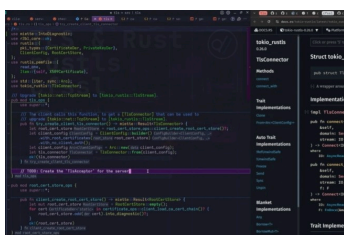


La fondation Rust dévoile son laboratoire d'innovation

Paul Krill, Infoworld (adapté par Dominique Filippone) , publié le 08 Septembre 2025

Invoquant un besoin croissant de gouvernance neutre et communautaire, la fondation Rust a annoncé un programme de soutien à des projets innovants avec son programme Innovation Lab. La bibliothèque de sécurité de la couche de transport Rustls est le premier à en bénéficier.



Rustls, une bibliothèque TLS flexible, performante et sécurisée pour la mémoire, est le premier projet soutenu par la fondation Rust dans le cadre de son programme Innovation Lab. (crédit : Rustls)

Soutenant le développement du langage de programmation open source Rust et de ses différents projets depuis sa création en 2021, la fondation Rust [lance sa dernière initiative : Innovation Lab](#). Son but ? Apporter une aide à des projets open source pertinents. Annoncé le 3 septembre, ce parrainage du

Rust Innovation Lab comprend un soutien au niveau gouvernance, juridique, réseautage, marketing et d'administration. La création de cette structure intervient à un moment charnière selon la fondation.

Le premier projet sponsorisé, Rustls, une bibliothèque TLS (Transport Layer Security) flexible, performante et sécurisée pour la mémoire explique l'organisme à but non lucratif. Cet outil répond à une demande croissante

ARTICLE SUIVANT
Plusieurs évolutions dans la version 2.3.0 de Kotlin e la couche

transport et au niveau de la mémoire. Il est utilisé pour établir des connexions sécurisées dans l'écosystème Rust explique la fondation dont ce projet selon elle démontre la capacité du langage à offrir sécurité et performances dans l'un des domaines les plus sensibles de l'infrastructure logicielle moderne.

Un soutien institutionnel fort pour durer

Les projets menés dans cet Innovation Lab devraient gagner en visibilité dans l'écosystème Rust, et attirer davantage de contributeurs, de partenaires et de soutiens financiers. Un support institutionnel fiable permettra à ces projets de rester durables, sécurisés et indépendants des fournisseurs selon la fondation. L'adoption de ce langage de développement s'est accélérée tant dans l'industrie que dans l'open source, et de nombreux projets écrits en Rust sont devenus des éléments essentiels de l'infrastructure logicielle mondiale, a souligné l'organisme. Avec l'intégration croissante de Rust dans tous les domaines, des plateformes cloud aux systèmes embarqués, un besoin croissant s'est fait sentir pour une gouvernance neutre et dirigée par la communauté, assure t'elle.

Article rédigé par

Paul Krill, Infoworld (adapté par Dominique Filippone)

NEWSLETTER LMI

Recevez notre newsletter comme plus de 50000 abonnés

OK

ARTICLE SUIVANT

Plusieurs évolutions dans la version 2.3.0 de Kotlin

Plusieurs évolutions dans la version 2.3.0 de Kotlin

Paul Krill, IDG NS (adapté par Jean Elyan) , publié le 28 Novembre 2025

Parmi les points forts de la version 2.3.0 de Kotlin désormais disponible en version finale, on trouve un vérificateur pour les valeurs de retour inutilisées, une interopérabilité améliorée avec Swift, la prise en charge de Java 25, des améliorations de la résolution contextuelle et des améliorations pour Swift, Wasm et JavaScript.

```
val generatorTask = project.tasks.register("generator") {
    val outputDirectory = project.layout.projectDirectory.dir("src/main/kotlin/generated")
    outputs.dir(outputDirectory)
    doLast {
        outputDirectory.file("generated.kt").asFile.writeText(
            """language=kotlin
            ***
            fun printHello() {
                println("hello")
            }
            ***.trimIndent()
        """
    )
}
kotlin.sourceSets.getByName("main").generatedKotlin.srcDir(generatorTask)
```

Jetbrains a lancé la version 2.3.0 de Kotlin avec quelques évolutions à la clé.
(Crédit JetBrains)

La dernière mise à jour de l'alternative Java de JetBrains se concentre sur la stabilisation des fonctionnalités et dispose d'un vérificateur expérimental pour les valeurs de retour inutilisées. Selon le fournisseur, cet outil avertit

lorsqu'une expression renvoie une valeur autre que Unit ou Nothing et n'est pas transmise à une fonction, vérifiée dans une condition ou utilisée d'une autre manière. « Cette fonctionnalité peut servir à détecter les bogues dans lesquels un appel de fonction produit un résultat significatif, mais où ce résultat est ignoré sans avertissement », a expliqué l'éditeur.

Deux fonctionnalités du langage précédemment en version bêta sont désormais stables dans Kotlin 2.3.0.

C'est le [CARTICLE SUIVANT](#) de la prise en charge des alias
Plusieurs évolutions dans la version 2.3.0 de Kotlin

de types imbriqués et des vérifications d'exhaustivité du flux de données pour les expressions `when`. « Kotlin 2.3.0 améliore également la résolution contextuelle de deux manières », indique par ailleurs JetBrains.

Premièrement, les supertypes scellés et englobants du type courant sont désormais considérés comme faisant partie de la portée contextuelle de la recherche.

Ensuite, dans les cas impliquant des opérateurs de type et des égalités, le compilateur génère à présent un avertissement si l'usage de la résolution contextuelle rend la résolution ambiguë.

Un effort sur Swift, WASM et JavaScript

Pour le langage Swift, la dernière itération de Kotlin via la compilation Kotlin/Native améliore l'interopérabilité du langage avec Swift grâce à l'exportation Swift, en ajoutant la prise en charge des classes `enum` natives et des paramètres de fonction variadiques (qui accepte un nombre variable de paramètres). « Auparavant, les `enums` Kotlin étaient exportées sous forme de classes Swift ordinaires », a rappelé l'éditeur. Dorénavant, le mappage est direct et les développeurs peuvent utiliser des `enums` Swift natives classiques. Le support Java de Kotlin a été mis à jour. Ainsi, le compilateur peut générer des classes contenant du bytecode Java 25, la dernière version de Java standard publiée le 16 septembre. De plus, la dernière version de Kotlin est entièrement compatible avec l'outil d'automatisation de construction Gradle, versions 7.6.3 à 9.0.0. « Les développeurs peuvent utiliser les versions de Gradle jusqu'à la dernière, mais doivent savoir que cela peut

ARTICLE SUIVANT

Plusieurs évolutions dans la version 2.3.0 de Kotlin

certaines nouvelles fonctionnalités peuvent ne pas fonctionner », a précisé JetBrains.

Pour WebAssembly, la mise à jour active par défaut la propriété `KClass.qualifiedName` sur les cibles Kotlin/Wasm. Cela signifie que les noms complets FQN (Fully Qualified Names) sont disponibles au moment de l'exécution sans configuration supplémentaire. Selon JetBrains, cette modification n'augmente pas la taille du binaire Wasm compilé grâce aux optimisations du compilateur. De plus, afin d'améliorer la compatibilité avec les environnements d'exécution WebAssembly modernes, une nouvelle proposition de gestion des exceptions WebAssembly est activée par défaut pour la cible `wasmWasi`, ajoutant ainsi la gestion des exceptions. Pour Kotlin/JS, qui transcompile le code Kotlin en JavaScript, Kotlin 2.3.0-RC permet d'exporter directement les fonctions de suspension vers JavaScript via l'annotation `@JsExport`. Il s'agit d'une fonctionnalité expérimentale. Selon JetBrains, l'activation de l'exportation des fonctions de suspension supprime le besoin de code standard et améliore l'interopérabilité entre Kotlin/JS et JavaScript/TypeScript (JS/TS). Les fonctions asynchrones de Kotlin peuvent désormais être appelées directement depuis JS/TS sans code supplémentaire.

La version RC de Kotlin 2.3.0 a été présentée le 18 novembre. Kotlin 2.3.0 devrait être disponible en version stable le mois prochain ou en janvier 2026. Les

ARTICLE SUIVANT

Plusieurs évolutions dans la version 2.3.0 de Kotlin

développeurs peuvent accéder à Kotlin 2.3.0-RC sur GitHub.

Article rédigé par

Paul Krill, IDG NS (adapté par Jean Elyan)

NEWSLETTER LMI

Recevez notre newsletter comme plus de 50000 abonnés

OK

ARTICLE SUIVANT

Plusieurs évolutions dans la version 2.3.0 de Kotlin