

DoujinDownloader

Simple app on **.NET Core 3.1** to create and support your local doujins repo. You can build project in **VS2019 (16.4.3+)** or in **VSCode (1.41.1+)** with [omnisharp-vscode](#) extension (**1.21.9+***).

Now only creates file with `uri` list, that you can use with, for example, [HitomiDownloader](#).



Icon is made by [Freepik](#) from [FlatIcon](#).



Current version

Current stable can be found here: `release v1.0.0`.

Information about changes since previous releases can be found in [changelog](#). This project supports [SemVer 2.0.0](#) (template is `{MAJOR}.{MINOR}.{PATCH}.{BUILD}`).

Previous versions can be found on [releases](#) and [branches](#) pages.

Dependencies

- [Newtonsoft.Json](#) – 12.0.3;
- [CommandLineParser](#) – 2.7.82;

Input parameters

Short name	Long name	Description	Required?
-i	--input	Input file (<code>.md</code> / <code>.json</code>)	Yes
-j	--json	Path to converted from <code>.md</code> <code>.json</code> file	No
-a	--artist	Artist name to get from input file	No
-u	--uris	Path to ready <code>uris.txt</code> file	No
	--version	Current version	
	--help	Message about command line options	

`-i/--input` is a `string` with path to `.md` or `.json` file with doujins list.

`-j/--json` is a `string` with path to converted `.json` file with doujins list. If not set – created near `.exe`.

`-a/--artist` is a **Artist's name** `string` to get from input file. If not set – gets all artist.

`-u/--uris` is a `string` with path to read `.txt` file with doujins uris. If not set – created near `.exe`.

Simple example looks like this: `DoujinDownloader -i Doujins.md -j Doujins.json -a ArtistName -u Uris.txt`.

How to use md

```
## Author name
### Subsection (tankoubon, etc)
- [ ] [Doujin name](uri), Language //Not Downloaded
- [x] [Doujin name]() //Downloaded
```

Uri, language and subsection are optional and can be null.

[Example of .md file](#)

How to use json

[Example of .json file](#)

Localization

Localizable strings are located in `Localization/Strings.resx` file. You can add your translation (e.g. added `Strings.Ru.resx` file) and create pull request.

Currently, application is available on **English** and **Russian** languages.

Contributing

Feel free to contribute, make forks, change some code, add [issues](#), etc.