

DIST★GE

Modern Staged Dependency Injection for Scala

Typelevel Modular Code
with
Context Minimization
through
Garbage Collection

Septimal Mind Ltd
team@7mind.io

The motivation behind DI

- ▶ Systems we work with may be represented as graphs of interlinked components where nodes are components (usually instances) and edges are references,
- ▶ The graph

“DI doesn’t compose with FP”



DI implementations are broken

So we may build better one:



Staged approach



Compile-Time and Runtime DI



Extension: Configuration Support



Pattern: Plan Completion



The Principle Behind: PPER



Garbage Collector



Context Minimization



Context Minimization for Tests



Context Minimization for Deployment



Type Tags for Partially Applied Higher-Kinded Types (Type Lambdas)



Typeclass instance injection (Implicit Injection)



Code example: IO Injection



Code example: Tagless Final Style



Dynamic Plugins



Tags



Plan Introspection



Trait Completion

- ▶ Runtime and Compile-time.

Factory Methods (Assisted Injection)

- ▶ Useful for Akka, lot more convenient than Guice,
- ▶ Runtime and Compile-time.

Status and things to do

Distage is:

- ▶ ready to use,
- ▶ in real production,
- ▶ all Runtime APIs are available,
- ▶ Compile-time verification, trait completion, assisted injections and lambda injections are available.

Our plans:

- ▶ Refactor Roles API,
- ▶ Support running Producer within a monad (IO),
- ▶ Support Scala.js,
- ▶ Support optional isolated classloaders (in foreseeable future),
- ▶ Publish compile-time Producer,
- ▶ Check our GitHub: <https://github.com/pshirshov/izumi-r2>.

DIStage is just a part of our stack

We have a vision backed by our tools:

- ▶ Idealingua: transport and codec agnostic gRPC alternative with rich modeling language,
- ▶ LogStage: zero-cost logging framework,
- ▶ *Fusional Programming and Design* guidelines. We love both FP and OOP,
- ▶ *Continuous Delivery* guidelines for Role-based process,
- ▶ *Percept-Plan-Execute* Generative Programming approach, abstract machine and computational model. Addresses Project Planning (see Operations Research). Examples: orchestration, build systems.

Altogether these things already allowed us to significantly reduce development costs and delivery time for our client.

More slides to follow.

Teaser: LogStage

Teaser: Idealingua

Thank you for your attention

<https://izumi.7mind.io/>

We're looking for clients, contributors, adopters and colleagues ;)

About the author:

- ▶ coding for 18 years, 10 years of hands-on commercial engineering experience,
- ▶ has been leading a cluster orchestration team in Yandex, “the Russian Google”,
- ▶ implemented “*Interstellar Spaceship*” – an orchestration solution to manage 50K+ physical machines across 6 datacenters,
- ▶ Owns an Irish R&D company, <https://7mind.io>,
- ▶ Contacts: team@7mind.io,
- ▶ Github: <https://github.com/pshirshov>