

# **Kotlin-based Build Scripts**

---

**New Opportunities in Gradle 3.0**

Tim Riemer


# About me

Tim Riemer

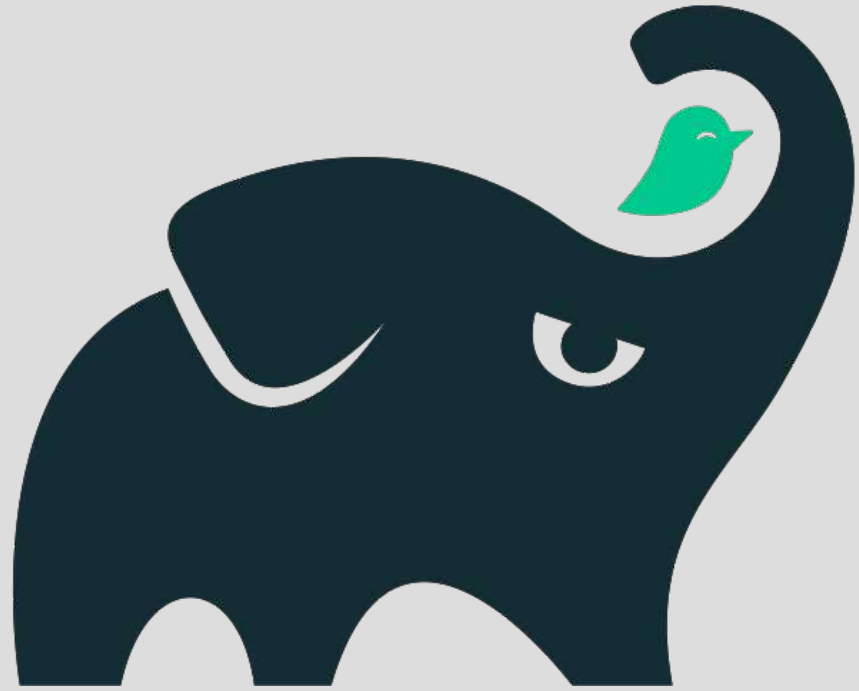
Integration Architect @ FNT GmbH

Java since 20 years

More than 10 years for a living

 @zordan\_f

 TimRiemer



# Gradle

Build Automation System

Written mostly in Groovy

Groovy DSL

Initial Release in 2007

Current Version 3.4.1

# Since 3.0

Daemon enabled by default (3.0)

Composite Builds (3.1)

Incremental Builds Improvements

Compile Avoidance (3.4)

Java-Library plugin (3.4)



# Kotlin

Statically-typed

Based on the JVM

Java interoperability

Developed by JetBrains since 2010

First Version 2012

Current Release 1.1.1

# Kotlin

Null-pointer safety

Extension functions

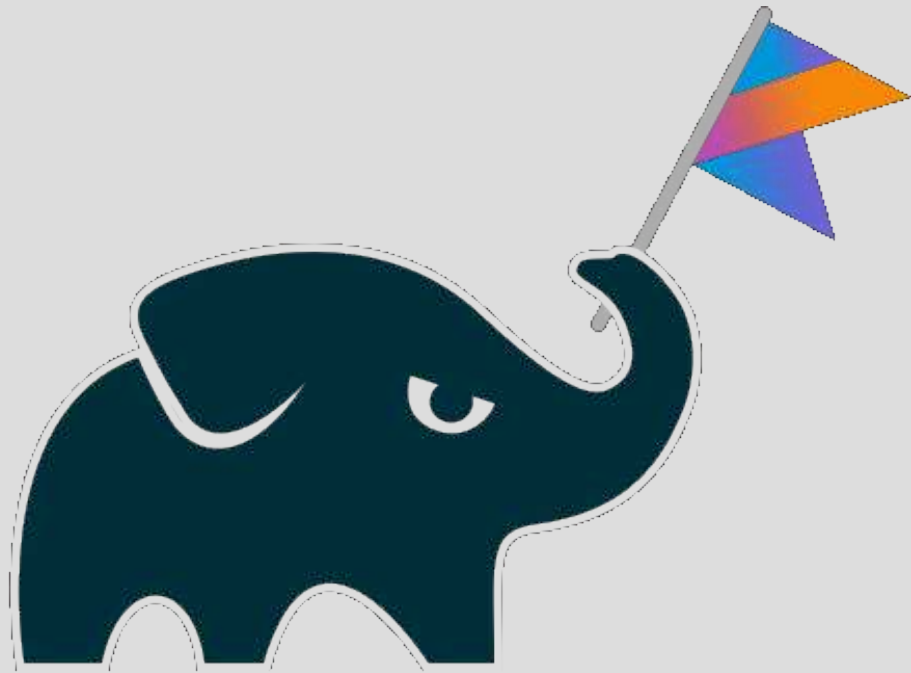
Infix notation

Scripting (.kts)

Type-safe DSLs

Good IDE support





# Gradle Script Kotlin

Announced in May 2016

Current Release 0.8.0

# Why?

Groovy DSL not designed for performance

Tooling not ideal

Build Scripts that are

- ... expressive

- ... declarative and concise

- ... fluid and readable

- ... highly performant

# Goals

Ease of migration

Preserve existing Gradle plugins

Use idiomatic Kotlin to write Scripts and Plugins

# Challenges

Dynamic concepts in statically-typed language

Only change syntax with good reason

Existing Gradle APIs with Closure and Action<T>

# Extension Functions

```
fun DependencyHandler.compile(  
    dependencyNotation: Any  
): Dependency =  
    add("compile", dependencyNotation)
```

# Lambda with receiver

```
inline fun <T: ModuleDependency> DependencyHandler.add(  
    configuration: String,  
    dependency: T,  
    dependencyConfiguration: T.() -> Unit): T =  
    dependency.apply {  
        dependencyConfiguration()  
        add(configuration, this)  
    }
```

# Delegated Properties

*// task into scope by name*  
***val clean by tasks***

*// task into scope by name and type*  
***val jar: Jar by tasks***



# Accessors

Just-in-time

Ahead-of-time

Ad-hoc

**DEMO**

# IDE Support

IntelliJ IDEA (Kotlin Plugin)

Eclipse (Buildship 2 and Kotlin Plugin)

Netbeans (Kotlin Plugin)

# What's coming next?

Release 1.0

Independent Update mechanism

Compilation speed / performance

Stable DSL (esp. `plugins{}`)

Custom receiver vs Closure / `Action<T>`

Auto detection of `build.gradle.kts`

`.kts` version of `settings.gradle`?

# Further Information

Github

<https://github.com/gradle/gradle-script-kotlin>

Gradle meets Kotlin

<https://blog.gradle.org/kotlin-meets-gradle>

Kotlin Slack

<http://slack.kotlinlang.org/>

Blogpost from Cédric Champeau

<http://melix.github.io/blog/2016/05/gradle-kotlin.html>

**Questions?**

**Thank you!**