

# Tastydoc

A documentation tool for dotty using TASTy files

Bryan Abate

19th June 2019

# Table of Contents

Introduction

Features

Architecture

Dottydoc vs Tastydoc

Problems & Further work

# What is Tastydoc ?

- ▶ Documentation tool for Dotty

# What is Tastydoc ?

- ▶ Documentation tool for Dotty
- ▶ Uses TASTy files

# What is Tastydoc ?

- ▶ Documentation tool for Dotty
- ▶ Uses TASTy files
- ▶ Outputs Markdown

# Table of Contents

Introduction

**Features**

Architecture

Dottydoc vs Tastydoc

Problems & Further work

- ▶ Annotations, modifiers (including scope modifiers), parameters, type parameters, and return types

# Accessible information

- ▶ Annotations, modifiers (including scope modifiers), parameters, type parameters, and return types
- ▶ Members, parents, constructors, known subclasses and companion



# Accessible information

- ▶ Annotations, modifiers (including scope modifiers), parameters, type parameters, and return types
- ▶ Members, parents, constructors, known subclasses and companion
- ▶ User documentation (Wiki-style & Markdown)

- ▶ Extract information from them

- ▶ Extract information from them
- ▶ Independent from the compiler

- ▶ To types

# Linking

- ▶ To types
- ▶ Companion

# Linking

- ▶ To types
- ▶ Companion
- ▶ Annotations

# Linking

- ▶ To types
- ▶ Companion
- ▶ Annotations
- ▶ Scope modifiers

# Linking

- ▶ To types
- ▶ Companion
- ▶ Annotations
- ▶ Scope modifiers
- ▶ Parents



- ▶ Easy to edit by hand & preview

# Markdown

- ▶ Easy to edit by hand & preview
- ▶ Easy to add own files

# Markdown

- ▶ Easy to edit by hand & preview
- ▶ Easy to add own files
- ▶ Easy for the user to make links

# Markdown

- ▶ Easy to edit by hand & preview
- ▶ Easy to add own files
- ▶ Easy for the user to make links
- ▶ Git hosting service have built-in preview

# Markdown

- ▶ Easy to edit by hand & preview
- ▶ Easy to add own files
- ▶ Easy for the user to make links
- ▶ Git hosting service have built-in preview
- ▶ Easy to convert to another format (HTML, PDF, etc.)

# Table of Contents

Introduction

Features

Architecture

Dottydoc vs Tastydoc

Problems & Further work

# Representation

- ▶ Contain information about an entity

# Representation

- ▶ Contain information about an entity
- ▶ Easy to use, no knowledge of TASTy required



# Representation

- ▶ Contain information about an entity
- ▶ Easy to use, no knowledge of TASTy required
- ▶ Code is easy to maintain

# Representation

- ▶ Contain information about an entity
- ▶ Easy to use, no knowledge of TASTy required
- ▶ Code is easy to maintain
- ▶ Similar to Dottydoc Entity → can reuse Dottydoc code

- ▶ Contain information about types

# Reference

- ▶ Contain information about types
- ▶ Necessary for linking

# Reference

- ▶ Contain information about types
- ▶ Necessary for linking
- ▶ Inspired by Dottydoc

- ▶ Access to all @ except @usecase and @define

# User documentation

- ▶ Access to all @ except @usecase and @define
- ▶ Support Wiki-style and Markdown

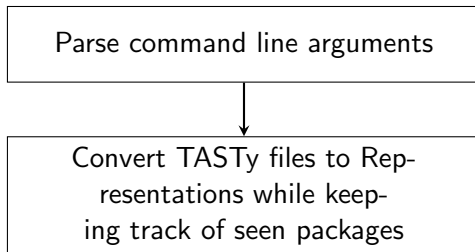
# User documentation

- ▶ Access to all @ except @usecase and @define
- ▶ Support Wiki-style and Markdown
- ▶ Uses Dottydoc code modified for:
  - ▶ Markdown output
  - ▶ Small changes in structure

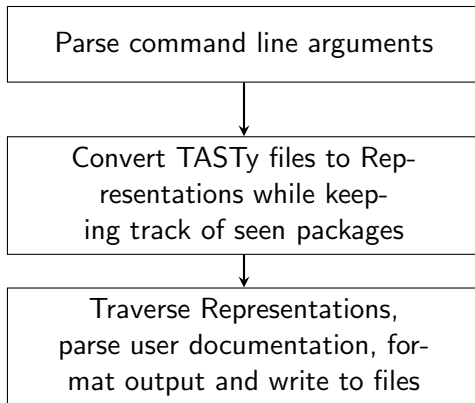


Parse command line arguments

# Workflow



# Workflow



# Table of Contents

Introduction

Features

Architecture

Dottydoc vs Tastydoc

Problems & Further work

# General comparison

- ▶ Compiler internals

# General comparison

- ▶ Compiler internals
- ▶ Markdown vs HTML/CSS

# Extra features

- ▶ Scope modifiers

# Extra features

- ▶ Scope modifiers
- ▶ Known subclasses



# Extra features

- ▶ Scope modifiers
- ▶ Known subclasses
- ▶ Refined types

# Bugs fixed

## ► Buggy output

```
final val BITS_PER_LAZY_VAL : [31m2L[0m
```

# Bugs fixed

- ▶ Buggy output

```
final val BITS_PER_LAZY_VAL : [31m2L[0m
```

- ▶ Wrong parents

# Bugs fixed

- ▶ Buggy output

```
final val BITS_PER_LAZY_VAL : [31m2L[0m
```

- ▶ Wrong parents
- ▶ Annotations

# Bugs fixed

- ▶ Buggy output

```
final val BITS_PER_LAZY_VAL : [31m2L[0m
```

- ▶ Wrong parents
- ▶ Annotations
- ▶ Compiler artifacts

# Bugs fixed

- ▶ Buggy output

```
final val BITS_PER_LAZY_VAL : [31m2L[0m
```

- ▶ Wrong parents
- ▶ Annotations
- ▶ Compiler artifacts
- ▶ potentially program breaking code

```
def parents: List[Entity] = this :: this.parents
```

# Table of Contents

Introduction

Features

Architecture

Dottydoc vs Tastydoc

Problems & Further work

# Problems

- ▶ Markdown escaping



# Problems

- ▶ Markdown escaping
- ▶ Linking inside code blocks

# Problems

- ▶ Markdown escaping
- ▶ Linking inside code blocks
- ▶ Section

# Problems

- ▶ Markdown escaping
- ▶ Linking inside code blocks
- ▶ Section
- ▶ IDs for linking

# Further work

- ▶ Markdown escaping

## Further work

- ▶ Markdown escaping
- ▶ Type lambdas

## Further work

- ▶ Markdown escaping
- ▶ Type lambdas
- ▶ Complex types

```
class Graph {  
    type Node = Int  
}  
def linkingGraph(g: Graph): g.Node = ???
```

## Further work

- ▶ Markdown escaping
- ▶ Type lambdas
- ▶ Complex types

```
class Graph {  
    type Node = Int  
}  
def linkingGraph(g: Graph): g.Node = ???
```

- ▶ Default values

# Further work

- ▶ Markdown escaping
- ▶ Type lambdas
- ▶ Complex types

```
class Graph {  
    type Node = Int  
}  
def linkingGraph(g: Graph): g.Node = ???
```

- ▶ Default values
- ▶ Extra user-documentation parsing



# Further work

- ▶ Markdown escaping
- ▶ Type lambdas
- ▶ Complex types

```
class Graph {  
    type Node = Int  
}  
def linkingGraph(g: Graph): g.Node = ???
```

- ▶ Default values
- ▶ Extra user-documentation parsing
- ▶ HTML/CSS

# Questions ?

?