AMSTERDAM |> ELIXIR

CODE COMPILING & HOT CODE RELOADING

PAUL ENGEL

DEVELOPER AT BETTY BLOCKS

RUBY AND JAVASCRIPT SINCE 2007

51 GITHUB SOURCE REPOSITORIES

ELIXIR SINCE 2016

CODE COMPILATION

WHY SHOULD WE?

WE CONSUME APPLICATIONS

APPLICATIONS CONSUME USER INPUT

USER INPUT NEEDS TO BE INTERPRETED

INTERPRETATION TAKES TIME

WELL-KNOWN EXAMPLE

EJS

handlebars.js

mustache.js

JAVASCRIPT TEMPLATE ENGINES

doT.js

liquid.js

hogan.js

BACK IN 2012

MUSTACHE.JS INTERNET EXPLORER 6
RENDERED IN 50 SECONDS
SO I WROTE TEMPLAYED.JS
RENDERED WITHIN A SECOND

USING HANDLEBARS.JS

USING HANDLEBARS.JS

```
var source   = '<h1>{{title}}</h1><p>{{body}}</p>';
var template = Handlebars.compile(source);
var context  = {title: 'My New Post', body: 'This is my first post!'};
var html     = template(context);

#=> <h1>My New Post</h1><p>This is my first post!</p>
```

USING HANDLEBARS.JS

```
var source   = '<h1>{{title}}</h1><p>{{body}}</p>';
var template = Handlebars.compile(source);
var context  = {title: 'My New Post', body: 'This is my first post!'};
var html     = template(context);

#=> <h1>My New Post</h1><p>This is my first post!</p>
```

"USER" INPUT

USING HANDLEBARS.JS

```
var source   = '<h1>{{title}}</h1><p>{{body}}</p>';
var template = Handlebars.compile(source);
var context  = {title: 'My New Post', body: 'This is my first post!'};
var html     = template(context);

#=> <h1>My New Post</h1><p>This is my first post!</p>
```

"USER" INPUT

CODE COMPILATION

USING HANDLEBARS.JS

```
var source   = '<h1>{{title}}</h1><p>{{body}}</p>';
var template = Handlebars.compile(source);
var context  = {title: 'My New Post', body: 'This is my first post!'};
var html     = template(context);

#=> <h1>My New Post</h1><p>This is my first post!</p>
```

"USER" INPUT

CODE COMPILATION

INVOKING THE COMPILED FUNCTION

USING HANDLEBARS.JS

```
var source   = '<h1>{{title}}</h1><p>{{body}}</p>';
var template = Handlebars.compile(source);
var context  = {title: 'My New Post', body: 'This is my first post!'};
var html     = template(context);

#=> <h1>My New Post</h1><p>This is my first post!</p>
```

"USER" INPUT

CODE COMPILATION

INVOKING THE COMPILED FUNCTION

A SMALL RECAP

A SMALL RECAP

SOURCE = INPUT

CONTEXT: "DON'T FORGET ME, PLEASE!"

HTML = RESULT

A SMALL RECAP

INPUT + CONTEXT > RESULT

A SMALL RECAP

INPUT + CONTEXT > RESULT

requires interpretation

A SMALL RECAP

INPUT + CONTEXT > RESULT

time consuming

A SMALL RECAP

INPUT + CONTEXT > RESULT

constant

A SMALL RECAP

INPUT ≠ CODE :'(

A SMALL RECAP

INPUT > FUNCTION :)

A SMALL RECAP

INPUT > FUNCTION :)

requires no interpretation (w00t!)

A SMALL RECAP

INPUT > FUNCTION > FUNCTION(CONTEXT) > RESULT

A SMALL RECAP

INPUT > FUNCTION > FUNCTION(CONTEXT) > RESULT

A SMALL RECAP

```
html1 = template(context1)
html2 = template(context2)
html3 = template(context3)
```

ELIXIR CODE COMPILATION

INPUT > FUNCTION

INPUT ≠ CODE

INPUT = TEXT CONFIRMING A SYNTAX?

INPUT = LOADED FROM A DATABASE?

INPUT = ???

MY NAME IS ABSTRACT SYNTAX TREE

… BUT MY FRIENDS CALL ME AST

INPUT > AST > FUNCTION

INPUT > AST > FUNCTION

http://bit.ly/2SucEXi

quote **MACRO**

Code **MODULE**

```
iex(1)> ast = quote do: 1 + 1
{:+, [context: Elixir, import: Kernel], [1, 1]}
```

```
iex(2)> Code.eval_quoted(ast)
{2, []}
```

```
iex(3)> ast = quote do: sum(1, 2 + 3)
{:sum, [], [1, {:+, [context: Elixir, import: Kernel], [2, 3]}]}
```

TIME FOR iex

http://bit.ly/2SucEXi

HOT CODE RELOADING

USING JAVASCRIPT

INPUT > FUNCTION > FUNCTION(CONTEXT) > RESULT

USING ELIXIR

AST > FUNCTION > FUNCTION.(ARGS) > RESULT

AST > FUNCTION > FUNCTION.(ARGS) > RESULT

time consuming

CODE COMPILATION = TIME CONSUMING *snif*

DISCORD'S FASTGLOBAL 💪

DISCORD'S FASTGLOBAL 😢

compilation time consuming

ENTER CLUSTORAGE

DISTRIBUTED CODE COMPILATION :)

ELIXIR CLUSTER

DESIGNATED COMPILER NODE

COMPILES TO BYTECODE

HOT CODE RELOAD CLUSTER NODES

:code **ERLANG MODULE**

```
:code.purge(module)
:code.delete(module)
:code.load_binary(module, nil, binary)
```

TMUX TIME! :)

https://github.com/archan937/clustorage