SLACK INTEGRATIONS IN ELIXIR/PHOENIX

or bots, slash commands, and webhooks—ohmy.

TYPES OF INTEGRATIONS

Bots

Slash commands

Webhooks

BOTS

Actual user, but interacts through Slack APIs

https://github.com/BlakeWilliams/Elixir-Slack

https://medium.com/carwow-product-engineering/getting-started-writing-a-slack-bot-with-elixir-b8877072f038#.arskooayq

SETTING UP THE PROJECT

```
$ mix new elixirbot –sup
```

Add dependencies, add worker child, etc.

Make bot user in slack.

Get API token--don’t commit this!

Skipping some steps here. Read README(s).

BUILDING THE BOT

```
# lib/slack.ex
defmodule Elixirbot.Slack do
  use Slack

  @token Application.get_env(:elixirbot,
__MODULE__)[:token]


  def start_link, do: start_link(@token, [])
end
```

BUILDING THE BOT

```
# lib/slack.ex
defmodule Elixirbot.Slack do
  ...

  def handle_message(message = %{type: "message"}, slack, state) do
    if Regex.run ~r/<@#{slack.me.id}>:?\shi/, message.text do
      send_message("<@#{message.user}> hello!", message.channel,
slack)
    end

    {:ok, state}
  end
end
```

BUILDING THE BOT

```
# lib/slack.ex
defmodule Elixirbot.Slack do
  ...
  def handle_message(%{type: “message”} ...
  def handle_message(_message, _slack, state) do
    {:ok, state}
  end
end
```

SLASH COMMANDS

Type /okcrb -> hits your server

Note: Some features require HTTPS and not a self-signed cert.

Pro tip! ngrok.com - “secure tunnels to localhost”

$ ngrok http 4000

$ https://a49052e4.ngrok.io -> localhost:4000

SLASH COMMANDS - THE SERVER

```
$ mix phoenix.new my_app


# web/router.ex
scope "/api", MyApp do
  pipe_through :api


  resources “/commands", CommandController, only:
[:create]
end
```

SLASH COMMANDS - THE SERVER

```
# web/controllers/command_controller.ex
defmodule MyApp.CommandController do
  use MyApp.Web, :controller

  def create(conn, params) do
    # Do some logic here
    json conn, %{response_type: "in_channel",
text: "this is a slash command!"}
  end
end
```

WEBHOOKS

Two types: incoming and outgoing

Incoming - Event on your server -> post in Slack

Outgoing - Event in Slack -> post to your server

Setup webhook in Slack admin

Get webhook URL. (Don’t commit this, again.)

WEBHOOKS - THE SERVER

```
# web/router.ex
scope "/webhooks", MyApp do
  pipe_through :api
  resources "/pings", PingController, only:
[:create]
end
```

WEBHOOKS - THE SERVER

```
# web/controllers/ping_controller.ex
defmodule MyApp.PingController do
  use MyApp.Web, :controller

  def create(conn, _params) do
    data =
      %{text: "Webhook!"}
      |> Poison.encode!

    HTTPoison.start
    HTTPoison.post!(System.get_env("SLACK_WEBHOOK_URL"), data)

    json conn, :ok
  end
end
```

WEBHOOKS - THE SERVER

```
data =
  %{text: "Webhook!"}
  |> Poison.encode!

HTTPoison.start
HTTPoison.post!(System.get_env("SLACK_WEBHOOK_URL"), data)

VS

Task.start_link(fn ->
  data =
    %{text: "Webhook!"}
    |> Poison.encode!

  HTTPoison.start
  HTTPoison.post!(System.get_env("SLACK_WEBHOOK_URL"), data)
end)
```

WEBHOOKS

```
# Non-wrapped
[info] POST /webhooks/pings
[info] Sent 200 in 274ms

# Wrapped in Task.start_link
[info] POST /webhooks/pings
[info] Sent 200 in 114µs

                    ^ What
```

DEMO

oh please work

THANKS!

Sample projects for bot and server:

https://github.com/johnmosesman/okcrb_elixir_slack_server

https://github.com/johnmosesman/okcrb_elixirbot

@johnmosesman