
fdroid-dl Documentation

Release 0.1.0

t4skforce

Jul 02, 2020

Contents:

1	Command Line Interface	3
2	Model's	5
3	Update Interface	7
4	Download Interface	9
5	Generic JSON Encoder	11
6	Index Postprocessor	13
7	Indices and tables	15
	Python Module Index	17
	Index	19

fdroid-dl is a python based f-droid mirror generation and update utility. Point at one or more existing f-droid repositories and the utility will download the metadata (pictures, descriptions,..) for you and place it in your local system.

```
Usage: fdroid-dl [OPTIONS] COMMAND [ARGS]...
```

```
# fdroid-dl update && fdroid update
```


CHAPTER 1

Command Line Interface

Main command parameters

Usage: `fdroid-dl [OPTIONS] COMMAND [ARGS]...`

Is a python based f-droid mirror generation and update utility. Point at one or more existing f-droid repositories and the utility will download the metadata (pictures, descriptions,..) for you and place it in your local system.

Simply run "`fdroid-dl update && fdroid update`" in your folder with repo and you are set.

Options:

<code>-d, --debug</code>	enable debug level logging
<code>-c, --config FILE</code>	location of your <code>fdroid-dl.json</code> configuration file [default: <code>fdroid-dl.json</code>]
<code>-r, --repo DIRECTORY</code>	location of your fdroid repository to store the apk files [default: <code>./repo</code>]
<code>-m, --metadata DIRECTORY</code>	location of your fdroid metadata to store the asset files [default: <code>./metadata</code>]
<code>--cache DIRECTORY</code>	location for <code>fdroid-dl</code> to store cached data [default: <code>./.cache</code>]
<code>--help</code>	Show this message and exit.

Commands:

<code>update</code>	starts updating process
---------------------	-------------------------

Update command parameters

Usage: `fdroid-dl update [OPTIONS] COMMAND [ARGS]...`

Options:

<code>--index / --no-index</code>	download repository index files [default: <code>True</code>]
<code>--metadata / --no-metadata</code>	download metadata asset files [default: <code>True</code>]

(continues on next page)

(continued from previous page)

<code>--apk / --no-apk</code>	download apk files [default: True]
<code>--apk-versions INTEGER</code>	how many versions of apk to download [default: 1]
<code>--src / --no-src</code>	download src files [default: True]
<code>--threads INTEGER</code>	configure number of parallel threads used for download [default: 10]
<code>--head-timeout INTEGER</code>	maximum time in seconds a HEAD request is allowed to take [default: 10]
<code>--index-timeout INTEGER</code>	maximum time in seconds index file download is allowed to take [default: 60]
<code>--download-timeout INTEGER</code>	maximum time in seconds file download is allowed to take [default: 60]
<code>--help</code>	Show this message and exit.


```
class fdroid_dl.model.Config (filename='fdroid-dl.json',      repo_dir='./repo',      meta-  
                                data_dir='./metadata', cache_dir='.cache', apk_versions=1)
```

Main Class responsible for handling the f-droid.json config file.

This Class represents the config file used by fdroid-dl and will default to <https://f-droid.org/repo/> as bas repo and add at least the **org.fdroid.fdroid** app for download into the queue.

index (*url*)

Return Index based on url.

Searches for downloaded index file based on repository url as configured in config file.

Parameters **url** (*str*) – url of repo in config file

Returns index found for given url

Return type *fdroid_dl.model.Index*

load (*file=None*)

Loads given file handler or filename into Config object and resolves.

repo (*url*)

Return RepoConfig based on url.

Searches in config file for given url and returns the corresponding RepoConfig Class representation of it. if the repo is marked as having errors it will not be found. A KeyError is being raised if given url is not found or has an error node in the json file.

Parameters **key** (*str*) – url of repo in config file

Returns index found for given key

Return type *fdroid_dl.model.Index*

Raises **KeyError** – raised if given url is not found or is in error state

```
class fdroid_dl.model.Index (key=None, filename=None, format='json', default_locale='en-US',  
                                store={})
```

Main Class responsible for handling the f-droid's index.xml and index-v1.json files.

```
monkeypatch (  
    fixup metadata paths -> url
```

CHAPTER 3

Update Interface

Manages updates

```
class fdroid_dl.update.Update (config, max_workers=10, head_timeout=10, index_timeout=60,  
                               download_timeout=60)  
    handles downloading of repo related data
```


CHAPTER 4

Download Interface

```
class fdroid_dl.download.FuturesSessionFlex (max_workers=1, user_agent='Mozilla/5.0
                                         (compatible; Googlebot/2.1;
                                         +http://www.google.com/bot.html)', *args,
                                         **kwargs)

    close ()
        Closes all adapters and as such the session

    map (pattern='http://', session=None)
        if called with session None -> default session for ctor is used

    request (*args, **kwargs)
        Maintains the existing api for Session.request.

        Used by all of the higher level methods, e.g. Session.get.

        The background_callback param allows you to do some processing on the response in the background, e.g.
        call resp.json() so that json parsing happens in the background thread.

        :rtype : concurrent.futures.Future

class fdroid_dl.download.FuturesSessionVerifiedDownload (*args, **kwargs)
```

Generic JSON Encoder

```
class fdroid_dl.json.GenericJSONEncoder(*, skipkeys=False, ensure_ascii=True,
                                         check_circular=True, allow_nan=True,
                                         sort_keys=False, indent=None, separators=None,
                                         default=None)
```

custom encoder so we can serialize Config to json

default (*obj*)

Implement this method in a subclass such that it returns a serializable object for *o*, or calls the base implementation (to raise a `TypeError`).

For example, to support arbitrary iterators, you could implement default like this:

```
def default(self, o):
    try:
        iterable = iter(o)
    except TypeError:
        pass
    else:
        return list(iterable)
    # Let the base class default method raise the TypeError
    return JSONEncoder.default(self, o)
```


CHAPTER 6

Index Postprocessor

```
class fdroid_dl.processor.IndexFileProcessor(*args, **kwargs)
```


CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`

f

- `fdroid_dl.download`, 9
- `fdroid_dl.json`, 11
- `fdroid_dl.model`, 5
- `fdroid_dl.processor`, 13
- `fdroid_dl.update`, 7

C

`close()` (*fdroid_dl.download.FuturesSessionFlex method*), 9

`Config` (*class in fdroid_dl.model*), 5

D

`default()` (*fdroid_dl.json.GenericJSONEncoder method*), 11

F

`fdroid_dl.download` (*module*), 9

`fdroid_dl.json` (*module*), 11

`fdroid_dl.model` (*module*), 5

`fdroid_dl.processor` (*module*), 13

`fdroid_dl.update` (*module*), 7

`FuturesSessionFlex` (*class in fdroid_dl.download*), 9

`FuturesSessionVerifiedDownload` (*class in fdroid_dl.download*), 9

G

`GenericJSONEncoder` (*class in fdroid_dl.json*), 11

I

`Index` (*class in fdroid_dl.model*), 5

`index()` (*fdroid_dl.model.Config method*), 5

`IndexFileProcessor` (*class in fdroid_dl.processor*), 13

L

`load()` (*fdroid_dl.model.Config method*), 5

M

`map()` (*fdroid_dl.download.FuturesSessionFlex method*), 9

`monkeypatch()` (*fdroid_dl.model.Index method*), 5

R

`repo()` (*fdroid_dl.model.Config method*), 5

`request()` (*fdroid_dl.download.FuturesSessionFlex method*), 9

U

`Update` (*class in fdroid_dl.update*), 7