

Migrating Spring Boot apps

From Annotation-based Config
to Functional with Kotlin

Me, myself and I

- **Kotlin fan**
- **Spring Boot lover**





European alternative to the “big” cloud-computing players

- **Privacy-minded**
- **Great support**



@nicolas_frankel

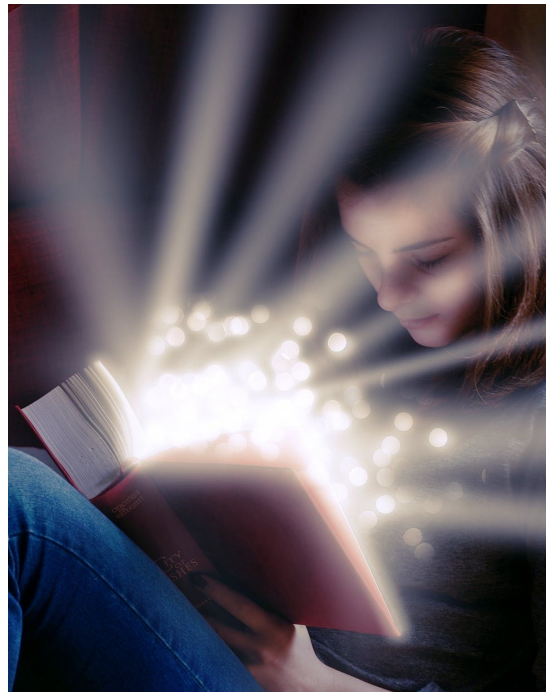
Spring Boot

- **Open Source**
- **Pivotal**
- **Spring Framework**
- **Convention over configuration**



The issue

- **Magic**
 - Via annotations
- **The solution?**
 - Functional configuration





Time for **DEMO**

Going further

- **Spring Fu**
 - Ko-Fu
 - ~~○ Ja-Fu~~
- **Experimental!**



Kofu configuration for Spring Boot

Kofu configuration is currently developed in Spring Fu, an incubator intended to ultimately contribute features related to Kotlin and functional configuration to Spring Framework, Boot and Data.

Kofu configuration for Spring Boot

- **Explicit configuration via a Kotlin DSL**
- **Based on Spring Boot infrastructure used in a functional way**
- **No feature enabled based on classpath detection**
- **Both declarative and programmatic**
- **Faster startup and lower memory consumption**
- **Minimal reflection & annotations usage**
- **Pure lambdas, no CGLIB proxy**

Takeaways

- **Controller to routes**
 - Logic in handlers
- **Kotlin Beans DSL**
- **Kotlin Routes DSL**
- **No more necessary to be reactive**





Thanks!

- <https://blog.frankel.ch/>
- @nicolas_frankel
- <https://git.io/fN3Qv>

