

# DoujinDownloader

---

Simple app on **.NET Core 3.1** to create and support your local doujins repo.

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 v1.0.1`.

For latest build from master you can use [docker image from github](#).

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.

## Build

---

Solution can be build in **VS2019 (16.6.2+)**. You can also build projects in **VSCode (1.46.1+)** with [omnisharp-vscode \(1.22.1+\)](#) extensions. Projects targets **.NET Core 3.1**, so you'll need **.NET Core 3.1.301 SDK**.

**Release** binaries are made by `publish.ps1` script. Take a look at it in the repo. Note, that running this script requires installed **PowerShell** or [PowerShell Core](#) for **Linux/OSX** systems.

## Docker Image

---

You can build docker image by running `docker.ps1` script with your **PowerShell/PowerShell Core (on Linux)**. It'll create `doujindownloader` image.

## Dependencies

---

- [CommandLineParser](#) – 2.8.0;

## 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 readu `.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](uri)
### Subsection //tankoubon, etc -- optional
- [ ] [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.