The Haskell WebSockets library
Dutch HUG Day

Jasper Van der Jeugt

April 20, 2012
Hello!

My name is Jasper
Student at UGent
I write Haskell
GhentFPG
@jaspervdj
jaspervdj.be
Overview

About WebSockets
Protocol versions: typeclass fun
Example: a webchat
WebSockets

**WebSocket** is a web technology providing bi-directional, full-duplex communication channels, usually between a browser and an HTTP server.
WebSockets

Starts as an HTTP request, usually over port 80

GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
...
WebSockets

Server upgrades the connection

HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade

...
WebSockets

Possible uses:
Chat services
Multiplayer games
Real-time notifications
...
WebSockets

```javascript
var ws = new WebSocket(uri);

ws.onmessage = function(event) {
    alert(event.data);
};

ws.onmessage = function() {
    ws.send('Hello, server. ');
};
```
About WebSockets

Supported browsers:

IE10 will also support the protocol
Overview

About WebSockets

Protocol versions: typeclass fun

Example: a webchat
Protocol versions: typeclass fun

Two main versions in use: hybi-00 and hybi-10

(other versions are similar)
About WebSockets

Important detail:
Protocol allows for version negotiation between client and server
Protocol versions: typeclass fun

Features:

hybi10  hybi00
Protocol versions: typeclass fun

Implementation:

hybi10

hybi00
Protocol versions: typeclass fun

Code on the following slides is simplified for conciseness and clarity...
Protocol versions: typeclass fun

sendTextData
  :: TextProtocol p
  ⇒ Text
  ⇒ WebSockets p ()
Protocol versions: typeclass fun

```
sendBinaryData :: BinaryProtocol p => ByteString => WebSockets p()
```
Protocol versions: typeclass fun

— Compatible with hybi–00
— and hybi–10

app :: TextProtocol p
  => WebSockets p ()

main :: IO ()
main = serve
  — Use hybi–00 or above
  (app :: WebSockets Hybi00 ())
Protocol versions: typeclass fun

```
class Protocol p =>
    TextProtocol p
class Protocol p =>
    BinaryProtocol p

instance TextProtocol Hybi00
instance TextProtocol Hybi10
instance BinaryProtocol Hybi10
```
Protocol versions: typeclass fun

```
class Protocol p where
  version :: p -> String
  handshake :: Iteratee ...
  ...

  implementations :: [p]
```
Protocol versions: typeclass fun

\{-# LANGUAGE \\
  ExistentialQuantification #-\} \\

— hybi-00 or above...

data Hybi00 = forall p.
  Protocol p ⇒ Hybi00 p
Protocol versions: typeclass fun

instance Protocol Hybi00 where
  version  (Hybi00 p) =
        version p
  handshake (Hybi00 p) =
        handshake p

  implementations =
    [ Hybi00 Hybi00,
      , Hybi00 Hybi10
    ]
Protocol versions: typeclass fun

```haskell
data Hybi00_ = Hybi00_
instance Protocol Hybi00_ where
  -- Actual implementation ...
  implementations = undefined

data Hybi10_ = Hybi10_
instance Protocol Hybi10_ where
  -- Actual implementation ...
  implementations = undefined
```
instance Protocol Hybi10 where
    version (Hybi10 p) =
        version p
    handshake (Hybi10 p) =
        handshake p

... implementations =
    [ Hybi10 Hybi10 ]
Overview

About WebSockets
Protocol versions: typeclass fun
Example: a webchat
Example: a webchat

Implementing a simple webchat

Live demo:

jaspervdj.be/websockets-example
Thanks for listening!

Questions?