





**MODÉLER UNE POTENTIELLE**

EWANA

MAGNUM2-QUALITÉ D'SI-THC MAS HAE SLÉ & DUE N T INBURG

N'existe pas dans la lib standard

Peut être encodé avec les génériques (vous avez vu comment) ... c'est pas compliqué mais il manque quelques fonctions pour être réellement utilisable (pure, map, flatmap, fold)

Si vous voulez faire du Java « PRO » vous pouvez utiliser VAVR

[https://www.vavr.io/vavr-docs/#\\_either](https://www.vavr.io/vavr-docs/#_either) ...

Ou passez à Scala ou Kotlin + Arrow-Kt

**Quabandonnez la JVM  pour OCaml, Rust, F# ou Haskell**

# MODÉLISER UNE ERREUR POTENTIELLE

## EN JAVA

N'existe pas dans la lib standard

Peut être encodé avec les génériques (vous avez vu comment) ... c'est pas compliqué mais il manque quelques fonctions pour être réellement utilisable (pure, map, flatmap, fold)

Si vous voulez faire du Java « PRO » vous pouvez utiliser VAVR  
[https://www.vavr.io/vavr-docs/#\\_either](https://www.vavr.io/vavr-docs/#_either) ...

Ou passez à Scala ou Kotlin + Arrow-Kt

Ou abandonnez la JVM 🐉 pour OCaml, Rust, F# ou Haskell



# TAKE AWAY

## LES RESULT / EITHER

A utiliser pour modéliser les cas d'erreur

Sécurisant

Facile à manipuler

Parfois encodé avec un seul paramètre, pour homogénéiser les erreurs, s'appelle souvent Try dans ce cas.

```
interface MyDomainErrors extends Error { };  
type Try<A> = Either<MyDomainErrors, A>;
```

