Beam ❤️ Kotlin: full pipeline with Midgard

Mazlum TOSUN
Head of Data and Cloud Group Bees
https://twitter.com/MazlumTosun3
About me

Mazlum TOSUN

Head of Data and Cloud GroupBees

❖ Google Cloud Evangelist, Data Architect, functional programming, Devops, Serverless...
❖ Fan

https://www.youtube.com/@GCPLearning-ce9bg
https://github.com/tosun-si
https://twitter.com/MazlumTosun3
https://www.linkedin.com/in/mazlum-tosun-900b1812/
https://medium.com/@mazlum.tosun
https://stackoverflow.com/users/9261558/mazlum-tosun
Some Idioms

Create object

```kotlin
data class Customer(val name: String, val email: String)
```

Default values for function parameters

```kotlin
fun foo(a: Int = 0, b: String = "") { ... }
```
Filter a list

```kotlin
val positives = list.filter { it > 0 }
```

Check presence element collection

```kotlin
if ("john@example.com" in emailsList) { ... }

if ("jane@example.com" !in emailsList) { ... }
```
String interpolation

```kotlin
println("Name $name")
```

Instance checks

```kotlin
when (x) {
    is Foo -> ...
    is Bar -> ...
    else -> ...
}
```
Why Kotlin

Read-only map and list

```kotlin
val list = listOf("a", "b", "c")

val map = mapOf("a" to 1, "b" to 2, "c" to 3)
```

Extensions functions

```kotlin
fun String.spaceToCamelCase() { ... }

"Convert this to camelcase".spaceToCamelCase()
```
**Singleton**

```kotlin
object Resource {
    val name = "Name"
}
```

**If-not-null shorthand**

```kotlin
val files = File("Test").listFiles()

println(files?.size) // size is printed if files is not null
```
If-not-null-else shorthand

```kotlin
val files = File("Test").listFiles()

println(files?.size ?: "empty") // if files is null, this prints "empty"

// To calculate the fallback value in a code block, use `run`
val fileSize = files?.size ?: run {
    return someSize
}
println(fileSize)
```
Why Kotlin

Execute a statement if null

```kotlin
val values = ...
val email = values["email"] ?: throw IllegalStateException("Email is missing")
```

Generic function that requires the generic type information

```kotlin
inline fun <reified T: Any> Gson.fromJson(json: JsonElement): T = this.fromJson(json,
T::class.java)
```
Check this link to have more examples

https://kotlinlang.org/docs/idioms.html
Why Kotlin instead of Scala?
Midgard

- Library for Beam Kotlin
- Proposes some extensions on PCollection
- Oriented to Functional Programming style
- Proposes map, flatMap and filter operator
- Executes operators and access to DoFn lifecycle and ProcessContext if needed
Example of usual Beam pipeline with map, flatMap and filter operations:

```java
val resultPlayers: PCollection<Player> = pipeline
    .apply("Create", Create.of(listOf(psgTeam, realTeam)))
    .apply("To Team with Slogan V2",
        MapElements
            .into(Descriptor.of(Team::class.java))
            .via(SerializableFunction { it.copy(slogan = "${it.slogan} VERSION 2") })
    )
    .apply("To Players",
        FlatMapElements
            .into(Descriptor.of(Player::class.java))
            .via(SerializableFunction { it.players })
    )
    .apply("Filter age > 25", Filter.by(SerializableFunction { it.age > 25 }))
```
The same pipeline with Midgard library:

```scala
import fr.groupbees.midgard.*

val resultPlayersMidgard: PCollection<Player> = pipeline
  .apply("Create", Create.of(listOf(psgTeam, realTeam)))
  .map("To Team with Slogan V2") { it.copy(slogan = "${it.slogan} VERSION 2") }
  .flatMap("To Players") { it.players }
  .filter("Filter age > 25") { it.age > 25 }
```
Midgard allows to propose map and flatMap operators and extensions while interacting with this lifecycle

```kotlin
val resultTeamMidgardMapLifeCycle: PCollection<Team> = pipeline
    .apply("Create", Create.of(listOf(psgTeam, realTeam)))
    .mapFn(
        name = "To Team with Slogan V2",
        transform = { it.copy(slogan = "${it.slogan} VERSION 2") },
        setupAction = { println("Setup Action") },
        startBundleAction = { println("Start Bundle Action") },
        finishBundleAction = { println("Finish Bundle Action") },
        teardownAction = { println("Teardown Action") }
    )
```
Lifecycle and access DoFn ProcessContext while applying the current transformation

```scala
// Simulate a side input for the slogan suffix.
val slogansSideInput: PCollectionView<String> = pipeline
  .apply("Read slogans", Create.of("VERSION 2"))
  .apply("Create as collection view", View.asSingleton)

val resultTeamMidgardMapContextLifeCycle: PCollection<Team> = pipeline
  .apply("Create", Create.of(listOf( PSGTeam, realTeam )))
  .mapFnWithContext(
    name = "To Team with Slogan V2",
    transform = { context -> toTeamWithSloganSuffixFromSideInput(slogansSideInput, context) },
    setupAction = { println("Setup Action") },
    sideInputs = listOf(slogansSideInput),
    startBundleAction = { println("Start Bundle Action") },
    finishBundleAction = { println("Finish Bundle Action") },
    teardownAction = { println("Teardown Action") }
  )
```
Lifecycle and access DoFn ProcessContext : side input

```kotlin
private fun toTeamWithSloganSuffixFromSideInput(
    sideInput: PCollectionView<String>,
    context: DoFn<Team, Team>.ProcessContext
): Team {
    val currentTeam: Team = context.element()
    val sloganSuffixSideInput: String = context.sideInput(sideInput)

    return currentTeam.copy(slogan = "${currentTeam.slogan} $sloganSuffixSideInput")
}
```
Midgard: code demo with real world pipeline

Qatar Fifa World Cup Teams Players Stats pipeline with Beam Kotlin and Midgard

Source Data

Team Players Stats Raw Data

Google Cloud Project

1. Cloud Storage
2. Dataflow
3. BigQuery

Midgard
Kotlin

Team Players Domain Data (Aggregation per team)

Dataviz

Looker Studio

topScorers
bestPassers
bestDribblers
goalKeeper
playersMostAppearances
playersMostDuelsWon
playersMostInterceptions
playersMostSuccessfulTackles
...
Midgard: possible evolutions in the future

- Add extensions for existing IOs:
  - TextIO
  - BigQueryIO
  - ......

- Add extensions for built in transform:
  - WithKey
  - GroupByKey
  - ......
Links to example projects

https://github.com/tosun-si/world-cup-qatar-team-stats-java

https://github.com/tosun-si/world-cup-qatar-team-stats-kotlin-midgard
Link of Midgard project

https://github.com/tosun-si/midgard

Feel free to contribute to the project, give me feedback, try it and support us with a Github star 🌟
Thank you :)