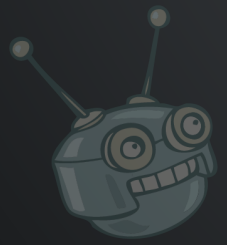
TOOLING FOR REAL WORLD HASKELL ENGINEERING DISCIPLINE

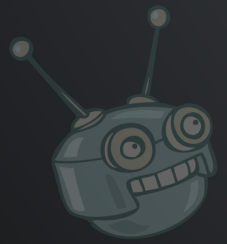
Enforcing project-specific rules over time

- Trevis Elser
- trevis@flipstone.com

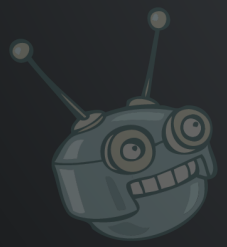


HENFORCER

- https://github.com/flipstone/henforcer
- https://hackage.haskell.org/package/henforcer

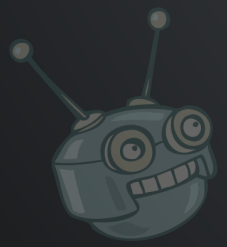


WHAT IS IT?



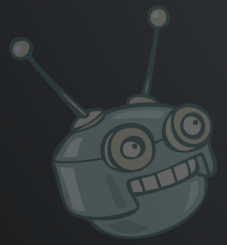
WHAT IS IT?

- GHC plugin



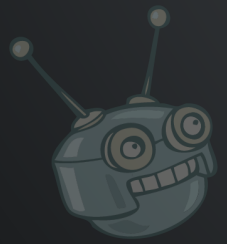
WHAT IS IT?

- GHC plugin
- Enforces project-specific rules at compile time

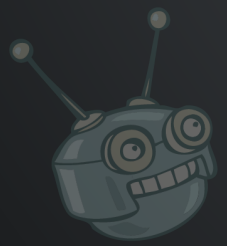


WHAT IS IT?

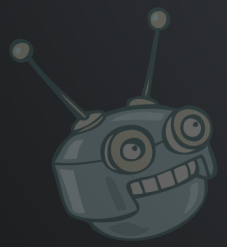
- GHC plugin
- Enforces project-specific rules at compile time
- Explicitly configured, no defaults



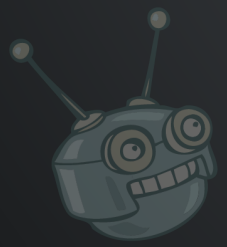
WHY A PLUGIN? HOW DID WE GET HERE?



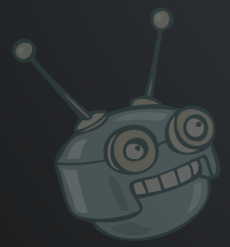
WHAT WE CHOSE TO ENFORCE – AND WHY



HADDOCK COVERAGE

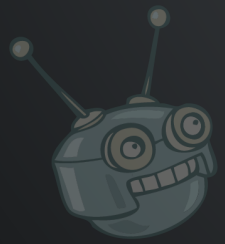


CONFIGURATION

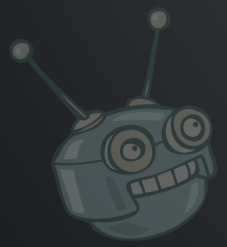


CONFIGURATION

```
[forAnyModule]
maximumExportsPlusHeaderUndocumented = 0
minimumExportsPlusHeaderDocumented = 1
```



CODE

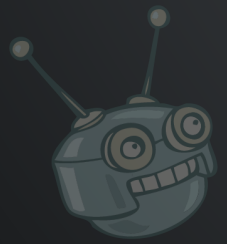


CODE

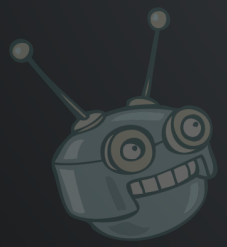
```haskell
{- |
Module      : Foo
-}
module Foo where

foo = putStrLn "abc"

{- | bar does a thing
-}
bar :: IO ()
bar = undefined
```

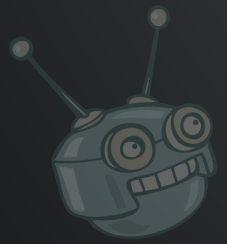


COMPILATION RESULT

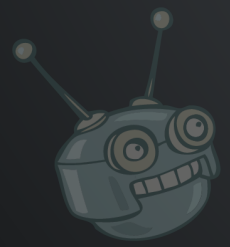


COMPILATION RESULT

```
[1 of 7] Compiling Foo                ( src/Foo.hs, dist-newstyle/b
<generated>: error: [HEN-95002]
    There were too many undocumented exports.
    The maximum allowed is: 0
    The number undocumented is: 1
```

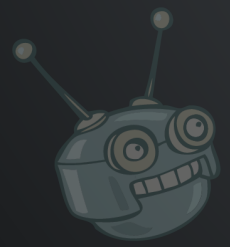


IMPORT REGULARITY

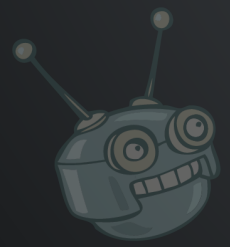


IMPORT REGULARITY

```
[forAnyModule]
allowedAliasUniqueness = { allAliasesUniqueExcept = [ "Export" ]
[[forAnyModule.allowedQualifications]]
module = "PetStore.Pet.Model"
[[forAnyModule.allowedQualifications.importScheme]]
qualified =  {qualifiedPost = true}
alias="PetModel"
```

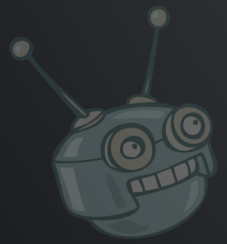


MODULE ENCAPSULATION

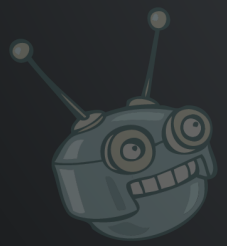


MODULE ENCAPSULATION

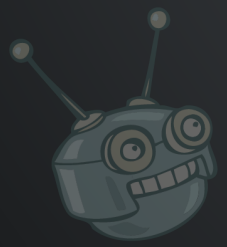
```
[forAnyModule]
encapsulatedTrees = ["PetStore.Pet"]
```



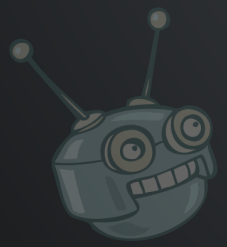
WHY WE SHOULD CARE



HCOVGUARD

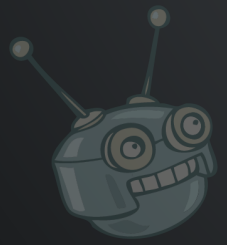


WHAT IS IT?



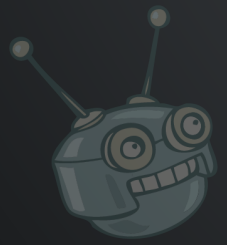
WHAT IS IT?

- Executable, test coverage rachet tool

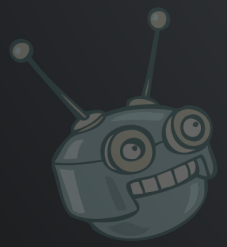


WHAT IS IT?

- Executable, test coverage rachet tool
- Builds on 'hpc' tool



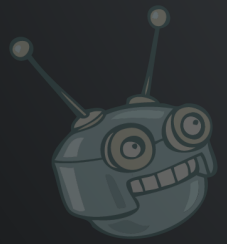
CONFIGURATION



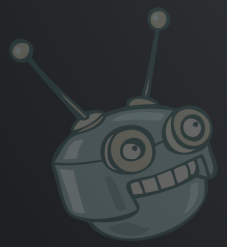
CONFIGURATION

```
[forAnyModule]

[forAnyModule.topLevel]
minimumCovered = 10
maximumUncovered = 5
```

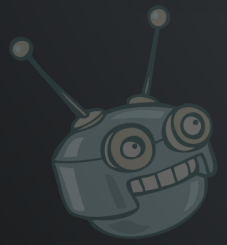


OUTPUT (VERBOSE)

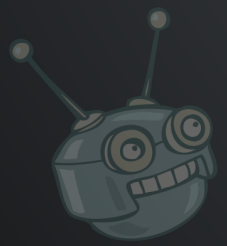


OUTPUT (VERBOSE)

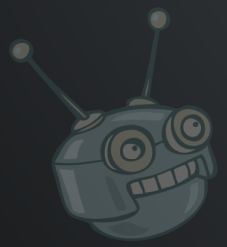
```
Module: Foo.Bar
  Expression: 209/209 covered
  Top-level:  6/6 covered
  Alternative:2/2 covered
  Local:      11/11 covered
  [PASSED] Foo.Bar
Module: Foo.Baz
  Expression: 327/345 covered
  Top-level:  9/9 covered
  Alternative:0/0 covered
  Local:      5/7 covered
Module: Foo.Baz
  [Hcovguard-3928] Maximum uncovered exceeded
    Category: expression
    Actual:   18 uncovered
    Allowed:  0 uncovered
```



WHY WE SHOULD CARE



INCREASING ENGINEERING DISCIPLINE



THANK YOU