

Extending kotlinx.serialization functionality with Arrow Meta

Karin-Aleksandra Monoid

Joker, 2021



About

- Senior Software Engineer at CircleCI
- Background in Kotlin & Scala (both front and backend)
- Living in Berlin with two cats
- GDE in Kotlin

About

- Senior Software Engineer at CircleCI
- Background in Kotlin & Scala (both front and backend)
- Living in Berlin with two cats
- GDE in Kotlin



Agenda

Why?

What is Arrow Meta?

Sample project

Showtime!

Project Stack

- Just backend with Ktor + Arrow
- Jackson was kind of "default"
- Smooth but not Kotlin-first

kotlinx.serialization – pros 🚀

- Kotlin-first
- No boilerplate for sealed classes

kotlinx.serialization – cons

- Not a lot of serializers out of the box
- No global naming strategy

Global Naming Strategy

UPPER_SNAKE_CASE -> camelCase

USER_NAME -> userName

Jackson: Object Mapper

```
val objectMapperUpperCamelCase =  
    jacksonObjectMapper().apply {  
        propertyNamingStrategy =  
        PropertyNamingStrategies.UPPER_SNAKE_CASE  
    }
```

Issue #33

Fields name policy converters #33

 Open

gildor opened this issue on 17 Nov 2017 · 21 comments

<https://github.com/Kotlin/kotlinx.serialization/issues/33>

Global Naming Strategy

```
@SerialName("USER_NAME")
```

```
val userName: String
```

What is Arrow Meta?



What is Arrow Meta?

**“Functional companion
to Kotlin's Compiler”**



Image source: <https://arrow-kt.io/>



GIF source: <https://github.com/arrow-kt/arrow-meta>

Why Arrow Meta?

- Already using other part of Arrow for FP



CORE

Functional companion to
Kotlin's Standard Library



FX

Functional Effects Framework
companion to KotlinX
Coroutines.



OPTICS

Deep access and
transformations over immutable
data



META

Metaprogramming library for
Kotlin compiler plugins

Why Arrow Meta?

- More functionality than KSP/KAPT
- Easier than kotlin

Sample Project



Link to repo

<https://bit.ly/3C2IHGp>



Setting: a Zoo

- various animals (actually, only cats and dogs)
- various sources (= 3rd party services)
- various standards of documentation (in JSON format with different naming policies)

Live coding time



Thoughts

- It works
- Not production ready
- Requires more knowledge than expected
- KSP is preferable if it fits

Resources

- <https://meta.arrow-kt.io/>
- <https://github.com/arrow-kt/arrow-meta-examples>
- <https://paranoidmonoid.github.io/articles/Kotlin%20and%20friends/Naming%20strategy%20plugin>