

# MORE

## ACCUMULATEURS D'ERREURS

On souhaite parfois remonter l'ensemble des erreurs rencontrées (formulaire, json, csv, compilation, ...) : nous avons besoin d'accumulateurs d'erreurs !

**Problème** Result.t va contenir la première erreur qui sera propagée par bind ou map

On aimerait quelque chose qui ressemble à

```
type ('ok, 'error) result =  
  | Ok of 'ok  
  | Error of 'error list
```

*(ce type n'existe pas dans Stdlib c'est une illustration)*

**Avec OCaml Base (Stdlib alternative), il existe une fonction** combine\_errors

val combine\_errors : ('ok, 'err) [t](#) list -> ('ok list, 'err list) [t](#)

**... mais cela devient vite fastidieux à manipuler**

# VALIDATION

ACCUMULATEUR D'ERREUR POUR VALIDER LES DATA STRUCTURES

- En Haskell, il existe le module Control.Monad.Validate

Dans les autres langages, il faut existe souvent des lib complémentaires :

- En OCaml, la lib Base fournit un module validate
- En Scala, la lib Cats fournit un type de donnée Validated
- En Kotlin, la lib Arrow-kt fournit un type de donnée Validated