

Frameless: An Introduction

Sabrina Konrad-lee

January 14, 2026

Why am I talking about this?

- ▶ Getting the most out of your tools
- ▶ Sometimes a perfect tool exists, you just don't know about it!
- ▶ We get trapped in the familiar, unrealized business value
- ▶ Takeaways

Spark

- ▶ Spark what is it? Driver, Cluster Manager, Executors, RDDs, Datasets,

Choosing Spark

Why Spark?

- ▶ Strongly typed scala
- ▶ Good general functionality
- ▶ Library support

Why not Spark?

- ▶ Not as fast or specialized
- ▶ SQL easier in other frameworks
- ▶ Making case classes repeatedly

Why we like strongly typed

- ▶ Compile time error catching
- ▶ Catching errors earlier, safer
- ▶ Higher confidence,
- ▶ Structured data, handling nulls

Frameless - What is it

- ▶ Doesn't spark already have a typed dataset?
- ▶ Underlying spark dataset
- ▶ "spark.implicits._" -runtime reflection vs X
- ▶ Typed Columns!
- ▶ Physical Plan improvements

Problems with Spark - Column

```
//case class DataExample(i: Int, j: String, k: Double)
ds.select("ii").show()
ds.select("i").show()
```

Frameless - Typed Column

```
object Exists {  
    def apply[T, V](column: Witness)(implicit e: Exists[T, column.T, V]): Exists[T, column.T, V] = e  
  
    implicit def deriveRecord[T, H <: HList, K, V]  
    (implicit  
        i0: LabelledGeneric.Aux[T, H],  
        i1: Selector.Aux[H, K, V]  
    ): Exists[T, K, V] = new Exists[T, K, V] {}  
}
```

Done using Shapeless HLists

Frameless - Typed Column

```
//case class DataExample(i: Int, j: String, k: Double)
ds.filter($"i" === "foo").show()
```

Figure 1: Spark

```
ds.filter(ds('i) === "18")
```

Figure 2: Frameless

Frameless - Combining Cats and Frameless

- ▶ cats-effect and cats MTL
- ▶ everything is lazy
- ▶ spark with for-comprehension

```
trait SparkDelayInstances {  * Itamar Ravid
    implicit def framelessCatsSparkDelayForSync[F[_]](implicit S: Sync[F]): SparkDelay[F] = new SparkDelay[F] {  * Itamar Ravid
        def delay[A](a: => A)(implicit spark: SparkSession): F[A] = S.delay(a)
    }
}
```

Example - Frameless and Cats

```
val result: Action[Seq[(Int, String)], Long] = for {
    sample <- typedDs.take[Action](1)
    count <- typedDs.count[Action]()
} yield (sample, count)
```

Questions