elixir dynamic module call

Asked 7 years, 11 months ago Modified 7 years, 11 months ago

Viewed 2k times

6

How can I call function func() in a module called App.Reporting.Name

based on the string "name" which is not known until runtime

using String.to_atom or to_existing_atom does not work :

```
 alias App.Reporting.Name
 module = "name" |> String.capitalise |>
String.to_atom
 apply(module, :func, [])
```

Without the alias, this does not work either

```
module = "App.Reporting.Name" |> String.to_atom
apply(module, :func, [])
```

I get an (UndefinedFunctionError) and (module :"App.Reporting.Name" is not available)

thanks

dynamic module elixir

Share

Improve this question

Follow

asked Dec 27, 2016 at 21:47

user2725682
139 ● 1 ● 10

2 Answers

Sorted by: Highest score (default)

9

Your second approach is almost correct, you just need to prefix `Elixir.` because `App.Reporting.Name` is equal to `:"Elixir.App.Reporting.Name"`, not `:"App.Reporting.Name"` since Elixir prefixes all module names (names starting with an uppercase letter) with `Elixir.` before turning it into an atom:

```
iex(1)> App.Reporting.Name ==
:"App.Reporting.Name"
false
iex(2)> App.Reporting.Name ==
:"Elixir.App.Reporting.Name"
true
```

So, this code should work:

```
module = "Elixir.App.Reporting.Name" |>
String.to_atom
apply(module, :func, [])
```

and so should this:

```
module = Module.concat(App.Reporting, "name" |>
String.capitalize |> String.to_atom)
apply(module, :func, [])
```

Share Improve this answer

Follow

answered Dec 28, 2016 at 4:37

Dogbert

222k ● 42 ● 417 ● 414

1

The reason yours isn't working is because the `String.to_atom` does just that, turns a string into an atom. Because there is no module called "App.Reporting.Name" it's most likely `App.Reporting.Name` it errors.

Not sure if this is the *best* way to do this, just one that sprang to mind. But you could do something like this:

```
iex(2)> module = "Casing"
"Casing"
iex(3)> Module.concat(String, "#{module}") |>
apply(:upcase, ["test sentence"])
"TEST SENTENCE"
```

Another solution could be to create a macro that automatically does this process, however that is not something I am that great at so you will have to go through the docs here for that one.

Share Improve this answer

Follow

answered Dec 27, 2016 at 22:58

Harrison Lucas

2,951 ● 20 ● 25