

Curso Elixir

Sección 13

MIX

Es una herramienta de compilación que se incluye con Elixir que proporciona tareas para crear, compilar, probar su aplicación, administrar sus dependencias y mucho más. Cuando instalamos elixir, también obtienes la herramienta mix.

Nuevo proyecto

Para crear un nuevo proyecto utilizamos el comando `mix new`

```
mix new kv --module KV
```

Con este comando se creará un carpeta kv y con la etiqueta `--module` especificamos que el nombre sea KV y no Kv

```
* creating README.md
* creating .formatter.exs
* creating .gitignore
* creating mix.exs
* creating lib
* creating lib/kv.ex
* creating test
* creating test/test_helper.exs
* creating test/kv_test.exs
```

Compilación del proyecto

El archivo mix.ex fue generado al crear el proyecto, este archivo es el responsable de copilar nuestro proyecto.

```
defmodule KV.MixProject do
  use Mix.Project

  def project do
    [
      app: :kv,
      version: "0.1.0",
      elixir: "~> 1.10",
      start_permanent: Mix.env == :prod,
      deps: deps()
    ]
  end

  # Run "mix help compile.app" to learn about applications
  def application do
    [
      extra_applications: [:logger]
    ]
  end

  # Run "mix help deps" to learn about dependencies
  defp deps do
    [
      # {:dep_from_hexpm, "~> 0.3.0"},
      # {:dep_from_git, git: "https://github.com/elixir-lang/my_dep.git", tag: "0.1.0"},
    ]
  end
end
```

```
end
end
```

En `mix.exs` se definen dos funciones públicas, que regresan la configuración, nombre y versión del proyecto. Además, hay una función privada la cual define las dependencias requeridas para el proyecto.

```
defmodule KV do
  @moduledoc """
  Documentation for KV.
  """

  @doc """
  Hello world.

  ## Examples

      iex> KV.hello()
      :world

  """
  def hello do
    :world
  end
end
```

Compilemos el proyecto con `mix compile`

```
# cd kv/
# mix compile

Compiling 1 file (.ex)
Generated kv app
```

El proyecto fue compilado, generando una aplicación llamada `kv app`. Todos los artefactos compilados se guardan en `_build`.

Interactuar con el proyecto

Una vez compilado el proyecto, podemos arrancar con `iex`. Dentro de la carpeta `kv` corre el siguiente comando:

```
# iex -S mix
iex> KV.hello
:world
```

Podemos agregar cambios al proyecto sin salirnos de `iex`, para que los cambios sean reflejados debemos de correr la función `recompile()`

```
iex> recompile()
Compiling 1 file (.ex)
:ok
iex> recompile()
:noop
```

Si algo ha sido compilado, veras el atomo `:ok`, de otro manera veras `:noop` si nada ha tenido cambios

Dependencias

La forma de agregar dependencias a nuestro proyecto es ingresando tuplas a la función privada `deps` dentro del archivo `mix.exs`

```
# Run "mix help deps" to learn about dependencies.
defp deps do
  [
    {:phoenix, "1.5"},
    {:phoenix_html, "~> 2.3"},
  ]
end
```

Una vez que agregamos las dependencias para el proyecto corremos el comando `# mix deps.get`

Entornos

Existen 3 tipos de entornos:

- `:dev`
- `:test`
- `:prod`

El entorno por defecto es `:dev`, para cambiar de entorno usamos la variable de entorno `MIX_ENV`

```
$ MIX_ENV=prod mix compile
```

Actividad 6

Crear aplicación para conectarnos a una base de datos:

Prerrequisitos:

- Elixir \geq 1.7
- Mysql \geq 5.2