CRYSTAL

CodeCamp | SF2017

Into the wild

Let’s do some JSON, DB and IO

manas academyX Twentify stickermule

JSON in Crystal

```
alias JSON::Type =
                    Nil | String | Bool | Int64 | Float64
                  | Array(JSON::Type)
                  | Hash(String, JSON::Type)
```

JSON in Crystal

```
{"foo": [111, 222, 333]}

JSON::Type
    json.as(Hash)["foo"].as(Array)[2].as(Int64)

JSON::Any
    json["foo"][2].as_i64
```

JSON in Crystal

```
JSON.parse(IO | String)       # => JSON::Any

JSON.parse_raw(IO | String)   # => JSON::Type
```

JSON in Crystal

```
parser = JSON::PullParser.new(IO | String)
parser.read_object do |key|
  parser.read_array do
    parser.read_string
  end
end
```

JSON in Crystal

Mappings

```
class User
  JSON.mapping(
    id: Int32,
    name: String
  )
end
```

JSON in Crystal

```
User.from_json(IO | String)

User.new(JSON::PullParser)
```

JSON in Crystal

```
string = JSON.build do |json|
  json.object do
    json.field "foo" do
      json.array { json.number 8 }
    end
  end
end
```

JSON in Crystal

```
JSON.build(io) do |json|
  json.object do
    json.field "foo" do
      json.array { json.number 8 }
    end
  end
end
```

DEMO: JSON parsing

Crystal DB

- Uniform API to access databases
- Enforces efficient resource usage
- Single implementation of common patterns

Crystal DB

```
require "mysql"

DB.open("mysql://usr:pwd@server/db") do |db|
  db.exec
    "DELETE FROM users WHERE id = ?", user_id
  db.scalar "SELECT count(*) FROM users"
end
```

Crystal DB

```
db.query "SELECT id, name FROM users" do |rs|
  rs.each do
    id = rs.read(Int32)
    name = rs.read(String)
  end
end
```

Crystal DB

```
name, age =
  db.query_one "SELECT name, age
                FROM users
                WHERE id = ?", user_id
               as: {String, Int32}
```

Crystal DB

```
users =
  db.query_all "SELECT id, name
                FROM users",
               as: {Int32, String}
```

Crystal DB

```
class User
  DB.mapping(
    id: Int32,
    name: String,
    age: Int32?
  )
end
```

Crystal DB

```
user =
  db.query_one "SELECT *
                FROM users
                WHERE id = ?", user_id
               as: User
```

DEMO: Crystal DB

We are Manas.

We build unconventional software_

https://manas.tech