ELIXIR CORE & ECOSYSTEM UPDATE

Code BEAM SF 2020

MICHAŁ MUSKAŁA

Elixir Core Team alumni

http://michal.muskala.eu/

https://github.com/michalmuskala/

@michalmuskala

AGENDA

- Elixir 1.9
- Elixir 1.10
- Future plans
- Community project updates

ELIXIR 1.9

- June 2019
- Releases
- Configuration changes
- Logger improvements
- Calendar types improvements

ELIXIR 1.9 - RELEASES

- self-contained directory for deployment
- `MIX_ENV=prod mix release`
- `bin/my_app start|start_iex|restart|stop`
- `bin_my_app rpc|remote|eval`
- `bin/my_app daemon|daemon_iex|install`

WHY RELEASES?

- Code preloading
- Configuration and customisation
- Self-contained
- Multiple releases

CONFIGURATION

- `config/config.exs`
- `config/releases.exs`
- `rel/vm.args.eex`
- `rel/env.sh.eex` and `rel/env.bat.eex`

ELIXIR 1.10

- January 2020
- 900+ contributors
- 10 000+ packages on hex.pm
- 1 100 000 000+ downloads

ELIXIR 1.10

- Requires OTP 21+
- OTP logger integration
- Release improvements
- Easier sorting
- `is_struct/1` and `is_map_key/2`

ELIXIR 1.10 - SORTING

```
iex> Enum.sort(["banana", "apple", "pineapple"])
["apple", "banana", "pineapple"]
```

ELIXIR 1.10 - SORTING

```
iex> Enum.sort(["banana", "apple", "pineapple"])
["apple", "banana", “pineapple"]

iex> Enum.sort(["banana", "apple", "pineapple"], &≥/2)
["pineapple", "banana", "apple"]
```

ELIXIR 1.10 - SORTING

```
iex> Enum.sort(["banana", "apple", "pineapple"])
["apple", "banana", “pineapple"]

iex> Enum.sort(["banana", "apple", "pineapple"], &≥/2)
["pineapple", "banana", "apple"]

iex> Enum.sort([~D[2019-12-31], ~D[2020-01-01]])
[~D[2020-01-01], ~D[2019-12-31]]

iex> Enum.sort([~D[2019-12-31], ~D[2020-01-01]],
               &(Date.compare(&1, &2) ≠ :lt))
[~D[2019-12-31], ~D[2020-01-01]]
```

ELIXIR 1.10 - SORTING

```
iex> Enum.sort(["banana", "apple", "pineapple"], :asc)
["apple", "banana", "pineapple"]
iex> Enum.sort(["banana", "apple", "pineapple"], :desc)
["pineapple", "banana", "apple"]

iex> Enum.sort([~D[2019-12-31], ~D[2020-01-01]], Date)
[~D[2019-12-31], ~D[2020-01-01]]
iex> Enum.sort([~D[2019-12-31], ~D[2020-01-01]], {:desc, Date})
[~D[2020-01-01], ~D[2019-12-31]]
```

ELIXIR 1.10 - CONFIGURATION

```
defmodule MyApp.DBClient do
  @db_host Application.get_env(:my_app, :db_host, "db.local")
  def start_link() do
    SomeLib.DBClient.start_link(host: @db_host)
  end
end
```

ELIXIR 1.10 - CONFIGURATION

```
defmodule MyApp.DBClient do
  def start_link() do
    SomeLib.DBClient.start_link(host: db_host())
  end
  defp db_host() do
    Application.get_env(:my_app, :db_host, "db.local")
  end
end
```

ELIXIR 1.10 - CONFIGURATION

```
@db_host Application.compile_env(:my_app, :db_host, "db.local")
```

ELIXIR 1.10 - COMPILER TRACING

- cross-reference database (function calls, references, structs)
- plug-in system to understand compiler's view of the world
- `@compile {:no_warn_undefined, OptionalDependency}`
- https://github.com/sasa1977/boundary

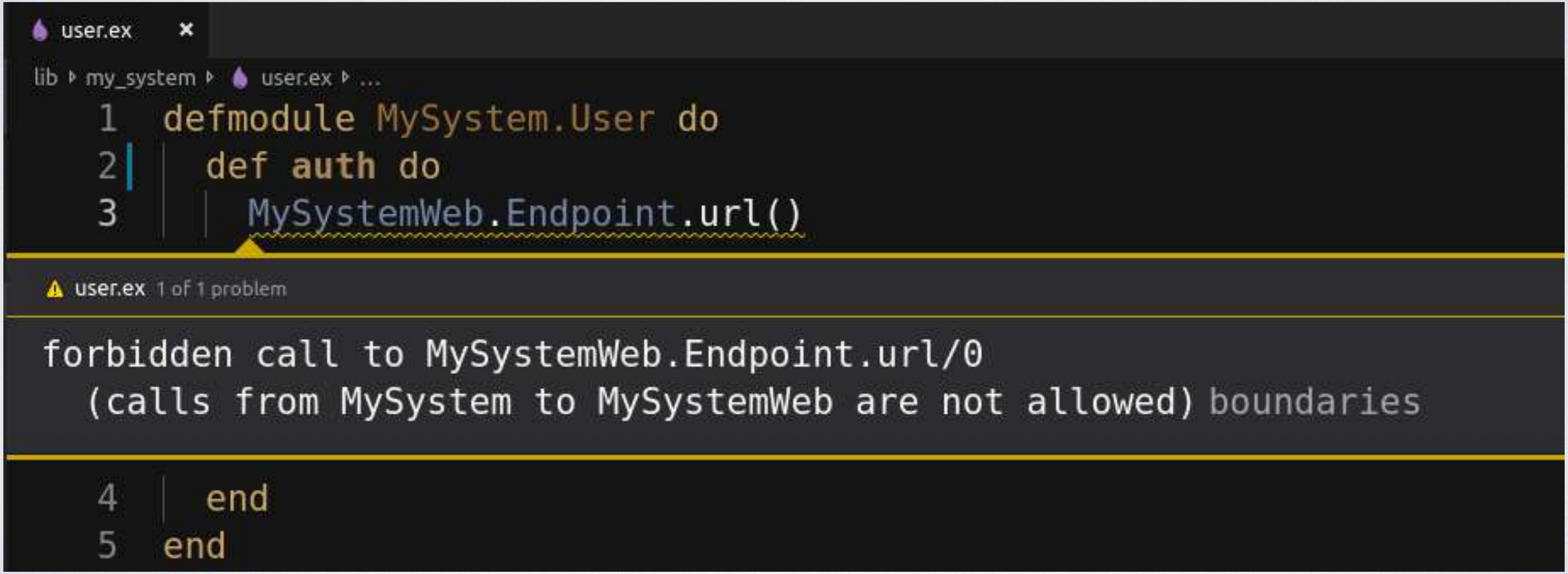
BOUNDARY

```elixir
defmodule MySystem do
  use Boundary, deps: [], exports: []
  # ...
end

defmodule MySystemWeb do
  use Boundary, deps: [MySystem], exports: [Endpoint]
  # ...
end

defmodule MySystem.Application do
  use Boundary, deps: [MySystem, MySystemWeb]
  # ...
end
```

BOUNDARY



user.ex ×

lib ▸ my_system ▸ user.ex ▸ ...

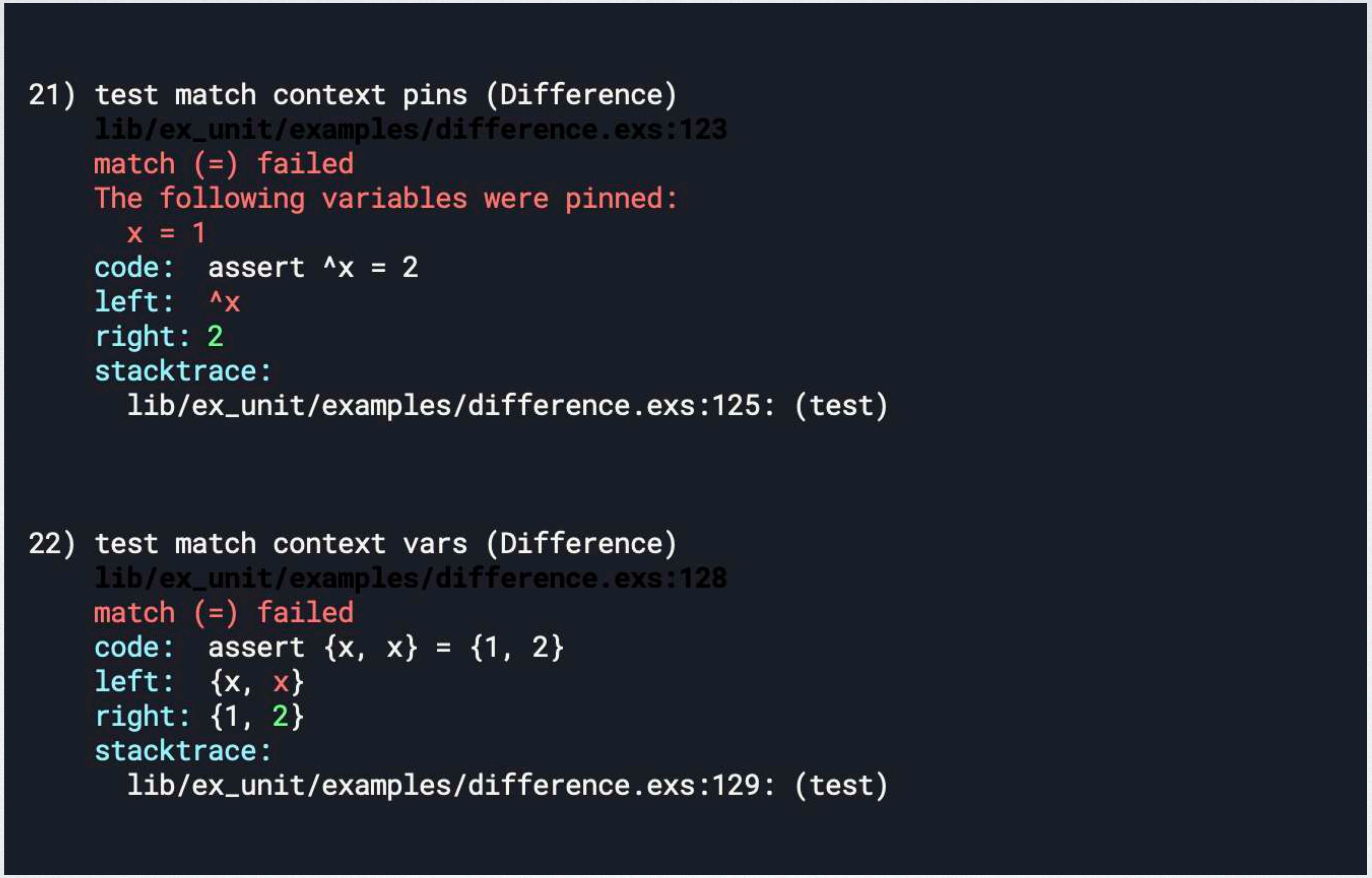
```elixir
defmodule MySystem.User do
  def auth do
    MySystemWeb.Endpoint.url()
```

⚠ user.ex 1 of 1 problem

```
forbidden call to MySystemWeb.Endpoint.url/0
  (calls from MySystem to MySystemWeb are not allowed) boundaries
```

```elixir
  end
end
```

ELIXIR 1.10 - EXUNIT DIFFING



```
21) test match context pins (Difference)
    lib/ex_unit/examples/difference.exs:123
    match (=) failed
    The following variables were pinned:
      x = 1
    code:  assert ^x = 2
    left:  ^x
    right: 2
    stacktrace:
      lib/ex_unit/examples/difference.exs:125: (test)



22) test match context vars (Difference)
    lib/ex_unit/examples/difference.exs:128
    match (=) failed
    code:  assert {x, x} = {1, 2}
    left:  {x, x}
    right: {1, 2}
    stacktrace:
      lib/ex_unit/examples/difference.exs:129: (test)
```

ELIXIR 1.11

- Scheduled for June 2020
- `Calendar.stfrtime/3`
- `is_struct/2` and `map.field` in guards
- Logger improvements
- Enforce application boundaries

- 1.11
  - Deprecate `Foo.bar` in favor of `Foo.bar()`
  - Deprecate `&Foo.bar()/0` in favor of `&Foo.bar`
- 1.12
  - Deprecate `mod.foo` in favor of `mod.foo()`
  - Deprecate `map.foo()` in favor of `map.foo`
- v2.0
  - Change the AST, so `foo.bar` has `nil` as args (the same as variables)

ELIXIR 1.11 - LOGGER

- Thanks to Łukasz Jan Niemier - `@hauleth`
- Support all levels of OTP logger
- Support structured logging by logging maps or keyword lists
- Allow level to be set per module with `Logger.put_module_level/2`

COMMUNITY

- HexDiff
- Lumen - WebAssembly BEAM implementation
- Mint - functional HTTP client
- ElixirLS
- Configuration - github.com/keathley/vapor and github.com/gmtprime/skogsra

COMMUNITY - NERVES

- Embedded framework in Elixir
- Recent updates - `nerves_pack` and `VintageNet`
- Documentation
- Remote edge computing
- Nerves on GRiSP 2 and other industrial hardware platforms

COMMUNITY - BROADWAY

- Tool for building data ingestion and data processing pipelines
- Builds on top of GenStage
- Support for for SQS, Kafka, RabbitMQ, Cloud PubSub
- Back-pressure, batching, graceful shutdown, testing, rate limiting, …

ELIXIR CORE & ECOSYSTEM UPDATE

Code BEAM SF 2020