

Функциональное программирование на Elixir

Владимир Цыганов
AnjLab Tech Talks
г. Владимир

Erlang

- Разработан для применения в распределенных отказоустойчивых системах
- Концепция “Let it crash”
- набор библиотек OTP для построения приложений, надежных на 99,999999%

Elixir

- Создан в 2012 г. Хосе Валимом (José Valim)
- Компилируется в байт-код для Erlang VM (BEAM)
- Вызов Erlang-функций без влияния на время исполнения кода
- Полиморфизм, макросы, pattern matching

Типы данных

- Value-типы: int, float, atom, range, regex
- Системные типы: PID, reference
- Коллекции: tuple, list, map, binary
- Функции

Pattern matching

```
x = 1
```

```
[head|tail] = [1, 2, 3, 4]
```

```
handle_open = fn
```

```
  { :ok, file } -> "First line: #{IO.read(file, :line)}"
```

```
  { _, error } -> "Error: #{:file.format_error(error)}"
```

```
end>
```

```
handle_open.(File.open("some_file"))
```

Модули

```
defmodule Factorial do
  def of(0), do: 1

  def of(n) when n > 0 do
    n * of(n-1)
  end

  def of(n) when n < 0 do
    IO.puts "Wrong number"
  end
end
```

Pipe оператор

```
def login(conn, user) do  
  conn  
  |> put_current_user(user)  
  |> put_session(:user_id, user.id)  
  |> configure_session(renew: true)  
end
```

МНОГОПОТОЧНОСТЬ

```
defmodule Spawn do
  def greet do
    receive do
      {sender, msg} ->
        send sender, { :ok, "Hello, #{msg}" }
        greet
    end
  end
end
```

```
pid = spawn(Spawn, :greet, [])
send pid, {self, "World!"}
receive do
  { :ok, message} ->
    IO.puts message
  after 500 ->
    IO.puts "The greeter has gone away"
end
```


Применение

- Микросервисы
- High-load сервисы
- Web (Phoenix Framework)

Phoenix Framework

- MVC веб фреймворк
- OTP-приложение
- Быстрый, многопоточный

Phoenix Framework

- Endpoint
- Router
- Controller
- View
- Template
- Channel

Спасибо за внимание

anjab.com

twitter.com/AnjLab

Ресурсы:

elixir-lang.org

phoenixframework.org