News from

Elixir & Dashbit

@josevalim / @elixirlang / @dashbit

Elixir v1.10

Elixir v1.10

- Jan / 2020
- > 900 Contributors
- > 10 650 packages on hex.pm
- > 1 300 000 000 downloads

Elixir v1.10

- Erlang/OTP 21+ requirement
- Improvements around releases
- Compilation tracers
- Compile-time configuration
- ExUnit pattern diffing

Erlang/OTP 21+ requirement

- Integration with Erlang's new logger
  - As of v1.10, all log levels, metadata, and messages are shared between Elixir and Erlang
  - Thanks to @hauleth!
- Added is_map_key/2 and is_struct/1 as guards

Improvements to releases

- Allow opt-out from dual boot system
- Support for release overlays
- Accept RELEASE_DISTRIBUTION=none
- Add :tar to support release steps

Compilation tracers

```
defmodule MyMod do
  if OtherMod.some_condition? do
    def some_fun do
      ...
    end
  end
end
```

Compilation tracers

```elixir
defmodule MyTracer do
  def trace(event, compile_env)
end
```

Compilation tracers

- Boundary - github.com/sasajuric/boundary
- import2aliases - github.com/hexpm/hexpm/pull/887
- Tutorial - blog.appsignal.com/2020/03/10/building-compile-time-tools-with-elixir-compiler-tracing-
- See “Code" module docs

Compilation environment

```elixir
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
```

Compilation environment

```
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
```

Compilation environment

```elixir
defmodule MyApp.DBClient do
  def start_link(opts \\ []) do
    host = Keyword.put_new(opts, :host, "db.local")
    SomeLib.DBClient.start_link(host: host)
  end
end
```

Compilation environment

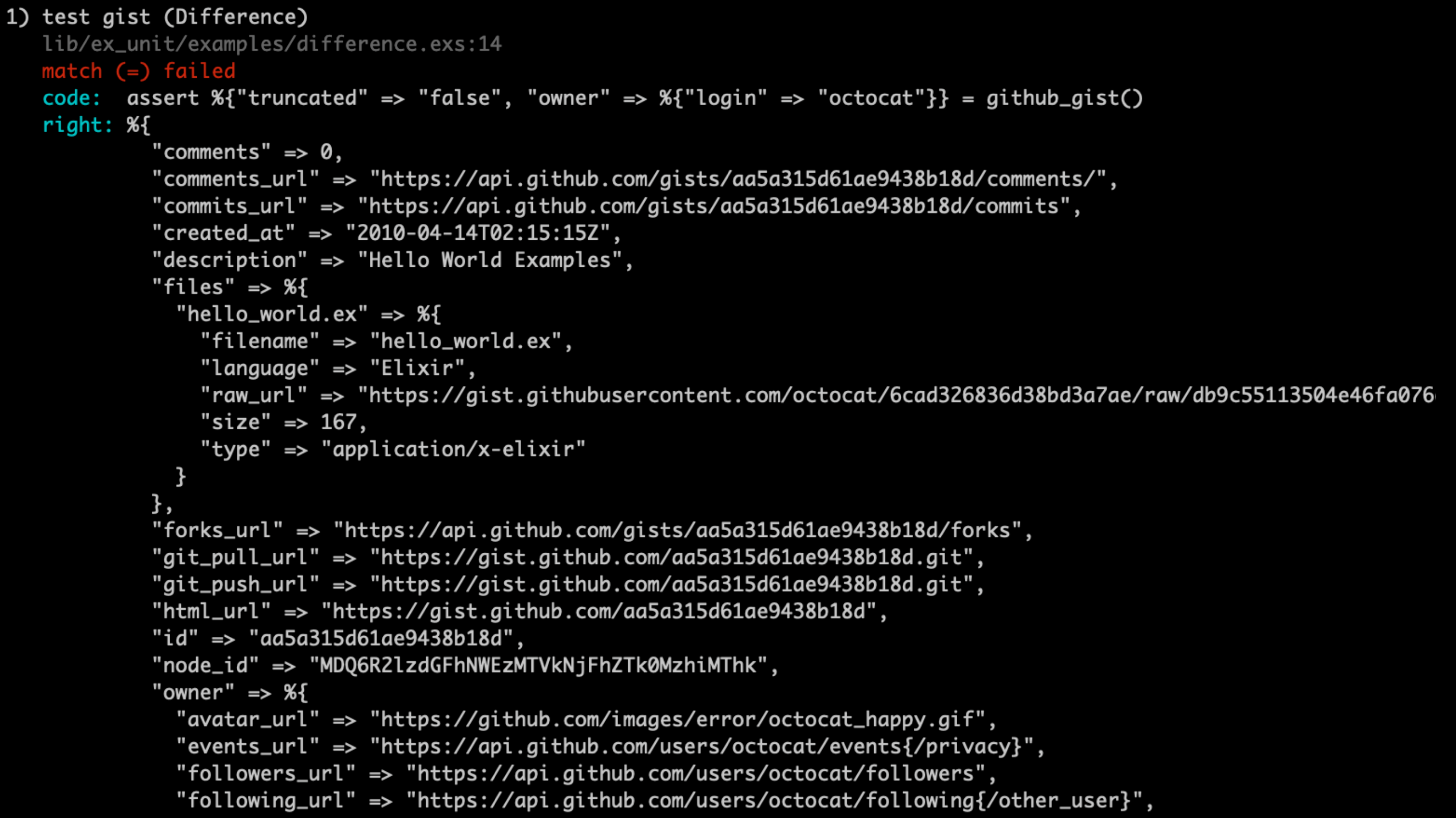
```elixir
defmodule MyApp.DBClient do
  @db_host Application.compile_env(:my_app, :db_host, "db.local")

  def start_link() do
    SomeLib.DBClient.start_link(host: @db_host)
  end
end
```

Compilation environment: stay ahead!

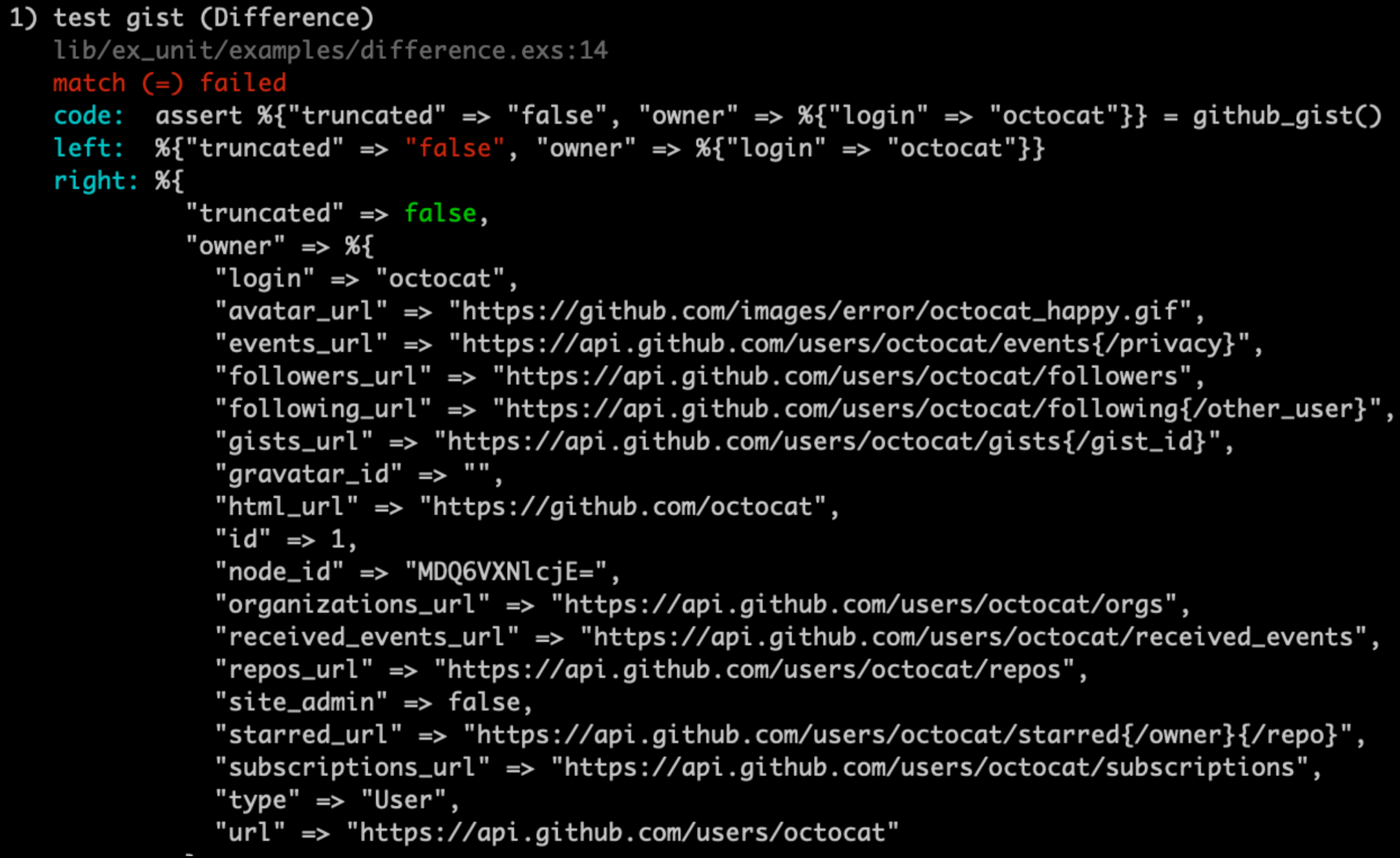
1. Don’t use the application environment
2. If you have to use it, use it at runtime
3. If you have to use it at compile-time, use compile_env

Compilation environment

```
test "gist" do
  assert %{
          "truncated" => "false",
          "owner" => %{"login" => "octocat"}
         } = github_gist()
end
```



```
1) test gist (Difference)
   lib/ex_unit/examples/difference.exs:14
   match (=) failed
   code:  assert %{"truncated" => "false", "owner" => %{"login" => "octocat"}} = github_gist()
   right: %{
            "comments" => 0,
            "comments_url" => "https://api.github.com/gists/aa5a315d61ae9438b18d/comments/",
            "commits_url" => "https://api.github.com/gists/aa5a315d61ae9438b18d/commits",
            "created_at" => "2010-04-14T02:15:15Z",
            "description" => "Hello World Examples",
            "files" => %{
              "hello_world.ex" => %{
                "filename" => "hello_world.ex",
                "language" => "Elixir",
                "raw_url" => "https://gist.githubusercontent.com/octocat/6cad326836d38bd3a7ae/raw/db9c55113504e46fa076
                "size" => 167,
                "type" => "application/x-elixir"
              }
            },
            "forks_url" => "https://api.github.com/gists/aa5a315d61ae9438b18d/forks",
            "git_pull_url" => "https://gist.github.com/aa5a315d61ae9438b18d.git",
            "git_push_url" => "https://gist.github.com/aa5a315d61ae9438b18d.git",
            "html_url" => "https://gist.github.com/aa5a315d61ae9438b18d",
            "id" => "aa5a315d61ae9438b18d",
            "node_id" => "MDQ6R2lzdGFhNWEzMTVkNjFhZTk0MzhiMThk",
            "owner" => %{
              "avatar_url" => "https://github.com/images/error/octocat_happy.gif",
              "events_url" => "https://api.github.com/users/octocat/events{/privacy}",
              "followers_url" => "https://api.github.com/users/octocat/followers",
              "following_url" => "https://api.github.com/users/octocat/following{/other_user}",
```



```
1) test gist (Difference)
   lib/ex_unit/examples/difference.exs:14
   match (=) failed
   code:  assert %{"truncated" => "false", "owner" => %{"login" => "octocat"}} = github_gist()
   left:  %{"truncated" => "false", "owner" => %{"login" => "octocat"}}
   right: %{
            "truncated" => false,
            "owner" => %{
              "login" => "octocat",
              "avatar_url" => "https://github.com/images/error/octocat_happy.gif",
              "events_url" => "https://api.github.com/users/octocat/events{/privacy}",
              "followers_url" => "https://api.github.com/users/octocat/followers",
              "following_url" => "https://api.github.com/users/octocat/following{/other_user}",
              "gists_url" => "https://api.github.com/users/octocat/gists{/gist_id}",
              "gravatar_id" => "",
              "html_url" => "https://github.com/octocat",
              "id" => 1,
              "node_id" => "MDQ6VXNlcjE=",
              "organizations_url" => "https://api.github.com/users/octocat/orgs",
              "received_events_url" => "https://api.github.com/users/octocat/received_events",
              "repos_url" => "https://api.github.com/users/octocat/repos",
              "site_admin" => false,
              "starred_url" => "https://api.github.com/users/octocat/starred{/owner}{/repo}",
              "subscriptions_url" => "https://api.github.com/users/octocat/subscriptions",
              "type" => "User",
              "url" => "https://api.github.com/users/octocat"
```

ExUnit Pattern Diffing

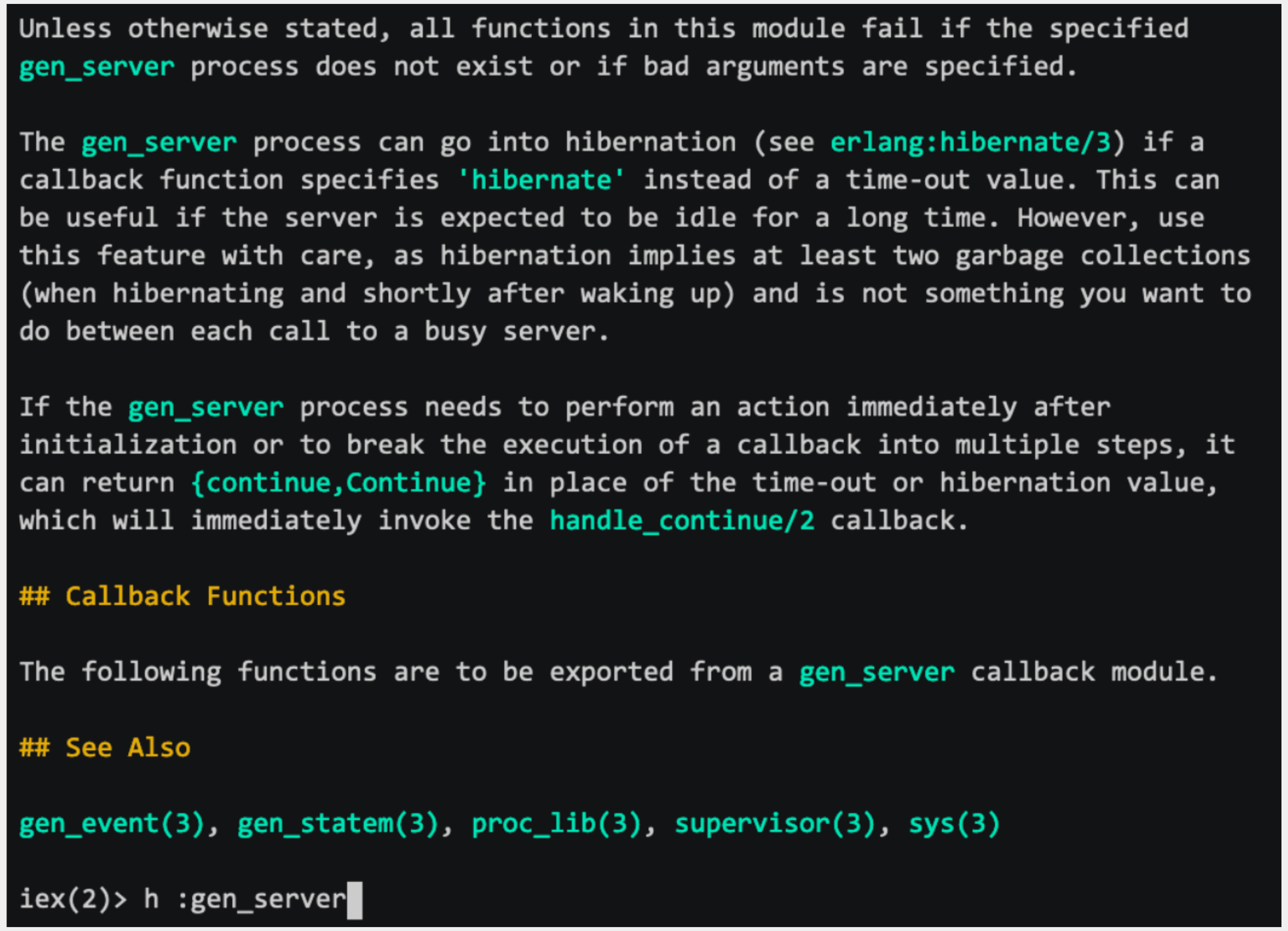
- Thanks to @lexmag for ExUnit value diffing
- Thanks to @ggcampinho and @jeremyowensboggs for ExUnit pattern diffing

Elixir v1.11

Elixir v1.11

- Calendar.strftime/3 for datetime formatting
- map.field and is_struct/2 are allowed in guards
- Add notice, critical, alert and emergency log levels
- Support structured logging
- Warn if using modules from non-dependencies

Elixir v1.11: Erlang docs in IEx



```
Unless otherwise stated, all functions in this module fail if the specified
gen_server process does not exist or if bad arguments are specified.

The gen_server process can go into hibernation (see erlang:hibernate/3) if a
callback function specifies 'hibernate' instead of a time-out value. This can
be useful if the server is expected to be idle for a long time. However, use
this feature with care, as hibernation implies at least two garbage collections
(when hibernating and shortly after waking up) and is not something you want to
do between each call to a busy server.

If the gen_server process needs to perform an action immediately after
initialization or to break the execution of a callback into multiple steps, it
can return {continue,Continue} in place of the time-out or hibernation value,
which will immediately invoke the handle_continue/2 callback.

## Callback Functions

The following functions are to be exported from a gen_server callback module.

## See Also

gen_event(3), gen_statem(3), proc_lib(3), supervisor(3), sys(3)

iex(2)> h :gen_server
```

- Thanks to the Erlang/OTP team
- Thanks to the Erlang Ecosystem Foundation - Documentation WG

Dashbit

Broadway v0.6

Broadway v0.6

- Date-ingestions and data-pipelines in Elixir made easy
- Supports: **Kafka**, AmazonSQS, RabbitMQ and Google CPS
- **v0.6**: supports rate-limiting (thank you @whatyouhide)
- **v0.6**: built-in telemetry events

Ecto v3.4

Ecto v3.4

- Built-in support to MSSQL via TDS
  Thanks to the massive effort from @mjaric
- Query language now supports JSON/embed expressions:
- from u in User, where: u.settings["panel_enabled"]
- from u in User, where: u.settings.panel_enabled

Phoenix-related

Phoenix Related

- A new authentication solution for Phoenix: dashbit.co/blog/a-new-authentication-solution-for-phoenix
- We have another open source project to announce really soon. Stay posted!

Thank you!

Elixir & Dashbit

@josevalim / @elixirlang / @dashbit