Tutorial: Microservices in Haskell

Alexander Abushkevich

March 2015

REST and DB access

alexanderaa.github.io/haskell-microservices

Agenda

- Microservices - definition and introductory notes
- Quick overview - request/response cycle and associated type conversions
- Focus on a variety of libraries, which help to (de)serialize JSON/XML/...

Agenda

- Microservices - definition and introductory notes
- Quick overview - request/response cycle and associated type conversions
- Focus on a variety of libraries, which help to (de)serialize JSON/XML/...

- Focus on a variety of libraries, which help to (de)serialize DB data
- REST in Haskell
- Putting all pieces together - compile and run the resulting microservice

Microservices - definition

```
1       val range f t =
2           let
3               fun range' acc f t =
4                   if (f >= t)
5                       then acc
6                       else (range' (f :: acc) (f+1) t)
7           in
8               range' [] f t
9           end
```

Yesod

Applicative functors

```
Prelude> import Control.Applicative
Prelude Control.Applicative> :t (<$>)
(<$>) :: Functor f => (a -> b) -> f a -> f b
```

Aeson

```haskell
instance AE.FromJSON Message where
    parseJSON (AE.Object v) =
        Message <$> (v AE..: "id"      )
                <*> (v AE..: "message" )
                <*> (v AE..: "status"  )
                <*> (v AE..: "dated"   )
```

PostgreSQL-simple

Questions?