

A landscape photograph showing a valley at sunset or sunrise. The sky is filled with warm, orange, and yellow hues. In the foreground, there are dark, silhouetted shapes that appear to be trees or bushes. In the middle ground, a small, dark building with a chimney is visible, emitting a plume of white smoke. The background consists of rolling hills and mountains under the colorful sky.

**Sister workshop**

# Sister

- BFF
- Playground for mobile
- Create endpoints, schemas and send fake data... or even more
- Let backend developers focus on important stuff

# **Why this workshop?**

**No mobile dev is  
actually doing so**

**(Well, only few of them)**

**Because the language?**  
**Because the framework?**  
**Because just backend fear?**

**Why me?**



*One does not simply walk into Mordor.*

**It's not that bad**

**It's actually easy**

**Because the language?**  
**Because the framework?**  
**Because just backend fear?**

# Agenda

- Setup
- IDEs
- Create endpoint
- Test and run
- Create documentation

# Setup

Sister readme

# IDEs

- VSCode + [ElixirLS](#)
- Rubymine + [Elixir plugin](#)
- Whatever you feel comfortable with

**Let's code**

# Create endpoint

- Router
- Controller
- View

# Router

- `sister_web/router.ex`
- Scopes

```
scope "/v3/worker", SisterWeb do
end
```

- Resource

```
resources("/workers", Workers.WorkerController, only: [:index, :show])
```

- `mix phx.routes`

# Controller

- Router -> WorkerController
- Gather data and perform tasks before rendering
- Action

```
@spec index(Plug.Conn.t(), any) :: Plug.Conn.t()
def index(conn, params)
```

- conn: Bunch of request data
- params: Request parameters

# Views

- WorkerController -> WorkerView
- Render data: html/json
- By default Phoenix will look for a view with the same name as the controller:
  - SisterWeb.Workers.WorkerController
  - SisterWeb.Workers.WorkerView
- Render Errors: SisterWeb.ErrorView

# Test and run

- Locally
- Staging
- Tests
- Kitt

# Locally

- Devkit/Devcloud
- No easier way

RUS 0  
CAN 0  
END 1

# Staging

LAUSANNE  
**2020**



# Testing (preferred way)

- mix test
- valid-one

```
get(conn, Routes.worker_path(conn, :index, auth_token: "valid-one"))
```

# Kitt

```
kitt -e s console ecs ecs-shell sister-http:latest  
sh
```

# Create documentation

- Schemas
- Assert against schemas
- Yaml

# Schemas

- doc/api/v3/schemas

# Assert against schemas

- import JsonSchema.Assertions
- assert\_schema("worker/workers/index.json", response)
- Validates our fake data
- When real backend developers do their job, they can be sure that they are not breaking the contract

# Yaml

- doc/api/v3/api.yaml
- doc/api/v3/paths/worker/workers/index.yaml
- Add examples on schemas  
"example": { "\$ref": "index.example.json" }
- mix docs
  - docker installed and running
  - It validates schemas and generates documentation

# Bonus track

- Localization
  - mix phraseapp
  - dgettext("leaves", "leave-detail-subtitle")
- mix format
- mix dializer

# References

Request life-cycle (Recomended)

Phoenix framework

Phoenix docs

Slack #elixir

@fertapric



**POR FIN TERMINÉ**