PHOENIX BEYOND COWBOY

Mat Trudel // mat@geeky.net // github.com/mtrudel/talks

LET'S TALK ABOUT **PHOENIX**

1.7 IS A
BIG ONE

VERIFIED ROUTES

QoL TWEAKS

FIRST-CLASS WEBSERVER SUPPORT

new visual project identity

Components

BETTER AUTH

TAILWIND CSS

SNAZZY GENERATORS

Unified HEEx Templates

FIRST-CLASS WEBSERVER SUPPORT

- Support Webservers other than Cowboy
- Phoenix shouldn't care what Webserver it is running in
- Cowboy becomes 'just another' Webserver
- Bandit now fully supports Phoenix
- Requires a generic interface for HTTP
  - This is what Plug does today
- Also requires a generic interface for WebSockets
  - Prior to Phoenix 1.7, this didn't exist

INTRODUCING

WEBSOCK

github.com/phoenixframework/websock
github.com/phoenixframework/websock_adapters

INTRODUCING WEBSOCK

- A generic abstraction over WebSockets
- Provides a server-agnostic way for applications like Phoenix to use WebSockets
- *"Plug for WebSockets"*, basically
- Completely distinct from Plug, but the two work in concert to facilitate *protocol upgrades*
  - Protocol upgrades are choreographed by the **WebSockAdapters** library

WHAT IS A WEBSOCKET?

WHAT IS A WEBSOCKET?

```
> GET /path HTTP/1.1
> Connection: Upgrade
> Upgrade: WebSocket
> Sec-WebSocket-*: …



< HTTP/1.1 101 Switching Protocols
< Connection: Upgrade
< Upgrade: WebSocket
< Sec-WebSocket-*: …



The connection is now a WebSocket,
with frames sent in both directions
```

THIS LOOKS A LOT LIKE PLUG!

PROTOCOL UPGRADES

```
> GET /path HTTP/1.1
> Connection: Upgrade
> Upgrade: WebSocket
> Sec-WebSocket-*: …



< HTTP/1.1 101 Switching Protocols
< Connection: Upgrade
< Upgrade: WebSocket
< Sec-WebSocket-*: …



The connection is now a WebSocket,
with frames sent in both directions
```

regular Plug call

```
def call(conn, _opts) do
  # WebSockAdapter wraps Plug.Cowboy.Http.Adapter
  WebSockAdapter.upgrade(conn, MyWebSock, state, [])
end
```

underlying Webserver
performs upgrade

Webserver talks to **MyWebSock**
via the WebSock behaviour

WEBSOCK BEHAVIOUR

- **init(state)**
  - Called whenever a WebSocket connection is established
- **handle_in(msg, state)**
  - Called whenever the client sends a WebSocket frame to the server
- also: **handle_info/2**, **handle_control/2**, **terminate/2**

DEMO #1

PUTTING IT ALL TOGETHER

- **WebSock** defines an API for WebSockets
- **Plug** provides some (purposely) dumb upgrade plumbing
- **WebSockAdapter** ties it all together
- **Bandit** & **Cowboy*** serve Plug and Websock

* Cowboy support actually implemented in WebSockAdapter

WHAT DOES THIS MEAN FOR PHOENIX?

PHOENIX 1.7'S WEBSOCKET SUPPORT IS BUILT **ENTIRELY** ON WEBSOCK

ANY

PLUG & WEBSOCK SERVER
CAN HOST PHOENIX

WEBSOCK_ADAPTERS PROVIDES SUPPORT FOR COWBOY

BANDIT IS
PLUG & WEBSOCK
NATIVE

END RESULT:

A FULLY **API-BASED** HTTP & WEBSOCKET STACK

WHAT DIFFERENCE DOES THIS MAKE FOR **ME**?

WEBSOCKET UPGRADES ARE

100%

PLUG-MEDIATED

WEBSOCKET HANDLERS CAN LIVE
ANYWHERE
IN PHOENIX'S ROUTER

YOU CAN
WRITE YOUR OWN
WEBSOCKET HANDLERS

(GREAT FOR LEGACY FRONTENDS)

DEMO #2

ADDONS LIKE
OBAN & LIVE DASHBOARD
CAN BE TOTALLY **SELF-CONTAINED**

COWBOY IS NOW JUST **'ONE AMONG MANY'** POSSIBLE WEB SERVERS

(THE NEXT ONE IS EASY!)

AN UPDATE ON

BANDIT

github.com/mtrudel/bandit

Up to 4x Faster HTTP/1.1

iolists everywhere

SUPER READABLE CODE

100% Elixir

Ergonomic API

COMPLETE PHOENIX SUPPORT

FULLY WIRED FOR TELEMETRY

½ THE PROCESS COUNT

3RD PARTY CONFORMANCE

OCT 2021
Bandit's first 'public' release 0.4.x

MAY 2022
Announced HTTP(S)-only Phoenix Support

NOV 2022
Released WebSock & Bandit 0.5.x

JAN 2023
Finalized performance improvements as 0.6.x

MAR 2023
Refined configuration & telemetry as 0.7.x

APR 2023
...not much left to do

ThousandIsland

Bandit

BANDIT **1.0-PRE** SERIES STARTS *TODAY!*

```
{:bandit, "~> 1.0-pre"}

{:thousand_island, "~> 1.0-pre"}
```

github.com/mtrudel/bandit

BANDIT

FULLY SUPPORTS

PHOENIX 1.7+

(ANYTHING THAT DOESN'T WORK IS A BUG)

SWITCHING TO BANDIT IS A **ONE LINE CHANGE**

(SEE THE BANDIT README FOR DETAILS)

github.com/mtrudel/bandit

SO WHAT ABOUT THE FUTURE?

FUTURE #1:

A FEW DEEPLY OBSCURE RFC ADDITIONS TO BANDIT, ALL ADDITIVE

RFC 8441

FUTURE #2:

SOME CHANGES TO PHOENIX'S PREFERRED API FOR SOCKET DEFINITION

FUTURE #3:

QIUC, HTTP/3 & WEBTRANSPORT

(THIS IS A MASSIVE UNDERTAKING WITH A TON OF DEPENDENCIES)

mat@geeky.net

github.com/mtrudel/bandit

github.com/mtrudel/talks

#bandit

#thousand_island

fin