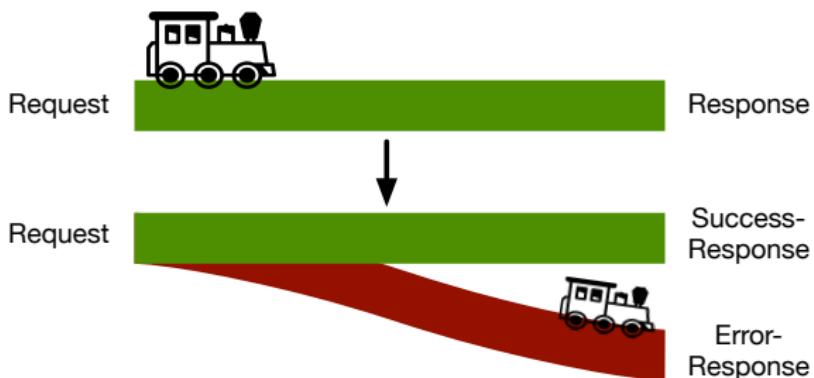


RAILWAY-ORIENTED-PROGRAMMING IN CLOJURE

Funktionales Error-Handling

EFP, SS2017



Jan-Philipp Willem

Fakultät für Informatik
Hochschule Mannheim

PROJECT-REPOSITORY

<https://github.com/jwillem/rop-clojure>

GLIEDERUNG

1. Who?
2. „Happy-Path“
3. Error by design
4. „Railway-Oriented-Programming“
5. Ausblick

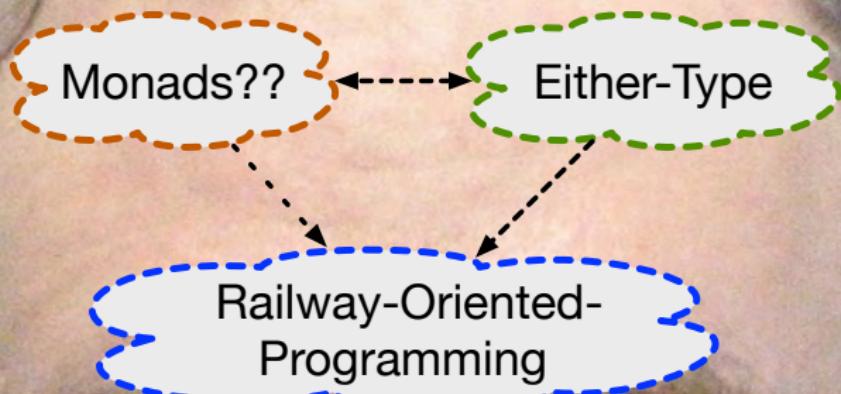
SCOTT WLASCHIN



<https://fsharpforfunandprofit.com/>

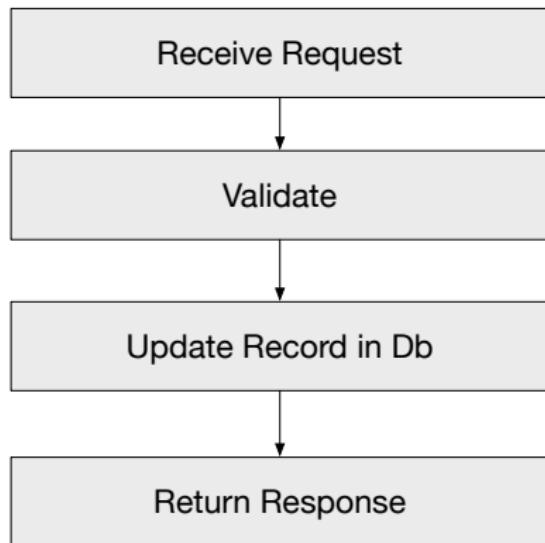
„LANGES LANGES ZITAT, LOREM
IPSUM, WEIL DESWEGEN UND SO
WEITER. FOOBAR IS BARBAZ AND
SO ON.“ (SCOTT WLASCHIN, CONFERENCE 2015)[1]

CONCEPT



„HAPPY-PATH“

USECASE: UPDATE FIRSTNAME + EMAIL OF USER (1/2)



```
{  
  "type": "user",  
  "firstname": "Max",  
  "email":  
    "max@muster.tld"  
}
```

USER-VALIDATION MIT NODE (JS)

2017-05-12

Railway-Oriented-Programming in Clojure

└ „Happy-Path“

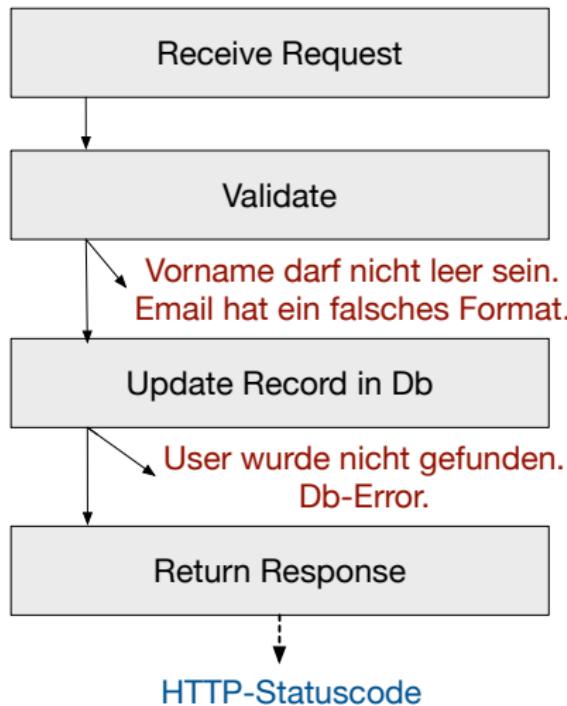
└ User-Validation mit Node (JS)

- node-coding-style

ERROR BY DESIGN

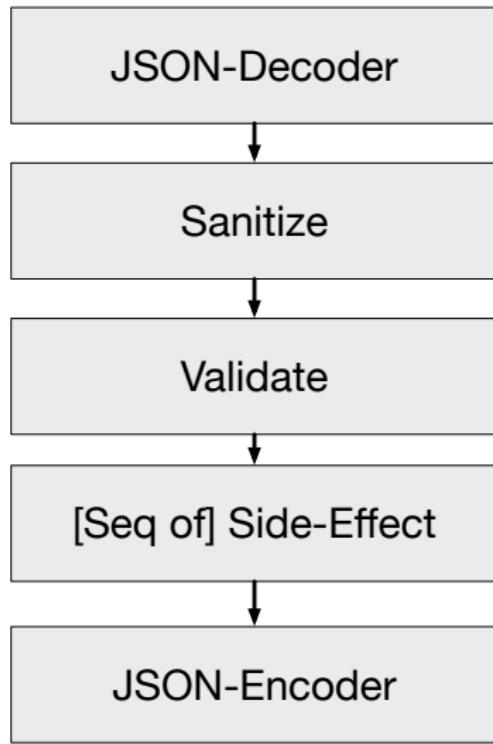
USECASE: UPDATE FIRSTNAME + EMAIL OF USER (2/2)

Und sinnvolle Fehler zurückgeben.



```
{  
  "type": "user",  
  "firstname": " ",  
  "email":  
    "max@@@muster.tld"  
}
```

DATA-FLOW MIT CLOJURE



„RAILWAY-ORIENTED- PROGRAMMING“

ÄPFEL & BIRNEN

GRUNDLEGENDE SCHIENENTEILE

→ switch-function

GRUNDLEGENDE SCHIENENTEILE

- switch-function
- bind

GRUNDLEGENDE SCHIENENTEILE

- switch-function
- bind
- pipe-bind

GRUNDLEGENDE SCHIENENTEILE

- switch-function
- bind
- pipe-bind
- compose

GRUNDLEGENDE SCHIENENTEILE

- switch-function
- bind
- pipe-bind
- compose
- clojure seq-abstraction

GRUNDLEGENDE SCHIENENTEILE

- switch-function
- bind
- pipe-bind
- compose
- clojure seq-abstraction
- rest-parameters, apply

USER-VALIDATION MIT ROP

AUSBLICK

WEITERE SCHIENENTEILE

- try/catch
- log
- ..

FAZIT

FIN

Danke für die Aufmerksamkeit. – Fragen?

REFERENCES

1 <https://vimeo.com/113707214>

2 https://guide.elm-lang.org/error_handling