

jazzy

github.com/realm/jazzy

a soulful way to generate docs for Swift & Objective-C



\$ jazzy

Soulful docs for Swift & Objective-C

**What's a
documentation
generator?**

Instead of parsing your source files, jazzy hooks into clang/SourceKit and uses the AST representation of your code and its comments for more accurate results.

```
1 //
2 // JAZMusician.h
3 // JazzyApp
4 //
5
6 #import <Foundation/Foundation.h>
7
8 /**
9  JAZMusician models, you guessed it... Jazz Musicians!
10  From Ellington to Marsalis, this class has you covered.
11  */
12 @interface JAZMusician : NSObject
13
14 /**
15  The name of the musician. i.e. "John Coltrane"
16  */
17 @property (nonatomic, readonly) NSString *name;
18
19 /**
20  The year the musician was born. i.e. 1926
21  */
22 @property (nonatomic, readonly) NSUInteger birthyear;
23
24 /**
25  Initialize a JAZMusician.
26  Don't forget to have a name and a birthyear.
27
28  @warning Jazz can be addicting.
29  Please be careful out there.
30
31  @param name The name of the musician.
32  @param birthyear The year the musician was born.
33
34  @return An initialized JAZMusician instance.
35  */
36 - (instancetype)initWithName:(NSString *)name birthyear:(NSUInteger)birthyear;
37
38 @end
39
```

Language: [Swift](#) [Obj-C](#) [Both](#)

JAZMusician

JAZMusician models, you guessed it... Jazz Musicians! From Ellington to Marsalis, this class has you covered.

Inheritance

↳ JAZMusician

Conforms To

[NSObject](#)

Import Statement

@import Foundation;

Methods

- initWithName:birthyear:

Initialize a JAZMusician. Don't forget to have a name and a birthyear.

Declaration

SWIFT

```
init(name name: String!,
      birthyear birthyear: Int)
```

OBJECTIVE-C

```
- (instancetype)initWithName:(NSString *)name birthyear:(NSUInteger)birthyear;
```

Parameters

<i>name</i>	The name of the musician.

Classes

- Manager
- Request

Global Variables

- AlamofireErrorDomain

Enums

- Method
- ParameterEncoding

Extensions

- String
- NSURL
- NSURLComponents
- NSURLRequest
- NSURLRequest
- Request
- Manager
- Request
- Manager

```
deinit

request(_:_:parameters:encoding:)
```

Creates a request for the specified method, URL string, parameters, and parameter encoding.

Declaration

```
SWIFT

public func request(method: Method, _ urlString: URLStringConvertible, parameters: [String :
AnyObject]? = default, encoding: ParameterEncoding = default) -> Request
```

Parameters

<i>method</i>	The HTTP method.
<i>URLString</i>	The URL string.
<i>parameters</i>	The parameters. <code>nil</code> by default.
<i>encoding</i>	The parameter encoding. <code>.URL</code> by default.

Return Value

The created request.

[Show on GitHub](#)

```
request(_:)

SessionDelegate
```

How do you *USE* it?

First off, you have to write documentation comments. Which is beneficial not only for jazzy, but for option-clicking in Xcode

Doxygen (ObjC-style)

```
/**
```

```
    Lorem ipsum dolor sit amet.
```

```
    @param bar Consectetur adipisicing elit.
```

```
    @return Sed do eiusmod tempor.
```

```
*/
```

```
func foo(bar: String) -> AnyObject { ... }
```

Restructured Text (ReST)

```
/**
```

```
    Lorem ipsum dolor sit amet.
```

```
    :param: bar Consectetur adipisicing elit.
```

```
    :returns: Sed do eiusmod tempor.
```

```
*/
```

```
func foo(bar: String) -> AnyObject { ... }
```

```
$ [sudo] gem install jazzy
```

\$

jazzzy

\$ jazzy

- Grabs all the *.swift files from the specified target and builds docs for them
- Target & destination are configurable...
- Outputs static site in docs/ by default

Design goals

- Generate source code docs matching Apple's official reference documentation
- Support for Xcode and Dash docsets
- High readability of source code comments
- Leverage modern HTML templating (Mustache)
- Let Clang/SourceKit do all the heavy lifting
- Compatibility with appledoc when possible

\$ jazzy help

-o, --output FOLDER	Folder to output the HTML docs to
-c, --[no-]clean	Delete contents of output directory before running. WARNING: If --output is set to ~/Desktop, this will delete the ~/Desktop directory.
-x arg1,arg2,...argN,	Arguments to forward to xcodebuild --xcodebuild-arguments
-a, --author AUTHOR_NAME	Name of author to attribute in docs (i.e. Realm)
-u, --author_url URL	Author URL of this project (i.e. http://realm.io)
-m, --module MODULE_NAME	Name of module being documented. (i.e. RealmSwift)
-d, --dash_url URL	Location of the dash XML feed (i.e. http://realm.io/docsets/realm.xml)
-g, --github_url URL	GitHub URL of this project (i.e. https://github.com/realm/realm-cocoa)
--github-file-prefix PREFIX	GitHub URL file prefix of this project (i.e. https://github.com/realm/realm-cocoa/tree/v0.87.1)
-s FILEPATH,	XML doc file generated from sourcekitten to parse
-r, --root-url URL	Absolute URL root where these docs will be stored
--module-version VERSION	module version. will be used when generating docset
--min-acl [private internal public]	minimum access control level to document (default is internal)
--[no-]skip-undocumented	Don't document declarations that have no documentation comments.
--podspec FILEPATH	
--docset-icon FILEPATH	
--docset-path DIRPATH	The relative path for the generated docset
--source-directory DIRPATH	The directory that contains the source to be documented
-v, --version	Print version number
-h, --help	Print this help message

**A few feature
highlights**

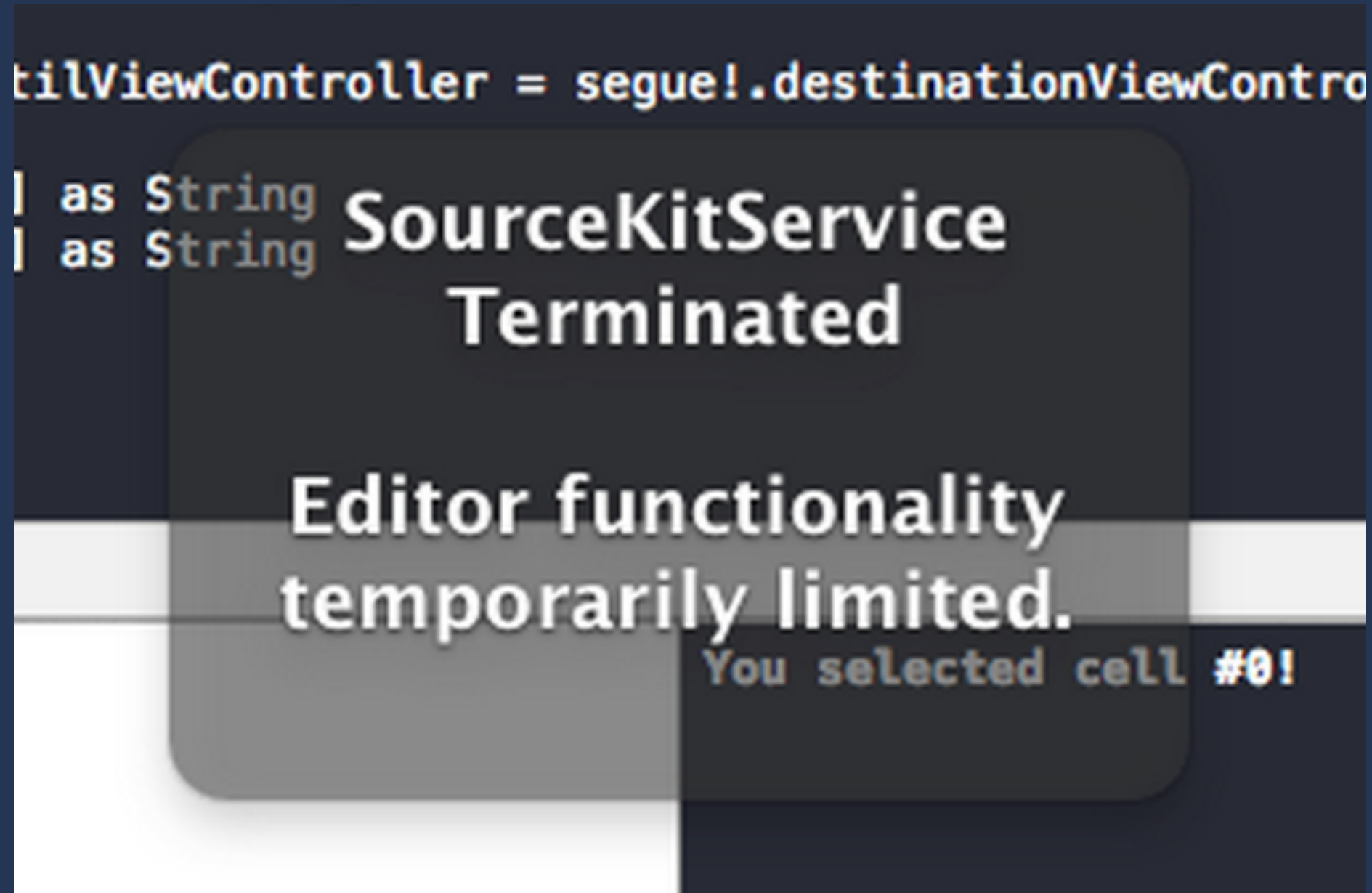
Getting there

0	E29CA807	01080000	22000000	0701382C	40ADF00A	A4500BAC	500ABF40	0AAF300A	BF200AAC
36	F00AA3B0	0A0741B2	D08CC22B	B8022DC8	C22BB0C2	2F88022B	BCC228AC	C24312B4	4229D042
72	28904228	D042283C	30422BBC	0229D402	2B94C22F	B84228B4	42291C04	E1B0D08C	C22BB442
108	2B94822B	D0C22F88	022BBCC2	28ACC283	13904229	8C022BFC	C228BC42	2BB44229	B8022DCC
144	02000000	81090000	B4000000	250C0000	0C000000	12055464	80C02030	28150000	53776966
180	74467261	6D330000	0500F003	58000000	4C4C564D	20332E35	73766E00	00000000	85110000
216	A3000000	3206100C	4A7401FC	04000000	00000000	010045EA	7C741600	00003F00	0000733A
252	43313053	77696674	4672616D	33364243	6C617373	15000000	54686973	20697320	74686520
288	27422720	636C6173	73010000	00010000	001A0000	002F2F2F	20546869	73206973	20746865
324	20274227	20636C61	73730A01	00511619	182C0000	006E0000	00733A46	43313053	77696674
360	4672616D	33364243	6C617373	63464D53	305F4654	36737472	696E6753	535F5330	5F160000
396	00422773	20737472	696E6720	696E6974	69616C69	7A657201	00000005	00000048	0000002F
432	2A2A0A20	20202042	27732073	7472696E	6720696E	69746961	6C697A65	720A2020	20200A20
468	2020203A	70617261	6D3A2073	7472696E	67205468	65207374	72696E67	0A202020	202A2F01
504	00A4C127	72160000	00410000	00733A43	31305377	69667446	72616D33	3641436C	61737315
540	00000054	68697320	69732074	68652027	41272063	6C617373	01000000	01000000	1C000000
576	2F2A2A0A	54686973	20697320	74686520	27412720	636C6173	730A2A2F	40000000	03000000
612	00000000	00000000	00000000	00000000	00000000	04000000	00000000	00000000	00000000
648	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	67000000
684	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
720	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
756	0F010000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
792	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
828	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
864	00000000	00000000	00000000						

```
,ú®      "      8,@≠  §P  `P  ø@  ø0  ø  ``
  f∞   A≤-å¬+Π  -»¬+∞¬/à  +°¬(`¬C  ¥B)-B
(êB(-B(<0B+°  )‘  +î¬/ΠB(¥B)  ·∞-å¬+¥B
+îç+-¬/à  +°¬(`¬É  êB)å  +,¬(°B+¥B)Π  -Ã
      Å      ¥      %      TdÄ¿  0(      Swif
tFram3      •  X      LLVM 3.5svn      ö
f      2      Jt      ,      EÍlt      ?      s:
C10SwiftFram36BClass      This is the
'B' class      /// This is the
'B' class      Q      ,      n      s:FC10Swift
Fram36BClasscFMS0_FT6stringSS_S0_
B's string initializer      H      /
**      B's string initializer
      :param: string The string      */
§i'r      A      s:C10SwiftFram36AClass
      This is the 'A' class
/** This is the 'A' class */@

g
```

SourceKit



SourceKitten

github.com/jpsim/SourceKitten

An adorable little framework and
command line tool for interacting with
SourceKit

SourceKitten

1. Swift code highlighting
2. Swift interface generation from ObjC headers
3. Generate Swift docs from Swift source
4. General purpose AST manipulation
5. USR generation

SourceKitten

```
$ sourcekitten
```

Available commands:

doc	Print Swift docs as JSON or Objective-C docs as XML
help	Display general or command-specific help
structure	Print Swift structure information as JSON
syntax	Print Swift syntax information as JSON
version	Display the current version of SourceKitten

SourceKitten Info

1. Written in Swift
2. Split in two parts: SourceKittenFramework + sourcekitten
3. Uses XPC connection to sourcekitd.framework, just like Xcode
4. Installable via Homebrew (`brew install sourcekitten`)
5. Similar architecture to Carthage, Commandant, package installer

You could do a *lot more* with SourceKitten

1. Swift editor/IDE
2. Swift linter
3. Test generator from doc comments
4. Swift mocks
5. Static analysis
6. Holistic way to diff Swift syntax versions

What's next for jazzy?

1. Objective-C
2. Mixed projects
3. Cross-language (calling ObjC from Swift, calling Swift from ObjC)
4. Cross-project hyperlinking on Coccadocs

Resources

- jazzy: github.com/realm/jazzy
- SourceKitten: github.com/jpsim/SourceKitten
- appledoc: github.com/tomaz/appledoc
- swift doc syntax: nshipster.com/swift-documentation
- SourceKit: jpsim.com/uncovering-sourcekit

SwiftLDN().questions?.askThem!

JP Simard, @simjp,
realm.io