

markdown和appledoc 项目文档编辑

李智慧

一、Macdown常用语法和简介

二、使用Doxygen、HeaderDoc、
appledoc生成项目API文档

Markdown



Markdown是一种 **轻量级的标记语言**，可以使用普通文本编辑器编写的语言。通过简单的标记语法，使普通的文本具有一定的格式

Markdown 的优点在于，它用简洁的语法代替了排版。而不像一般我们用的处理软件 Word 或 Pages 有大量的排版、字体设置。它使我们专注于内容的编写。

Markdown 不仅支持文字排版，还支持 插入 图片， 连接， 视频， 音频 等。而这些操作我们都可以只用键盘完成。

MacDown 即可以把 内容导出为 HTML 文件、PDF 文件、.md文件还可以实时预览。

MacDown

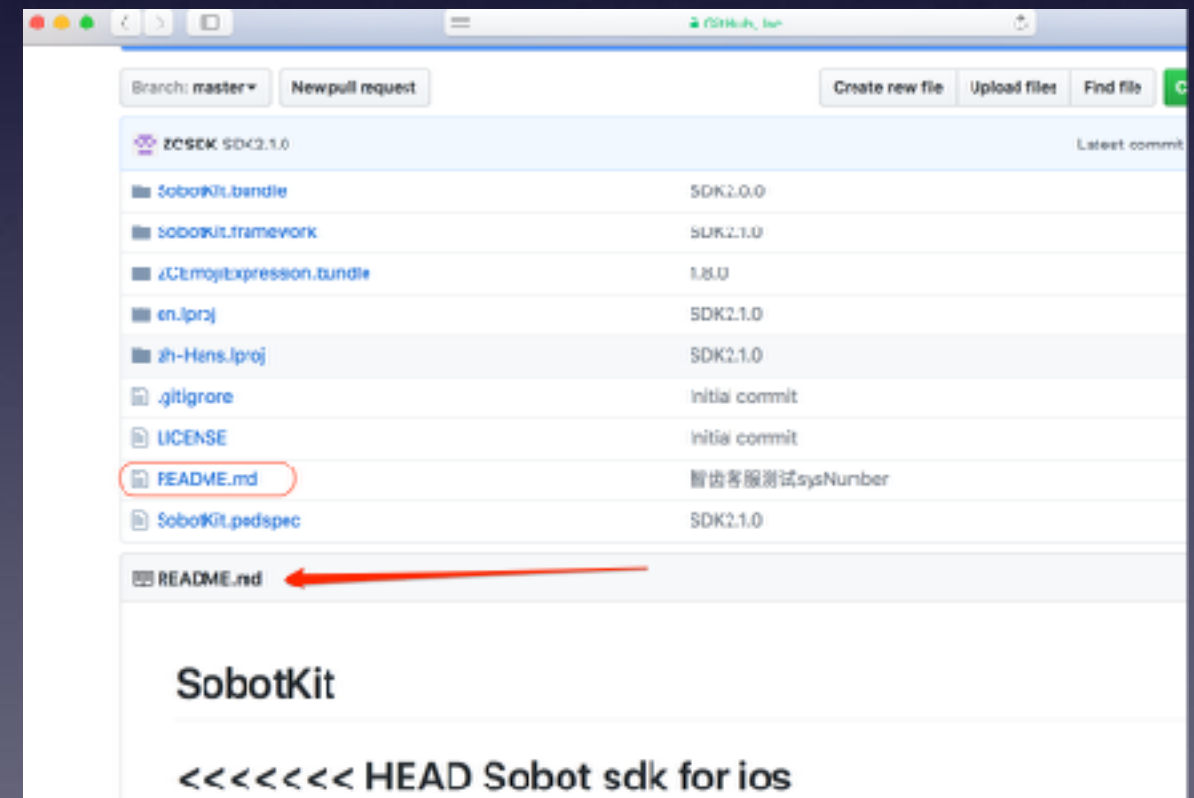


使用场景

项目的说明文档、技术开发文档

现在多个 论坛或者博客都支持 [markdown](#) 语法，例如：CSDN、简书等。

GitHub的readme.md文件



常用的Markdown 编辑器

- OSX

Atom
Byword
Mou
Typora
MacDown
RStudio

- Windows

Atom
MarkdownPad
Miu
Typora
RStudio

- 浏览器插件

MaDo (Chrome)
Marxico (Chrome)

- Linux

Atom
ReText
UberWriter
RStudio

- 高级应用

Sublime Text 2 + MarkdownEditing

- iOS

Byword

##一、集成准备 <div id="一"></div>

###1.1注册AppKey<div id="1.1"></div>

请先确保已注册智齿客服账号https://www.sobot.com。

###1.2下载智齿SDK<div id="1.2"></div>

点击下载最新SDK包。

[iOS_SDK](https://github.com/ZCSDK/SobotKit)

###1.3包文件说明 <div id="1.3"></div>

解压下载【iOS_SDK】，下载完毕为一个压缩包，包含（SobotKit.framework、SobotKit.bundle和ZCEmojiExpression.bundle）、SobotDemo和相关说明文档。

文件名	说明
SobotKit.framework	智齿SDK核心代码库
SobotKit.bundle	图片资源文件
ZCEmojiExpression.bundle	表情资源文件
ZCLibInitInfo.h	实体类：基础参数类（自定义用户配置信息）
ZCKitInfo.h	实体类：基础参数（UI自定义）
ZCSobot.h	连接客户端到智齿客服，该文件提供接入功能
SobotKit.h	该文件引用所有SDK需要的文件

一、集成准备

1.1注册AppKey

请先确保已注册智齿客服账号<https://www.sobot.com>。

1.2下载智齿SDK

点击下载最新SDK包。 [iOS_SDK](#)

1.3包文件说明

解压下载【iOS_SDK】，下载完毕为一个压缩包，包含（SobotKit.framework、SobotKit.bundle和ZCEmojiExpression.bundle）、SobotDemo和相关说明文档。

文件名	说明
SobotKit.framework	智齿SDK核心代码库
SobotKit.bundle	图片资源文件
ZCEmojiExpression.bundle	表情资源文件
ZCLibInitInfo.h	实体类：基础参数类（自定义用户配置信息）
ZCKitInfo.h	实体类：基础参数（UI自定义）
ZCSobot.h	连接客户端到智齿客服，该文件提供接入功能
SobotKit.h	该文件引用所有SDK需要的文件

标题

Markdown 支持两种标题的语法，类 `Setext` 和类 `atx` 形式

类 `Atx` 形式则是在行首插入 1 到 6 个 `#`，对应到标题 1 到 6 阶

`#`一级标题

`##`二级标题

`###`三级标题

`####`四级标题

`#####`五级标题

`#####`六级标题

`#####`7级没有效果了

一级标题

二级标题

三级标题

四级标题

五级标题

六级标题

#7级没有效果了

可以选择性地「闭合」类 atx 样式的标题，这纯粹只是美观用的，若是觉得这样看起来比较舒适，你就可以在行尾加上 #，而行尾的 # 数量也不用和开头一样（行首的井字符数量决定标题的阶数）

这是 H1

这是 H2

这是 H3

这是 H1

这是 H2

这是 H3

Setext 形式是用底线的形式，利用 **=**（最高阶标题）和 **-**（第二阶标题）

This is an H1

=====

This is an H2

This is an H1

This is an H2

字体颜色

```
<font face="黑体">我是黑体字</font>
<font face="微软雅黑">我是微软雅黑</font>
<font face="STCAIYUN">我是华文彩云</
font>
<font color=#0099ff size=7 face="黑
体">color=#0099ff size=72 face="黑体"</
font>
<font color=#00ffff size=72>color=#00ffff</
font>
<font color=gray size=72>color=gray</font>
```

我是黑体字 我是微软雅黑 我是华文彩云

color=#0099ff

size=72 face="黑体"

color=#00ffff

color=gray

粗体、斜体

斜体

斜体

****粗体****

粗体

斜体

粗体

斜体

斜体

粗体

粗体

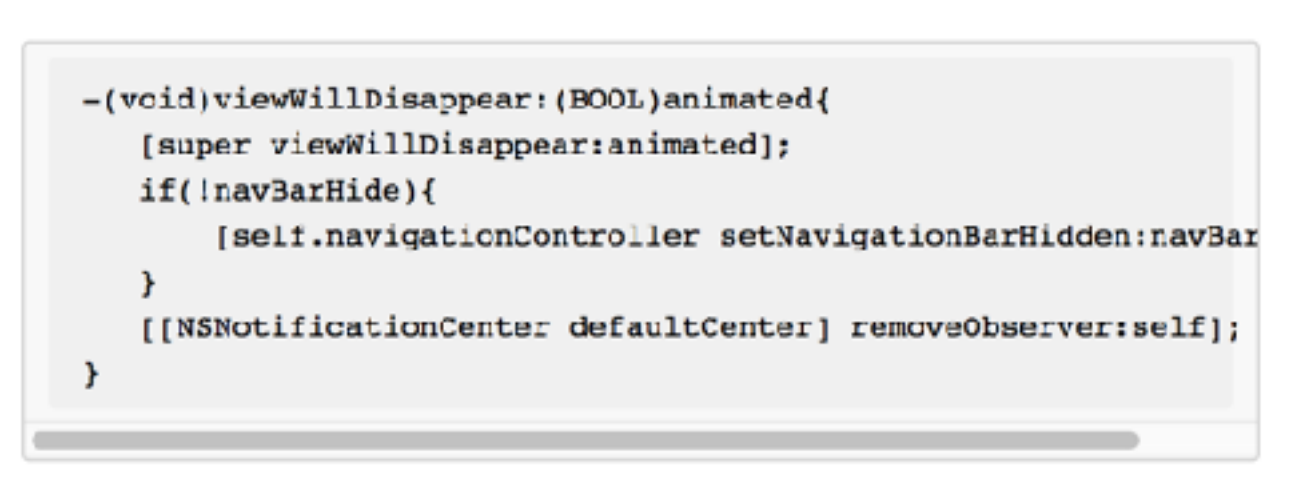
斜体

粗体

插入代码

和程序相关的写作或是标签语言原始码通常会有已经排版好的代码区块，通常这些区块我们并不希望它以一般段落文件的方式去排版，而是照原来的样子显示，Markdown 会用 `<pre>` 和 `<code>` 标签来把代码区块包起来

```
...  
-(void)viewWillDisappear:(BOOL)animated{  
    [super viewWillDisappear:animated];  
    if(!navBarHide){  
        [self.navigationController  
setNavigationBarHidden:navBarHide];  
    }  
    [[NSNotificationCenter defaultCenter]  
removeObserver:self];  
}  
...
```



```
-(void)viewWillDisappear:(BOOL)animated{  
    [super viewWillDisappear:animated];  
    if(!navBarHide){  
        [self.navigationController setNavigationBarHidden:navBar  
    }  
    [[NSNotificationCenter defaultCenter] removeObserver:self];  
}
```

... 或者 `<code></code>`

如果要标记一小段行内代码，你可以用反引号把它包起来 (```)

插入图片

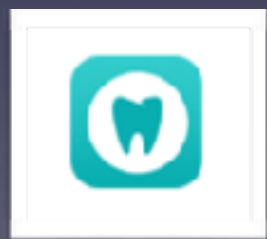
Markdown 使用一种和链接很相似的语法来标记图片，同样也允许两种样式：
行内式和参考式

`![Alt text](/path/to/img.jpg)`

`![Alt text](/path/to/img.jpg "Optional title")`

`![Alt text][id]`

`![图片](Icon_SDK.png)`



- 一个感叹号 `!`
- 接着一个方括号，里面放上图片的替代文字
- 接着一个普通括号，里面放上图片的网址，最后还可以用引号包住并加上 选择性的 'title' 文字

`![图片](http://www.sobot.com)`



到目前为止，**Markdown** 还没有办法指定图片的宽高，如果需要的话，可以使用普通的 `` 标签

插入语音

```
<audio controls>
```

```
<source src="http://demo.mimvp.com/html5/take_you_fly.mp3" />
```

```
</audio>
```



插入视频

```
<video id="video" controls="" preload="none" poster="http://media.w3.org/2010/05/sintel/poster.png">  
  <source id="mp4" src="http://media.w3.org/2010/05/sintel/trailer.mp4" type="video/mp4">  
  <source id="webm" src="http://media.w3.org/2010/05/sintel/trailer.webm" type="video/webm">  
  <source id="ogv" src="http://media.w3.org/2010/05/sintel/trailer.ogv" type="video/ogg">  
</video>
```



插入链接

Markdown 支持以比较简短的自动链接形式来处理网址和电子邮件信箱，只要是用方括号包起来，**Markdown** 就会自动把它转成链接。一般网址的链接文字就和链接地址一样

This is [an example](http://example.com/ "Title") inline link.

[This link](http://example.net/) has no title attribute.

```
<p>This is <a href="http://example.com/" title="Title">
an example</a> inline link.</p>
```

```
<p><a href="http://example.net/">This link</a> has no
title attribute.</p>
```

[iOS_SDK](https://github.com/ZCSDK/SobotKit)

```
<http://example.com/>
```

Markdown 会转为：

```
<a href="http://example.com/">http://example.com/</a>
```

This is [an example](http://example.com/) inline link.

[This link](http://example.net/) has no title attribute.

This is [an example](https://github.com/ZCSDK/SobotKit) inline link.

[This link](https://github.com/ZCSDK/SobotKit) has no title attribute.

[iOS_SDK](https://github.com/ZCSDK/SobotKit)

<http://example.com/>

Markdown 支持有序列表和无序列表

有序列表则使用数字接着一个英文句点

1. 第一点
2. 第二点
4. 第三点

1. 第一点
2. 第二点
3. 第三点

```
<ol>  
<li>Bird</li>  
<li>McHale</li>  
<li>Parish</li>  
</ol>
```

1. Bird
2. McHale
3. Parish

无序列表

无序列表使用星号、加号或是减号作为列表标记

```
+ 1
  * 1-1
  - 1-2
  - 1-3
    - 1-1-1
    + 1-1-2
* 2
```

```
• 1
  ◦ 1-1
  ◦ 1-2
  ◦ 1-3
    ▪ 1-1-1
    ▪ 1-1-2
• 2
```

区块引用

Markdown 标记区块引用是使用类似 email 中用 `>` 的引用方式。如果你还熟悉在 email 信件中的引言部分，你就知道怎么在 Markdown 文件中建立一个区块引用，那会看起来像是你自己先断好行，然后在每行的最前面加上 `>`

区块引用可以嵌套（例如：引用内的引用），只要根据层次加上不同数量的 `>`。引用的区块内也可以使用其他的 Markdown 语法，包括标题、列表、代码区块等

Block Quote

```
> Angle brackets `>` are used for block quotes.  
Technically not every line needs to start with a `>` as long as  
there are no empty lines between paragraphs.  
> Looks kinda ugly though.  
> > Block quotes can be nested.  
> > > Multiple Levels  
>  
> Most markdown syntaxes work inside block quotes.  
>  
> * Lists  
> * [Links][arbitrary_id]  
> * Etc.
```

Block Quote

Angle brackets `>` are used for block quotes.
Technically not every line needs to start with a `>` as long as there are no empty
lines between paragraphs.
Looks kinda ugly though.

Block quotes can be nested.

Multiple Levels

Most markdown syntaxes work inside block quotes.

- Lists
- [Links](#)
- Etc.

表格

This is a table:

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

This is a table:

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

Left Aligned	Center Aligned	Right Aligned
col 3 is	some wordy text	\$1600
col 2 is	centered	\$12
zebra stripes	are neat	\$1

Left Aligned	Center Aligned	Right Aligned
col 3 is	some wordy text	\$1600
col 2 is	centered	\$12
zebra stripes	are neat	\$1

- : 为右对齐。
- :----- 为左对齐。
- :-----: 为居中对齐。
- 为使用默认居中对齐。

```
<table>

<tr>
<td><center> 属性名称 </center></td>
<td><center> 数据类型 </center></td>
<td><center> 说明 </center></td>
</tr>

<tr>
<td><center> thumbUrl </center></td>
<td><center> String </center></td>
<td><center>商品图片链接地址</center></td>
</tr>

<tr>
<td><center> title </center></td>
<td><center> String </center></td>
<td><center>商品标题</center></td>
</tr>

<tr>
<td><center> desc </center></td>
<td><center> String </center></td>
<td><center>商品摘要， 商品的描述</center></td>
</tr>

<tr>
<td><center> label </center></td>
<td><center> String </center></td>
<td><center>商品标签， 例如价格</center></td>
</tr>

<tr>
<td><center> link </center></td>
<td><center> String </center></td>
<td><center>发送链接， 发送给客服的内容</center></td>
</tr>

</table>
```

属性名称	数据类型	说明
thumbUrl	String	商品图片链接地址
title	String	商品标题
desc	String	商品摘要， 商品的描述
label	String	商品标签， 例如价格
link	String	发送链接， 发送给客服的内容

数学公式

<script type="text/javascript" src="http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=default"></script>

****行间公式****

$$A^T S = B$$

****行内公式****

$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

$A_1 = 5$

$\frac{a}{b}$

$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

行间公式

$$A_S^T = B$$

行内公式

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$A_1 = 5$$

$$\frac{a}{b}$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
<math display="block">
  <msubsup><mi>A</mi> <mi>S</mi> <mi></mi></msubsup>
  <mo>=</mo>
  <mi>5</mi>
</math>
```

$$A_S = 5$$

锚

[1.1 注册AppKey](#1.1)

1.1注册AppKey<div id="1.1"></div>

分割线

在一行中用三个以上的星号、减号、底线来建立一个分隔线，行内不能有其他东西。你也可以在星号或是减号中间插入空格。下面每种写法都可以建立分隔线

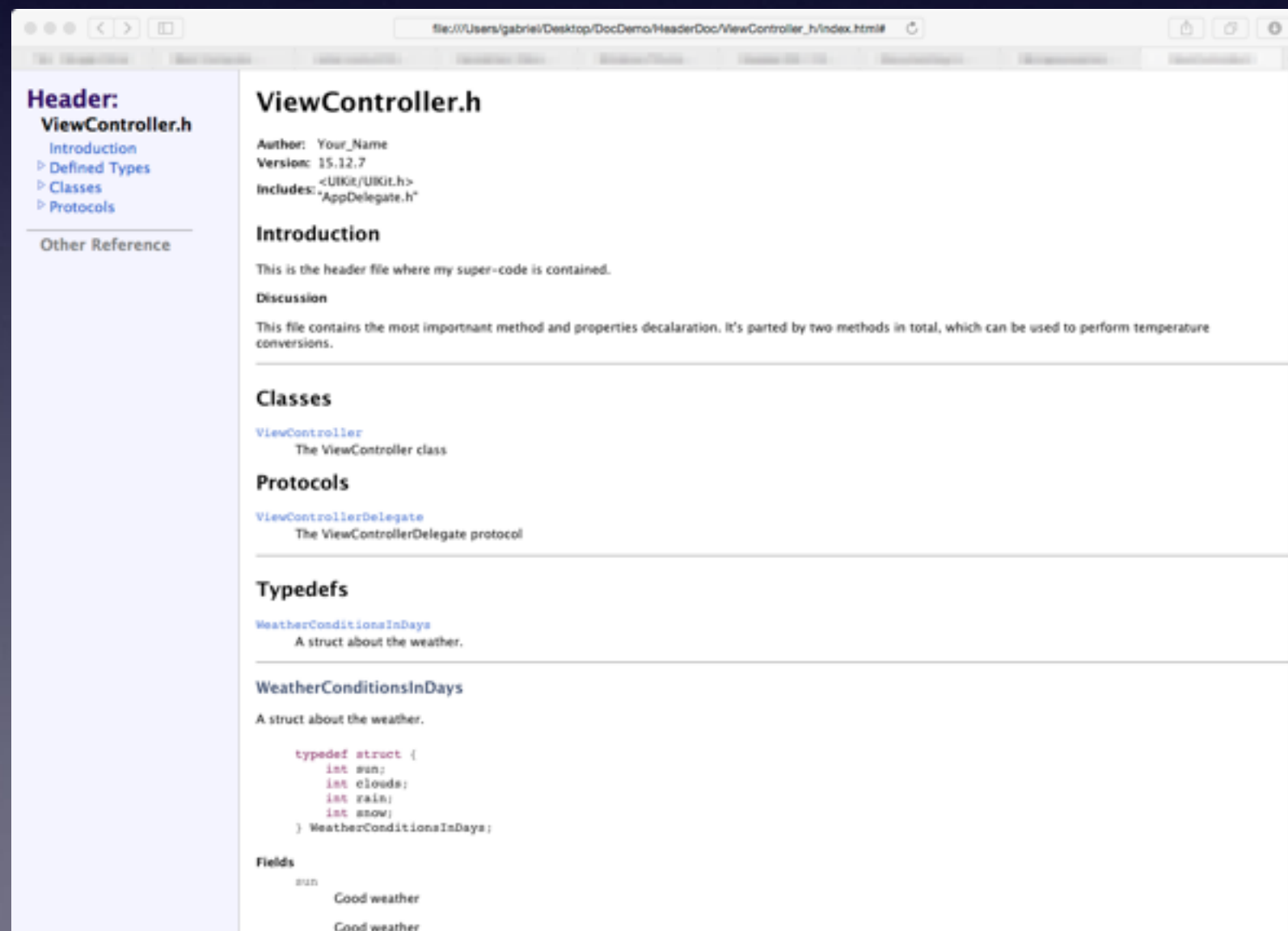
* * *

- - -

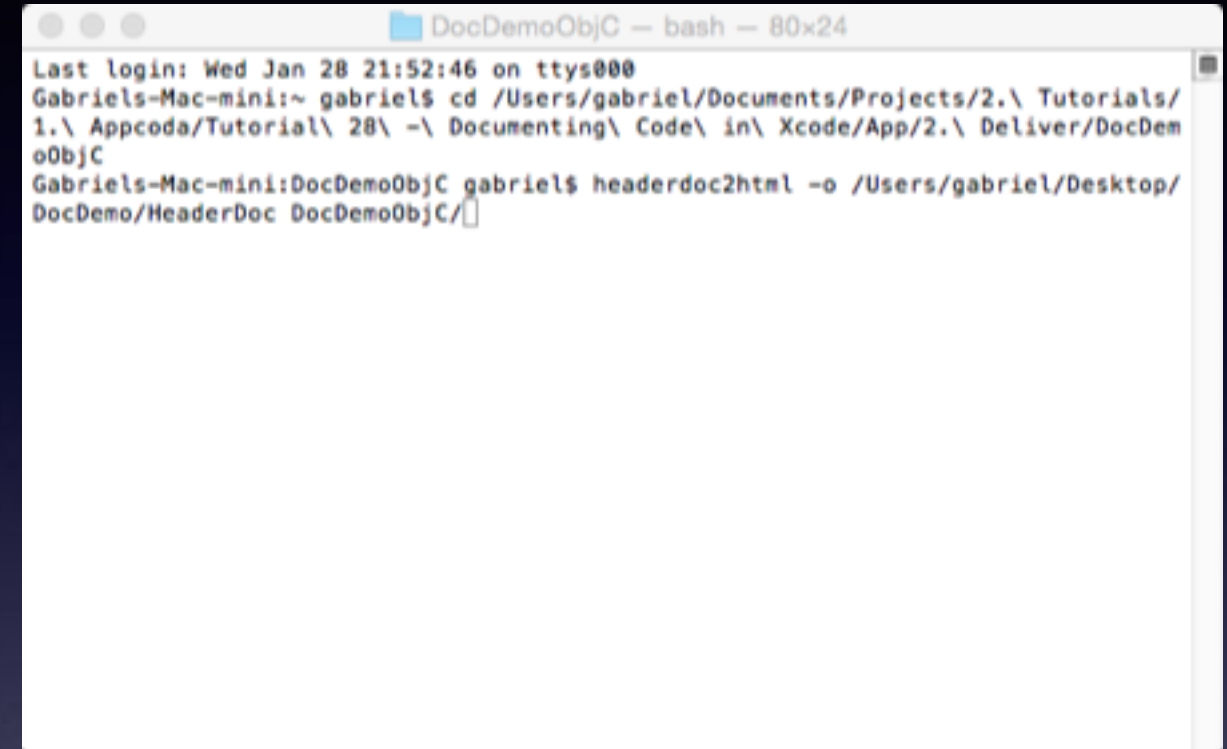
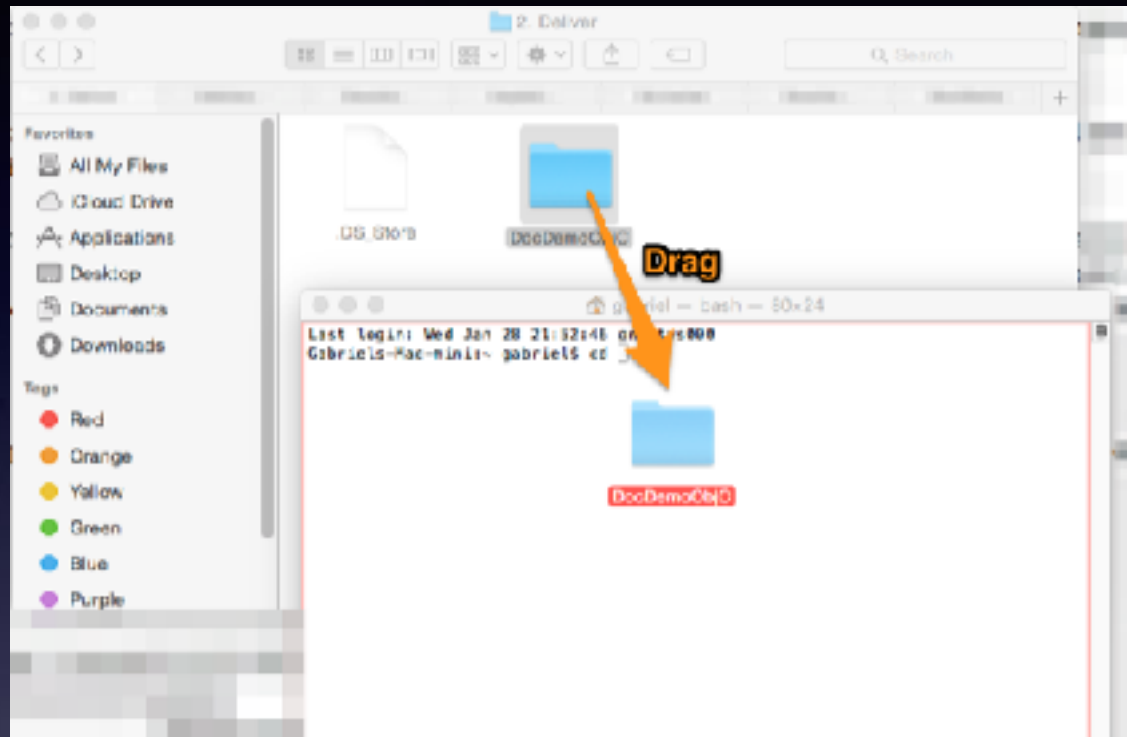


HeaderDoc

" HeaderDoc是一种从C或C++头文件的注释中生成HTML的引用文档的工具。它是用Perl编写的以便于移植。与JavaDoc 相似，它允许开发者容易地文档化他们的接口，并且将接口信息输出到HTML。"



HeaderDoc使用

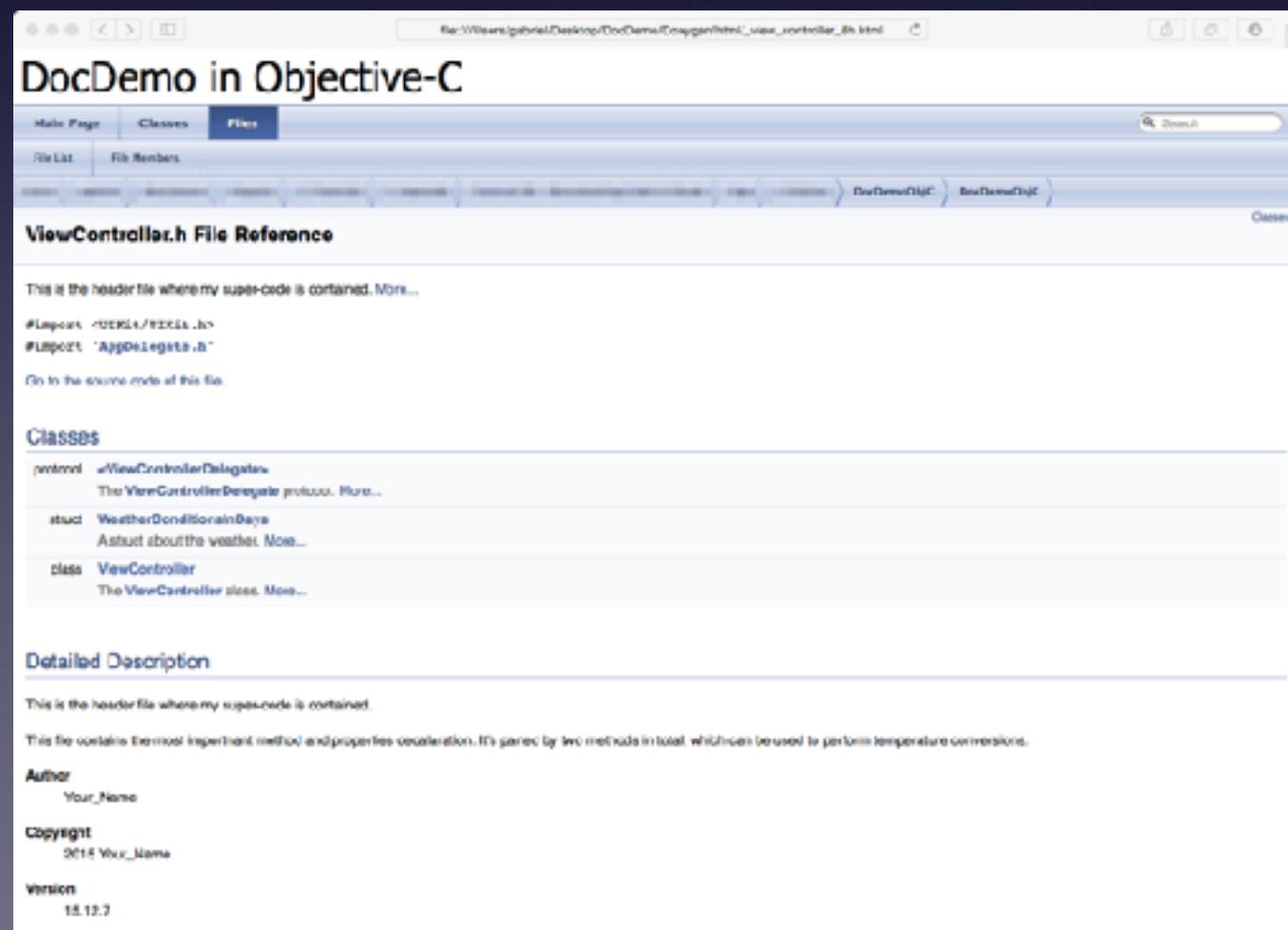


1.headerdoc2html -o /Users/gabriel/Desktop/DocDemo/HeaderDoc DocDemoObjC/

2.gatherheaderdoc

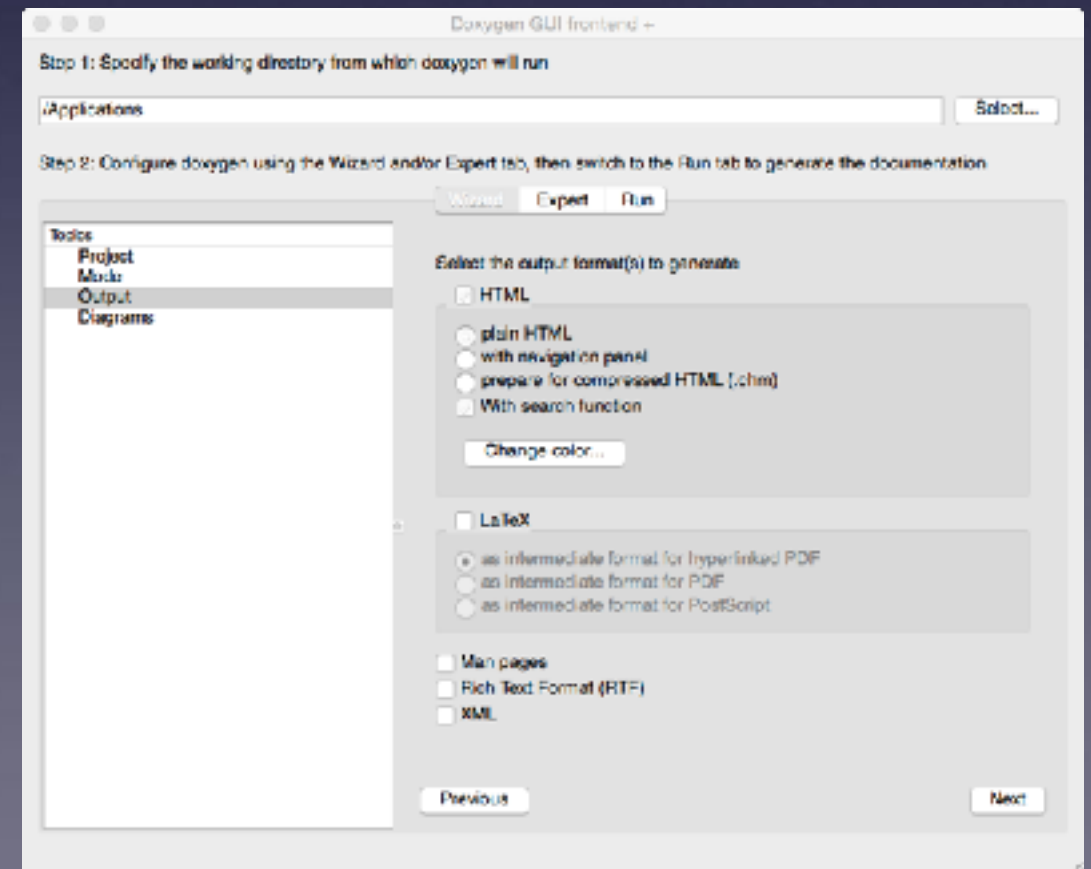
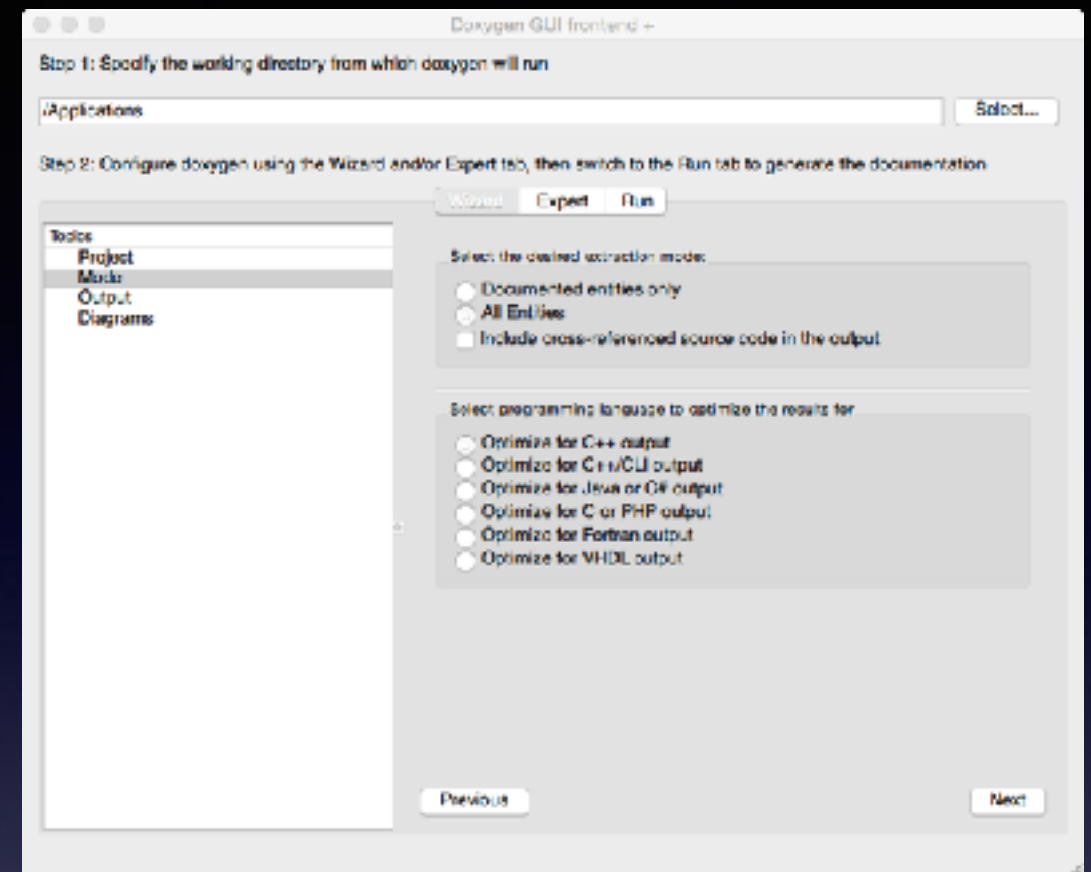
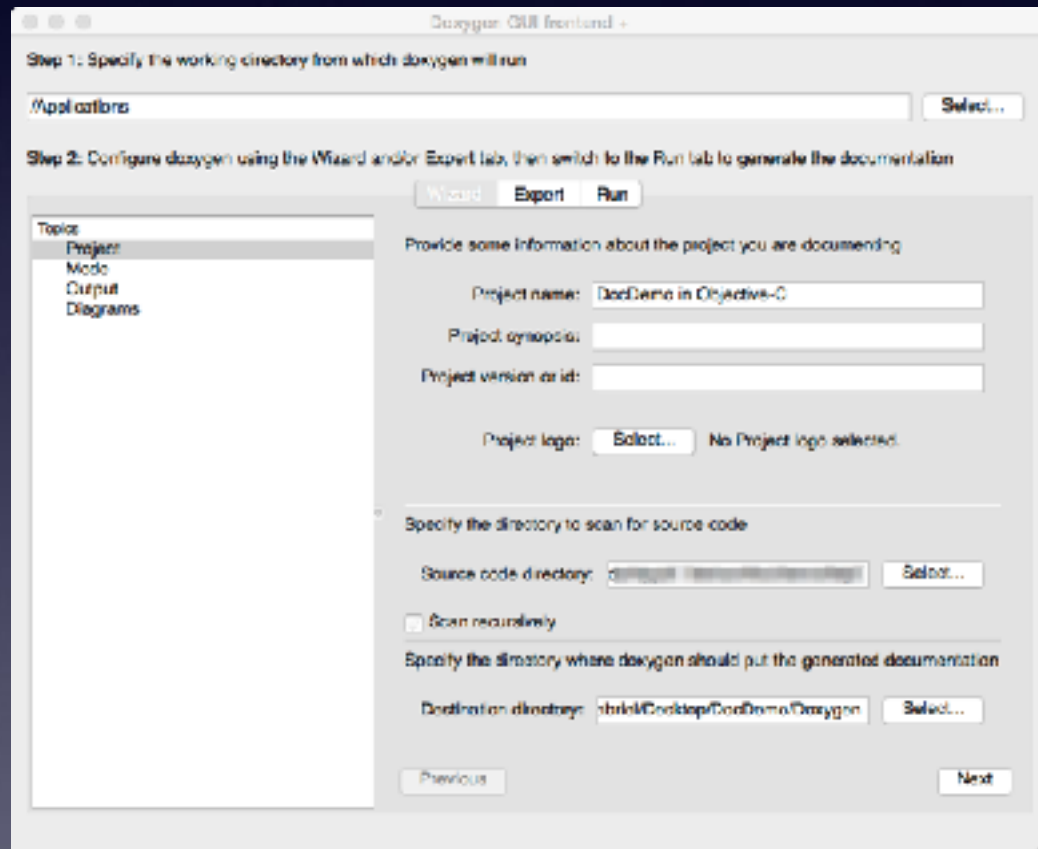
Doxygen

Doxygen是一种开源跨平台的，已类似JavaDoc风格描述的文档系统，完全支持C、C++、Java、Objective-C和IDL语言，部分支持PHP、C#。注释语法与Qt-Doc、KDoc和JavaDoc兼容。Doxygen可以从一套归档源文件开始，生成HTML格式的在线类浏览器，或离线的LATEX、RTF参考手册



Doxygen使用

Doxygen



appledoc

appledoc只为Objective服务，能生成和Apple一个风格的文档，功能齐全，使用方便，还可以直接编译成docset安装进Xcode



SobotKit com.sobot

Hierarchy

SobotKit Reference

Class References

- ZCActionSheet
- ZCChatBaseCell
- ZCGoodsCell
- ZCImageChatCell
- ZCKitInfo
- ZCLibClient
- ZCLibInitInfo
- ZCMLEmojiLabel
- ZCMLEmojiLabelRegexPlistManager
- ZCProductInfo
- ZCRichTextChatCell
- ZCSatisfactionCell
- ZCSobot
- ZCSobotCore
- ZCStoreConfiguration
- ZCTTTAttributedLabel
- ZCTTTAttributedLabelLink
- ZCTipsChatCell
- ZCUIBaseController
- ZCUIChatController
- ZCUIChatKeyboard
- ZCUIChatShare

Protocol References

- RatingViewDelegate
- ZCActionSheetDelegate
- ZCChatCellDelegate
- ZCReceivedMessageDelegate
- ZCTTTAttributedLabel
- ZCTTTAttributedLabelDelegate
- ZCUIBackActionSheetDelegate
- ZCUIKeyboardDelegate
- ZCUIManagerDelegate
- ZCUIRecordDelegate
- ZCUIVoiceDelegate
- ZCXAlbumDelegate

Constant References

- BottomButtonClickTag
- ButtonClickTag
- EmojiBoardActionType
- ExitType
- NSTimerListenerType
- PageClickTag
- RecordState

appledoc使用

1. 安装下载

```
git clone git://github.com/tomaz/appledoc.git  
cd ./appledoc  
sudo sh install-appledoc.sh
```

2. 验证

```
appledoc --version
```

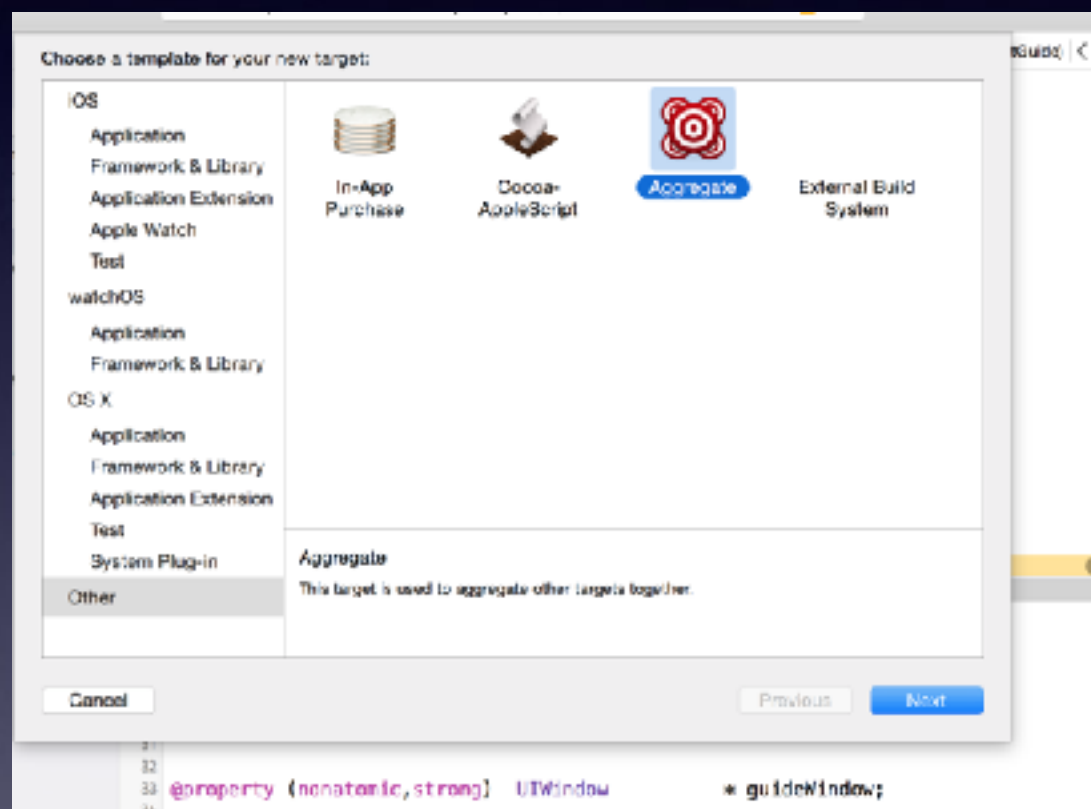
3. 文件路径

```
cd + “文件夹目录”
```

4. 生成docset

```
appledoc --project-name MyProject --project-company ibireme ./
```


5.添加Target



6.添加Run Script

```
#appledoc Xcode script
# Start constants
company="ACME";
companyID="com.ACME";
companyURL="http://ACME.com";
target="iphones";
#target="macosx";
outputPath="~/help";
# End constants
/usr/local/bin/appledoc \
--project-name "${PROJECT_NAME}" \
--project-company "${company}" \
--company-id "${companyID}" \
--docset-atom-filename "${company}.atom" \
--docset-feed-url "${companyURL}/${company}/%DOCSETATOMFILENAME" \
--docset-package-url "${companyURL}/${company}/%DOCSETPACKAGEFILENAME" \
--docset-fallback-url "${companyURL}/${company}" \
--output "${outputPath}" \
--publish-docset \
--docset-platform-family "${target}" \
--logformat xcide \
--keep-intermediate-files \
--no-repeat-first-per \
--no-warn-invalid-crossref \
--exit-threshold 2 \
"${PROJECT_DIR}"
```


注释

- **@brief:** 使用它来写一段你正在文档化的method, property, class, file, struct, 或 enum的短描述信息。
- **@discussion:** 用它来写一段详尽的描述。如果需要你可以添加换行。
- **@param:** 通过它你可以描述一个 method 或 function的参数信息。你可以使用多个这种标签。
- **@return:** 用它来制定一个 method 或 function的返回值。
- **@see:** 用它来指明其他相关的 method 或 function。你可以使用多个这种标签。
- **@sa:** 同前一条类似。
- **@code:** 使用这个标签, 你可以在文档当中嵌入代码段。当在Help Inspector当中查看文档时, 代码通过在一个特别的盒子中用一种不同的字体来展示。始终记住在写的代码结尾处使用**@endcode**标签。
- **@remark:**在写文档时, 用它来强调任何关于代码的特殊之处。

- **@file:** 使用这个标签来指出你正在记录一个文件（header 文件或不是）。如果你将使用Doxygen来输出文档，那么你最好在这个标签后面紧接着写上文件名。它是一个top level 标签。
 - **@header:** 跟上面的类似，但是是在 HeaderDoc中使用。当你不使用Doxygen时，不要使用上面的标签。
 - **@author:** 用它来写下这个文件的创建者信息
 - **@copyright:** 添加版权信息
 - **@version:** 用它来写下这个文件的当前版本。如果在工程生命周期中版本信息有影响时这很重要。
-
- **@class:** 用它来指定一个class的注释文档块的开头。它是一个top level标签，在它后面应该给出class名字。
 - **@interface:** 同上
 - **@protocol:** 同上两个一样，只是针对protocols
 - **@superclass:** 当前class的superclass
 - **@classdesign:** 用这个标签来指出你为当前class使用的任何特殊设计模式（例如，你可以提到这个class是不是单例模式或者类似其它的模式）。
 - **@coclass:** 与当前class合作的另外一个class的名字。
 - **@helps:** 当前class帮助的class的名字。
 - **@helper:** 帮助当前class的class名字。