

# DocGist

DocGist is a URL proxy tool that converts [AsciiDoc](#) documents fetched from Gists (<http://gist.github.com>), GitHub repositories, Dropbox folders and other sources to HTML. The conversion to HTML is performed in the browser (client-side) using the [Asciidoctor.js](#) JavaScript library. DocGist can render documents located anywhere, as long as the host permits cross-domain access.

The DocGist project is found at <https://github.com/asciidoctor/docgist> where you can get the code, file issues and contribute.



DocGist uses the [latest release \(1.5.3\)](#) of Asciidoctor. If you want to use the syntax of older versions, add `:compat-mode:` to the document header or by setting it in the btn:[Attributes] menu. There's more information on this in the [migration guide](#).

## The basics

### How to use

- Create/locate a gist on GitHub (or use a file in Dropbox).
  - Write text using AsciiDoc syntax in it.
  - Save it.
- Enter the URL (or id) of the gist in the form on top of this page and hit Enter on your keyboard.
- The page is rendered.
- Share the URL to the page with others so they can read it.

For other possible hosts that can store documents, see [Source documents](#).

### Images

Relative image URLs are resolved to the same location as the document. If the images are located elsewhere, use the `imagesdir` attribute to point out the location. See the [Asciidoctor user manual](#) for the details. There's also an [image example](#) here on DocGist.

```
image::sunset.jpg[]
```

Is rendered as:

[sunset] | *sunset.jpg*

## ☰ Table of contents

The table of content appears below the header by default. You can alter this by setting a different value for the `toc` attribute. The available options are found in the `btn:[Attributes]` menu.

To add a table of contents at any location you want, put this in the place you want it to appear:

```
toc::[]
```

and set the `toc` attribute to `macro` (in the document or via the `btn:[Attributes]` menu).

There's such a `toc` macro embedded in this document. If you set the `toc` attribute to `macro` on this page, the table of contents will appear below this line!

## ⌞ Source code highlighting

By default, source code is highlighted using [CodeMirror](#). See the [CodeMirror example](#) for more information.

Simply include the source code like this:

```
[source,ruby]
.app.rb
----
require 'sinatra'
get '/hi' do
  "Hello World!"
end
----
```

This is how it gets rendered:

*app.rb*

```
require 'sinatra'
get '/hi' do
  "Hello World!"
end
```

[Prettify](#) is supported as well. See the [Prettify example](#) for how to activate it.

[highlight.js](#) is supported too. See the [highlight.js example](#) for how to activate it.

If you want to set a default language for `source` blocks, set the `source-language` attribute in the document header. See [Source language](#) for how to use it.



- Only one source highlighter can be defined per document when using DocGist.
- The source highlighter must be set in the document header.



#### Experimental Features in DocGist

- Highlighting can be used inline as well, for example `[src-ruby]'do'` renders like: `do`.
- Multiple code snippets in different languages can be combined into a tabbed view. See the [Tabbed source](#) example for more information.

## ▶▶ Advanced features

### 📊 Math

Thanks to [MathJax](#) mathematical expressions can be typeset in DocGist documents. The content can be written as [AsciiMath](#) or use [TeX/LaTeX](#) notation.

For block content, do like this:

```
[stem]
++++
sqrt(4) = 2
++++
```

It will render like below:

$$\sqrt{4} = 2$$

You can use it inline as well, for example:

Water (stem:[H\_2O]) is a critical component.

Which renders like this: Water ( $H_2O$ ) is a critical component.



There's no need for special delimiters around the expression as the MathJax documentation suggests. This is handled automatically by AsciiDoctor!

The default notation is AsciiMath, but LaTeX can be used like this:

```
latexmath:[C = \alpha + \beta Y^{\gamma} + \epsilon]
```

$C = \alpha + \beta Y^{\gamma} + \epsilon$

## Keyboard shortcuts and more

This is a common shortcut on Windows systems: `kbd:[Ctrl+Alt+Delete]`.

Here's the source:

```
kbd:[Ctrl+Alt+Delete]
```

You might want to represent a user interface button like `btn:[Save]`.

Here's how to do it:

```
btn:[Save]
```



Here on DocGist we use actual (extra small) buttons from Bootstrap. In normal AsciiDoctor it would have looked more along the lines of **[ Save ]**.

Yet another nifty feature, representing a menu item:

```
menu:File[New...]
```

Source:

```
menu:File[New...]
```



Representing keyboard shortcuts, buttons, and menu items are experimental features in AsciiDoctor. DocGist has the **experimental** flag set by default for your convenience!

For the reference documentation, see [User Interface Macros](#).

## Tips and tricks

Links to other DocGists can use this syntax: `link:./?5897167[DocGist intro]` which renders as [DocGist intro](#). This makes things easier in case you run DocGist locally to fiddle with the “backend” (it’s a frontend really).

You can view the source Gist of this page by clicking on the green button in the navbar.

How about some UML? The following image is included using the syntax `image::http://yuml.me/c9ce39b0.png[]`.

`[c9ce39b0] | http://yuml.me/c9ce39b0.png`

The URL is a reference to a yUML image. yUML is tool for creating UML diagrams online.

## Source documents

DocGist can view documents fetched from a number of different sources. It parses the URL to find out how to use it, and tries to be a bit smart to keep DocGist URLs short.

### GitHub Gist

A public or private GitHub Gist can be used. Example URL: <https://gist.github.com/nawroth/5897167> A GitHub Gist allows for storing multiple documents, but DocGist will only the first AsciiDoc document it finds.

### GitHub File

A file in a public git repo hosted at GitHub. Example URL: <https://github.com/asciidoctor/docgist/blob/master/gists/example.adoc>

### Dropbox Public folder

Put a file in the *Public* folder of your Dropbox, grab the URL to it. Example URL: <https://dl.dropboxusercontent.com/u/10666617/AsciiDoc/example.adoc>

### Dropbox shared private file

Use the share-link of a private file in Dropbox. Example URL: <https://www.dropbox.com/s/ttib5v9pfs23p9z/example.adoc>

### Copy.com

Use the public link to the document. Example URL: <https://copy.com/LdKsHnQbEZsl29BW>

### Etherpad

**Etherpad** is an online editor providing collaborative editing in real-time. There are different Etherpad hosts, these are ones that worked well when we tried them: <https://beta.etherpad.org/>, <https://piratepad.ca/>, <https://factor.cc/pad/>, <https://pad.systemli.org/>, <https://pad.fnordig.de/>, <https://notes.typo3.org/>, <https://pad.lqdn.fr/>, <https://pad.okfn.org/>, <https://beta.publishwith.me/>, <https://etherpad.tihlde.org/>, <https://pad.tihlde.org/>, <https://etherpad.wikimedia.org/>, <https://etherpad.fr/>, <https://piratenpad.de/>, <https://bitpad.co.nz/>, <http://notas.dados.gov.br/>, <http://free.primarypad.com/>, <http://board.net/>, <https://pad.odoo.com/>, <http://pad.planka.nu/>, <http://qikpad.co.uk/>, <http://pad.tn/>, <http://lite4.framapad.org/>, <http://pad.hdc.pw/> Note that the content might get removed from the host after some time. Example URL: <https://beta.etherpad.org/p/docgist>

### Google Docs Documents

Write normal AsciiDoc in the document. Then use the *share* link, set so that anyone with the link can view (at least) the document. Example URL: <https://docs.google.com/document/d/1lEQTEAQRVxTtDfQ1N9HJ6azeOLYCKlGG17NOs72NWsU/edit>

## Any URL

DocGist accepts documents from any host which allows for cross-domain requests. This is the kind of error you'll see in the console when a host doesn't support it:

```
No 'Access-Control-Allow-Origin' header is present
on the requested resource.
Origin 'http://gist.asciidoctor.org' is therefore
not allowed access.
```

To make it work, the server should respond with one of the following headers: `Access-Control-Allow-Origin:http://gist.asciidoctor.org` or `Access-Control-Allow-Origin:*`.

### *Additional services that should get added?*

Please tell us if there's some additional service you'd like to see support for. There's two different ways a host can be integrated:

#### **Direct file access**



Dropbox and Google Docs are examples of this. We only calculate the URL of the file, and request it. The server must respond with the correct headers as outlined above.

#### **Through a public API**

This is what we use for GitHub. Note that we can only use an API as long as authorization isn't required.

To suggest an addition, file an [issue](#)!