

IF140303-Web Application Development

Session-03: Elixir Documentation and Testing

PRU/SPMI/FR-BM-18/0222

Alfa Yohannis



Overview of Elixir Documentation

- Elixir provides powerful tools for creating documentation directly within your code.
- `@moduledoc` and `@doc` are used to document modules and functions.
- Mix tasks like `mix docs` can generate HTML documentation from your code.

Adding Dependencies for Documentation



- To generate documentation, add `ex_doc` to your `mix.exs` dependencies.

```
1  defp deps do
2    [
3      {:ex_doc, "~> 0.12"}
4    ]
5  end
```

@moduledoc: Module Documentation

- Use @moduledoc at the top of your module to provide an overview.
- Describes the purpose, usage, and any relevant details of the module.

Example: @moduledoc

```
1  defmodule Lottery do
2  @moduledoc """
3  This module provides functionalities for managing a lottery
   system.
4  It includes functions for creating, shuffling, checking for
   numbers,
5  saving, loading, and distributing numbers within the lottery
   pool.
6  """
7  end
```

@doc: Function Documentation

- @doc is used to document individual functions.
- Include descriptions, parameters, return values, and examples.

Example: @doc

```
1  @doc """
2  Generates a pool of lottery numbers with different pots.
3
4  ## Returns
5  - A list of lottery numbers with their respective pot
6    numbers.
7
8  ## Examples
9  iex> Lottery.generate_pool()
10  ["Number 1 in Pot 1", "Number 2 in Pot 1", ...]
11  """
12  def generate_pool do
13    # implementation
14  end
```

Generating Documentation

- Run `mix docs` to generate HTML documentation.
- The output will be stored in the `doc` directory.
- View it by opening `index.html` in your browser.

Overview of Testing in Elixir

- Elixir uses `ExUnit` for writing and running tests.
- `ExUnit.Case` is used to define a test module.
- Tests are written as functions prefixed with `test`.

Basic Example of a Test

```
1  defmodule LotteryTest do
2    use ExUnit.Case
3
4    test "generate_pool creates 20 lottery numbers" do
5      assert length(Lottery.generate_pool()) == 20
6    end
7  end
```

Doctests in Elixir

- Doctests allow you to run the examples in your documentation as tests.
- Ensure the examples in your `@doc` annotations are correct.
- Use `doctest ModuleName` in your test module.

Adding Doctests to Your Tests

```
1  defmodule LotteryTest do
2    use ExUnit.Case
3    doctest Lottery
4  end
```

Running Tests

- Run `mix test` to execute all your tests.
- Ensure your code is thoroughly tested and your documentation examples are correct.

Summary

- We covered how to document Elixir code using `@moduledoc` and `@doc`.
- We explored how to write and run tests, including using doctests.
- Proper documentation and testing are essential for maintainable and reliable code.