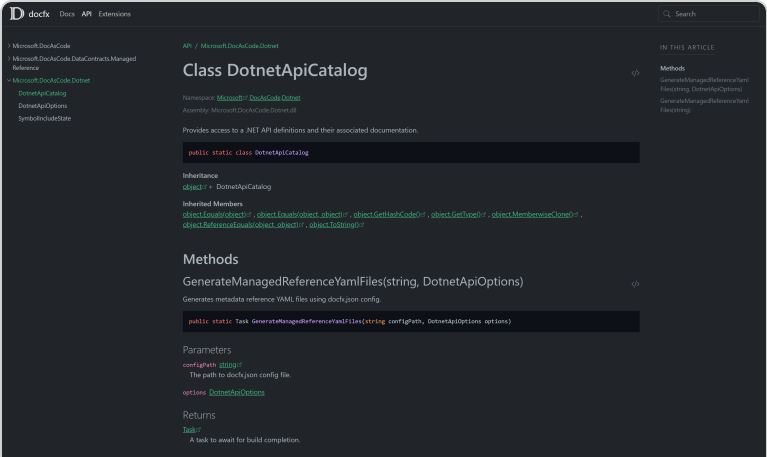
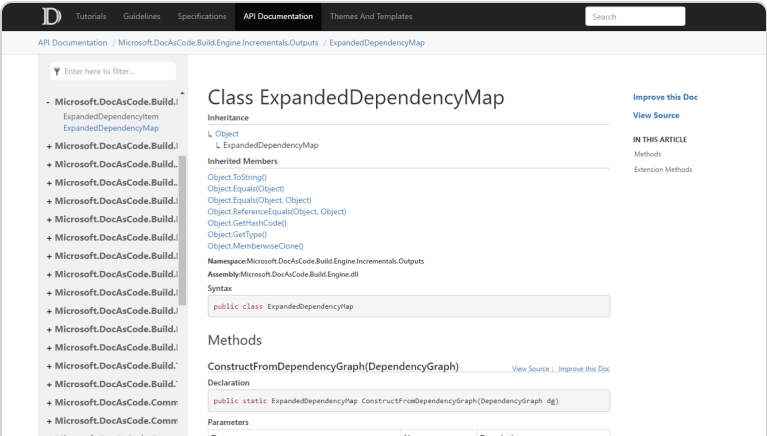


Templates



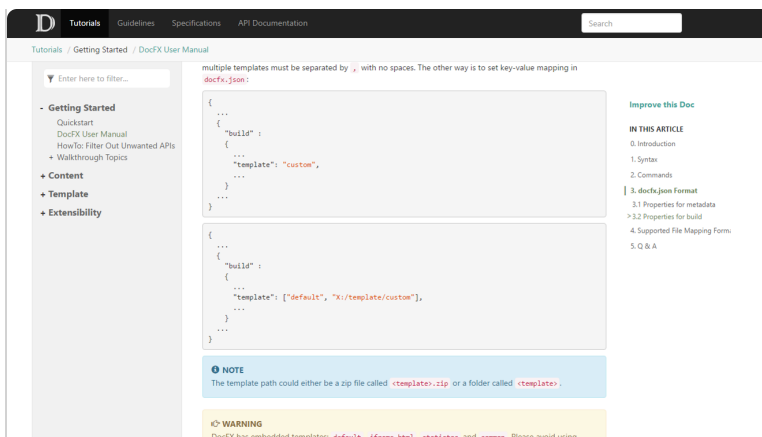
modern

The modern template



default

The default template

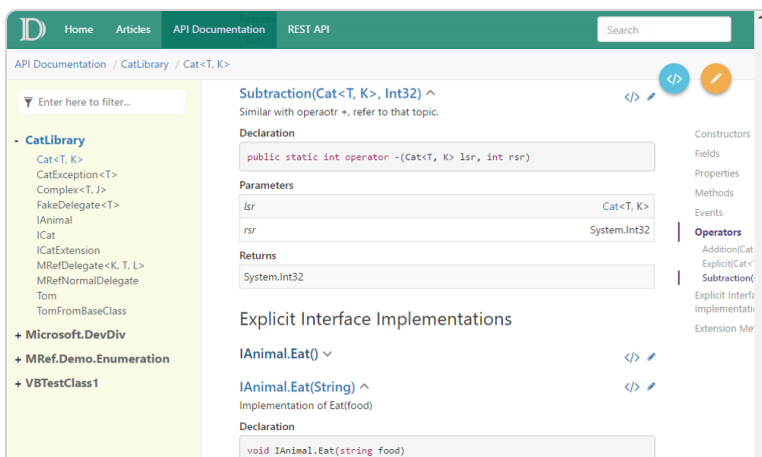


statictoc

The template similar to default template however with static toc. With static toc, the generated web pages can be previewed from local file system.

docfx.json: "template": "statictoc"

docfx: -t statictoc



mathew

A simple template

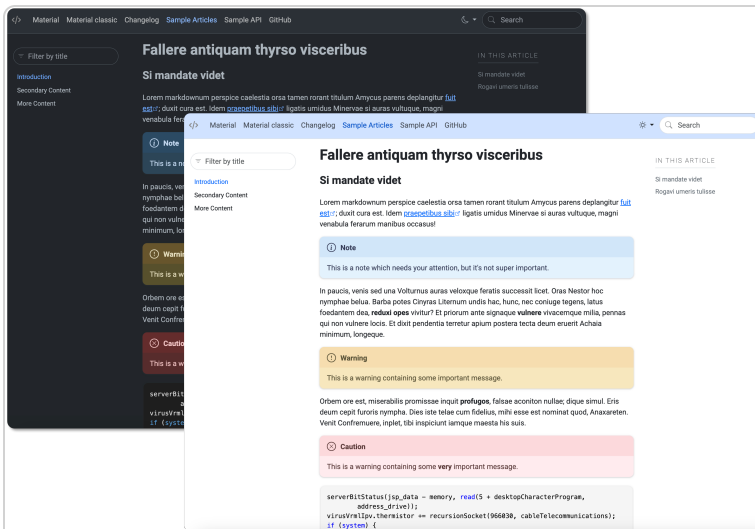
docfx.json: "template":

["default", "mathew/src"]

docfx: -t default, mathew/src

docfx init: `git clone`

`https://github.com/MathewSachin/docfx-tmpl.git` mathew



DocFX Material

A simple material theme for DocFX

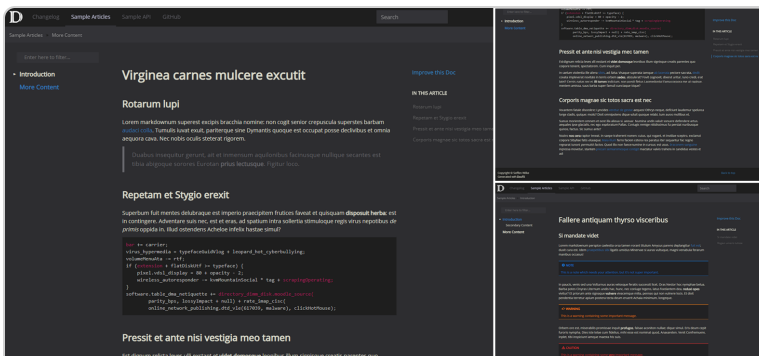
docfx.json: `"template":`

`["default", "material/material"]`

docfx: `-t default,material/material`

docfx init: `git clone`

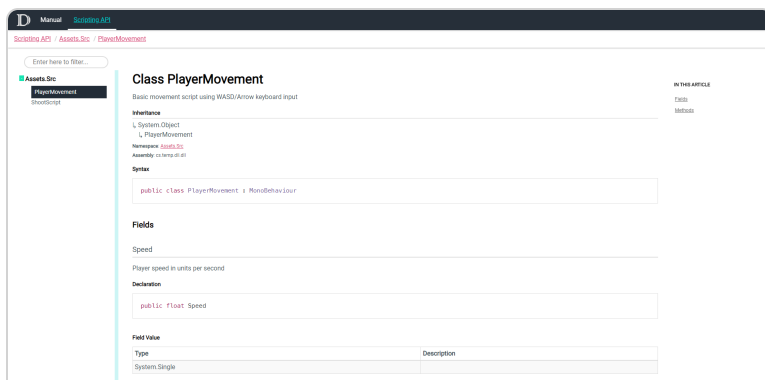
`https://github.com/ovasquez/docfx-material.git` material



darkFX

A dark theme for DocFX .

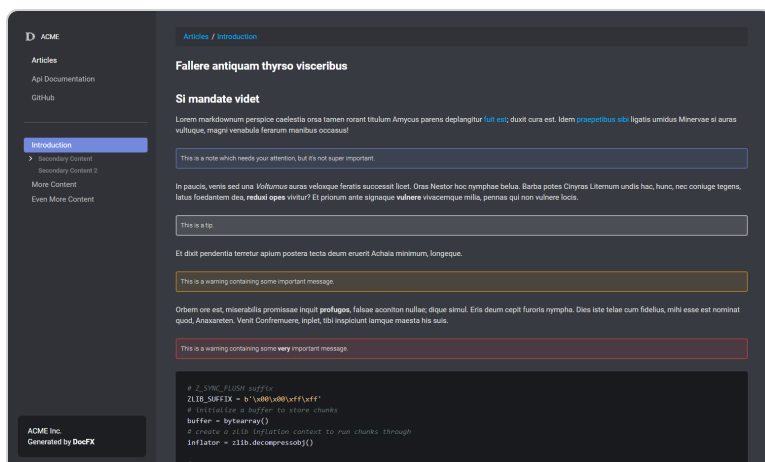
```
docfx.json: "template":  
["default", "templates/darkfx"]  
docfx: -t default, templates/darkfx  
docfx init: git clone  
https://github.com/steffen-  
wilke/darkfx.git darkfx
```



[UnityFX](#)

A theme for Unity-esque documentation

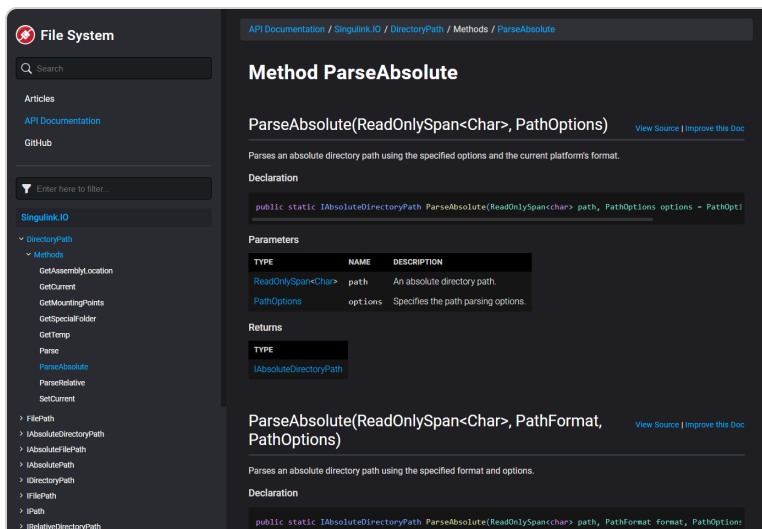
```
docfx.json: "template":  
["default", "templates/unity"]  
docfx: -t statictoc
```



[DiscordFX](#)

DocFX template to create documentation similar to Discord

```
docfx.json: "template":  
["default", "templates/discordfx"]  
docfx: -t default, templates/discordfx
```



SingulinkFX

Customizable responsive DocFX template designed with memberpage plugin compatibility to produce docs similar to Microsoft .NET docs.

```
docfx.json: "template":  
["default", "templates/singulinkfx"]  
docfx: -t default, templates/singulinkfx
```

Enter here to filter...

Installation

DocFX Minimal Template

DocFX Minimal Template is a minimal theme derived from default template.

Features

- Full width (Container-fluid in Bootstrap)
- Minimal white pages
- Simple interface without a breadcrumb
- Table of contents aligned left

Installation

1. Download source files of DocFX minimal template as a zip file from [Here](#) or [GitHub](#).
2. Create `templates` folder in your docfx project folder.
3. Extract the zip file and copy `minimal` folder into the `templates` folder.
4. Apply minimal template by adding `minimal` in your `docfx.json`.

```
"build": {
  "template": {
    "default": "templates/minimal"
  },
}
```

[Improve this Doc](#)

IN THIS ARTICLE

[Features](#)
[Installation](#)

Generated by DocFX

[Back to top](#)

Minimal

A minimal template.

```
docfx.json: "template":
["default", "templates/minimal"]
docfx: -t default, templates/minimal
```

Packages

The screenshot shows the REST API documentation for the 'Pet' resource. The left sidebar lists the API endpoints: 'Pet Store API' (pet, store, user) and 'Contacts API'. The main content area displays the 'AddPet' endpoint, which is a POST request to '/pet'. It includes a 'Request' section with a sample body, a 'Parameters' table, and a 'Responses' table. The 'Responses' table shows a 405 status code for 'Invalid input'. A note at the bottom states: 'NOTE: Add pet only when you needs.'

Pet
Description for pet tag

AddPet
Add a new pet to the store

Request

POST /pet

Parameters

Name	Type	Value	Notes
*body			Pet object that needs to be added to the store

Responses

Status Code	Description	Samples
405	Invalid input	

NOTE: Add pet only when you needs.

UpdatePet
Update an existing pet

[rest.tagpage](#)

It splits the *REST* model into tag level model. With this plugin enabled, operations with the same tag are grouped into one page. If the operation is in multiple tags, it would be included in first tag level page.

```
docfx.json: template: ["default", "  
<output>/rest.tagpage.<version>/content"]  
docfx: -t default,<output>/rest.tagpage.  
<version>/content  
docfx init: nuget install rest.tagpage -  
OutputDirectory <output>
```

The screenshot shows the REST API documentation for the 'DeletePet' endpoint. The left sidebar lists the API endpoints: 'Pet Store API' (addPet, createUser, createUsersWithArrayInput, createUsersWithListInput, deleteOrder, deletePet, deleteUser, findPetsByStatus, findPetsByTags, getInventory, getOrderById, getPetById, getUserByName, loginUser, logoutUser, placeOrder, updatePet, updatePetWithForm, updateUser, uploadFile) and 'Contacts API'. The main content area displays the 'DeletePet' endpoint, which is a DELETE request to '/pet/{petId}'. It includes a 'Request' section with a sample body, a 'Parameters' table, and a 'Responses' table. The 'Responses' table shows 400 and 404 status codes for 'Invalid ID supplied' and 'Pet not found' respectively.

DeletePet
Deletes a pet

pet

DeletePet
Deletes a pet

Request

DELETE /pet/{petId}

Parameters

Name	Type	Value	Notes
api_key	string		
*petId	integer		Pet id to delete

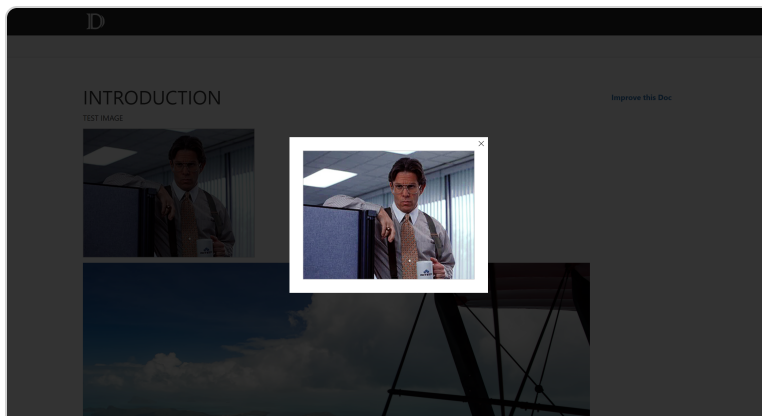
Responses

Status Code	Description	Samples
400	Invalid ID supplied	
404	Pet not found	

[rest.operationpage](#)

It splits the *REST* model into operation level model. If it's enabled together with `rest.tagpage`, the *REST* model will split to tag level first, then split to operation level.

```
docfx.json: template: ["default", "  
<output>/rest.operationpage.  
<version>/content"]  
docfx: -t default,  
<output>/rest.operationpage.  
<version>/content  
docfx init: nuget install  
rest.operationpage -OutputDirectory  
<output>
```

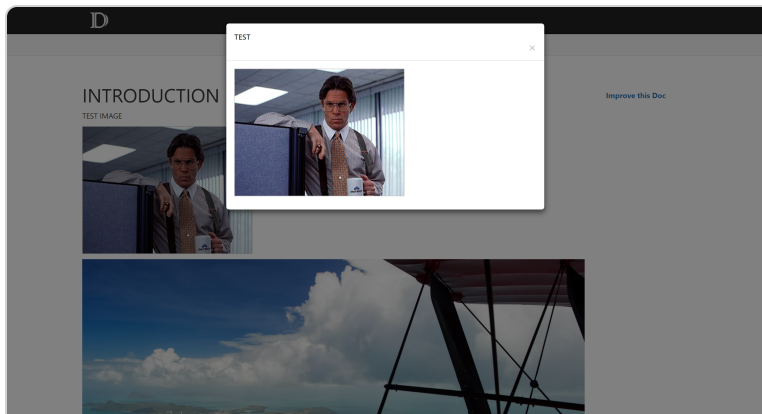


[docfx-lightbox-plugin](#) ([Featherlight](#))[↗](#)

A template which adds a lightbox to each image, using the jquery plugin Featherlight.

```
docfx.json: "template":["default","docfx-  
lightbox-plugin/templates/lightbox-  
featherlight"]  
docfx: -t default,docfx-lightbox-  
plugin/templates/lightbox-featherlight
```

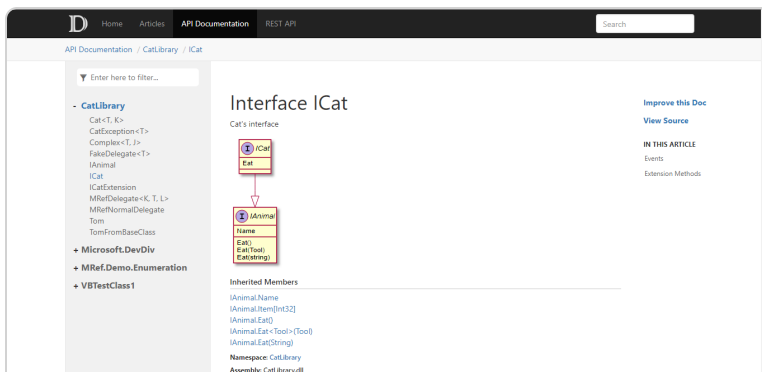
```
docfx init: git clone
https://github.com/roel4ez/docfx-
lightbox-plugin.git docfx-lightbox-plugin
```



[docfx-lightbox-plugin \(Bootstrap Modal\)](#)

A template which adds a lightbox to each image, using the Modal window from Bootstrap.

```
docfx.json: "template": ["default", "docfx-
lightbox-plugin/templates/bootstrap-
modal"]
docfx: -t default,docfx-lightbox-
plugin/templates/bootstrap-modal
docfx init: git clone
https://github.com/roel4ez/docfx-
lightbox-plugin.git docfx-lightbox-plugin
```



[DocFx.Plugins.PlantUml](#)

A template to render PlantUml diagrams from markdown code blocks.

```
docfx.json: "template":  
["default", "DocFx.Plugins.PlantUml/template"]  
docfx: -t  
default,DocFx.Plugins.PlantUml/template  
docfx init: nuget install  
DocFx.Plugins.PlantUml -ExcludeVersion -  
OutputDirectory .
```

Tools

```
# This is an automatically generated file
items:
- name: Getting started
  href: getting-started/README.md
  items:
- name: Using DocFx and Companion Tools to generate a Documentation website
  href: getting-started/README.md
- name: Deploy the DocFx Documentation website to an Azure Website automatically
  href: getting-started/deploy-docfx-azure-website.md
- name: Customize the Look and Feel
  href: getting-started/customize-look-and-feel.md
```

[DocFxTocGenerator](#)

Generate a Table of Contents (TOC) in YAML format for DocFX. It has features like the ability to configure the order of files and the names of documents and folders. This tool is part of the DocFx Companion Tools set that can be installed using Chocolatey.

```
docfx: TocGenerator -d <docs folder> [-o
<output folder>] [-vsi]
docfx init: git clone
https://github.com/Ellerbach/docfx-
companion-tools.git
```

```
cs\fr\plant-production\ex
/.. /.attachments/de.png
erenced:
ttachments\de.png
ttachments\en.jpg
urn code 1
```

[DocLinkChecker](#)

Validate links in documents and check for orphaned attachments in the .attachments folder. The tool indicates whether there are errors or warnings, so it can be used in a CI pipeline. It can also clean up orphaned attachments automatically. This tool is part of the DocFx Companion Tools set that can be installed using Chocolatey.

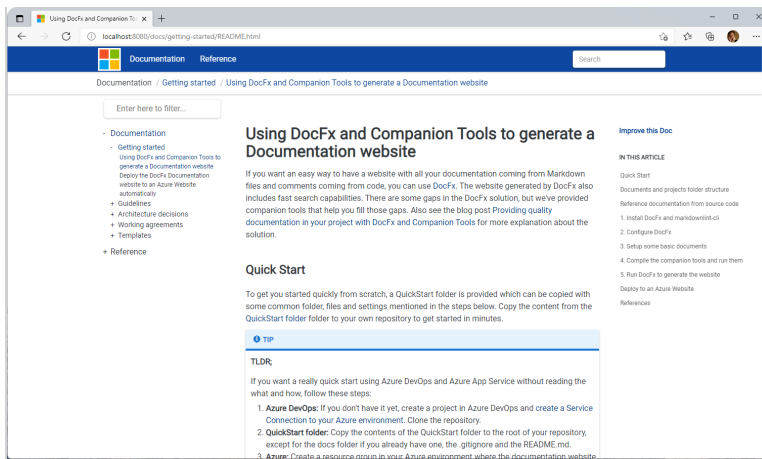
```
docfx: DocLinkChecker -d <docs folder> [-vac]
docfx init: git clone
https://github.com/Ellerbach/docfx-companion-tools.git
```

```
Translating C:\DMP\userdocs\de\Data-size-estimation.md
Translating
.....
Saving C:\DMP\userdocs\fr\Data-size-estimation.md
Translating C:\DMP\userdocs\de\index.md
Translating
..
Saving C:\DMP\userdocs\fr\index.md
Translating C:\DMP\userdocs\de\plant-production\example.md
Translating
....
Saving C:\DMP\userdocs\fr\plant-production\example.md
Translating C:\DMP\userdocs\de\plant-production\second-file.md
Translating
..
Saving C:\DMP\userdocs\fr\plant-production\second-file.md
Translating C:\DMP\userdocs\de\plant-production\another-dir\another\and-another\something.md
Translating
..
Saving C:\DMP\userdocs\fr\plant-production\another-dir\another\and-another\something.md
Process finished. 5 translated and properly created. Please make sure to run the Markdown linter and also check the file links and images.
```

[DocLanguageTranslator](#)

Allows to generate and translate automatically missing files or identify missing files in multi language pattern directories. This tool is part of the DocFx Companion Tools set that can be installed using Chocolatey.

```
docfx: DocLanguageTranslator -d <docs folder> [-k <key>] [-l <location>] [-cv]
docfx init: git clone
https://github.com/Ellerbach/docfx-companion-tools.git
```



DocFx Quick Start

A repo containing documentation, configuration and sample pipelines to help you get started quickly with DocFx. The Quick Start can be used as a reference or to copy elements from it to your own repo. The Quick Start itself can be generated to a website using DocFx as well. It uses the DocFx Companion Tools *DocFxTocGenerator* and *DocLinkChecker*.

docfx init: `git clone`

`https://github.com/mtirionMSFT/DocFxQuickStart.git`



[Addin for Cake Build System](#)

Cake AddIn that generates documentation for .Net API reference and markdown files using DocFx.