Lesson 3: Mix & CI/CD

Wannes Fransen & Tom Eversdijk

UC Leuven

2020

Mix

Project structure

CI / CD

What is mix?

- Build tool
- Create your application
- compile / test your code
- Dependency management
- Custom commands
- Configuration management
- ...

Sample usage

- mix new sample_application (- - sup)
- mix phx.new application_name (- - no-html, - - database)

- mix run
- mix test
- mix compile
- mix deps.get
- mix deps.compile

Mix

Project structure

CI / CD

Basic mix project structure

Umbrella projects

Instead of building a single large monolith, you can structure your code with multiple isolated contexts.

- poor man's microservices solution
- compiled and run under the same BEAM instance
- Dependencies between applications must be explicitly defined
- Degree of separation, not fully decoupled!

Different configurations in each application for the same dependency or use different dependency versions, then it is likely your codebase has grown beyond what umbrellas can provide.

Sample generated project - umbrella structure

mix phx.new demo –umbrella –database mysql

```
hello_umbrella
|-- _build
|-- apps
|   |-- hello                  => domain application
|   |   |-- ...
|   |-- hello_web              => web application
|   |   |-- ...
|-- config                     => shared config
|-- deps
```

Sample generated project - domain structure

```
hello
|-- lib
|   |-- hello
|   |   |-- foo_context      => context folder
|   |   |   |-- foo.ex       => foo-related modules
|   |   |-- foo_context.ex   => context module
|   |   |-- application.ex   => starts app processes
|   |   |-- repo.ex          => module for db operations
|   |   |-- ...
|   |-- hello.ex             => app interface
|-- priv
|   |-- repo
|   |   |-- migrations
|   |   |-- seeds.ex         => default data in your db
|   |   |-- ...
|-- test
|   |-- ...
```

Sample generated project - web structure

```
hello_web
|-- controllers
|   |-- foo_controller.ex
|   |-- bar_controller.ex
|-- templates
|   |-- foo
|   |   |-- index.html.eex
|   |   |-- ...
|   |-- bar
|   |   |-- index.html.eex
|   |   |-- ...
|-- views
|   |-- foo_view.ex
|   |-- bar_view.ex
```

General guidelines

- No domain code in controller
- No direct usage of Repo in your web project
- Controllers will use Contexts to communicate with your domain

Mix

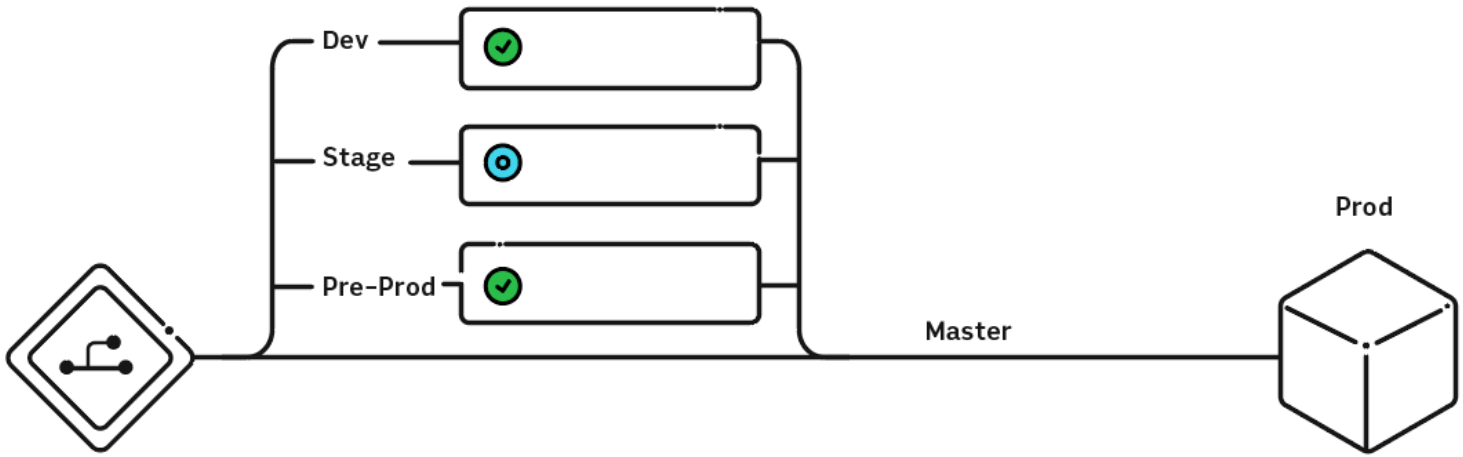
Project structure

CI / CD

What is Continuous Integration?

- [BLOG POST]
- multiple developers pushing small, frequent changes to a shared repository
- [COMMON PRACTICES]
- often comes with Continuous Deployment
- CircleCI build has generally multiple steps:
  - dependencies
  - testing
  - deployment

How your repo will look like



feel free to simplify this to following branches:

- dev
- prod

you can configure your CI/CD to do specific task on specific branches (e.g. only deploy prod branch)

Github repo demo

follow the guide in the guide folder