Haskell - A Webserver
Functional Programming

Jens Egholm Pedersen and Anders Kalhauge

Spring 2018

Outline

Scotty
- Setup
- JSON
- GET
- POST
- Further information

Scotty

- Setup
- JSON
- GET
- POST
- Further information

Scotty Stack Setup

In the `package.yaml` file:

```
...
dependencies:
- base >= 4.7 && < 5
- containers
- scotty
- text
- bytestring
- aeson
```

Scotty Hello World!

In app/Main.hs

```haskell
{-# LANGUAGE OverloadedStrings #-}
module Main where

import Web.Scotty

main :: IO ()
main = scotty 3000 $ do
  get "/hello/:name" $ do
    name <- param "name"
    html $ mconcat [ "<h1>Hello␣"
                   , name
                   , "␣from␣Scotty!</h1>"
                   ]
```

Scotty build and run

First build the project, this takes time the first time

```
$ stack build
```

Then execute the server:

```
$ stack exec <project-name>-exe
```

Scotty JSON

```haskell
{-# LANGUAGE OverloadedStrings #-}
{-# LANGUAGE DeriveGeneric #-}

import GHC.Generics

import Web.Scotty
import Data.Aeson (FromJSON, ToJSON)

data Person = Person
  { name :: String
  , email :: String
  , age :: Integer
  } deriving (Show, Generic)
instance ToJSON Person
instance FromJSON Person
```

Scotty get

```haskell
get "/person" $ do
  json $ Person "Kurt" "kurt@mail.dk" 27
```

Scotty POST

```haskell
post "/person" $ do
  person <- jsonData :: ActionM Person
  json person
```

Scotty Types

```
ghci> t: scotty
scotty :: Port -> ScottyM () -> IO ()
```

```
ghci> :t get
get :: RoutePattern -> ActionM () -> ScottyM ()
```

```
ghci> t: json
json :: ToJSON a => a -> ActionM ()
```

Further Information

- 24 Days of Hackage: scotty
  `https://ocharles.org.uk/blog/posts/2013-12-05-24-days-of-hackage-scotty.html`
- School of Haskell:
  `https://www.schoolofhaskell.com/`
- Practical Haskell - Building a JSON API
  `http://seanhess.github.io/2015/08/19/practical-haskell-json-api.html`