jb>



Credits

- © Presentation template by [SlidesCarnival](#)

Thanks!

Any questions?



Nishant Srivastava

www.nisrulz.com