
GuessIt Documentation

Release 2.1.4

Remi Alvergnat

Jun 01, 2017

Contents

1	Migration note	3
2	Install	5
3	Usage	7
4	REST API	9
5	Support	11
6	License	13

GuessIt is a python library that extracts as much information as possible from a video filename.

It has a very powerful matcher that allows to guess properties from a video using its filename only. This matcher works with both movies and tv shows episodes.

For example, GuessIt can do the following:

```
$ guessit "Treme.1x03.Right.Place,.Wrong.Time.HDTV.XviD-NoTV.avi"
For: Treme.1x03.Right.Place,.Wrong.Time.HDTV.XviD-NoTV.avi
GuessIt found: {
    "title": "Treme",
    "season": 1,
    "episode": 3,
    "episode_title": "Right Place, Wrong Time",
    "format": "HDTV",
    "video_codec": "XviD",
    "release_group": "NoTV",
    "container": "avi",
    "mimetype": "video/x-msvideo",
    "type": "episode"
}
```


CHAPTER 1

Migration note

GuessIt 2 has been rewritten from scratch. GuessIt is now a release name parser only, and support for additional features like hashes computations has been dropped.

To migrate from guessit 0.x or 1.x, please read the migration page.

CHAPTER 2

Install

Installing GuessIt is simple with `pip`:

```
$ pip install guessit
```

You can also install from sources.

CHAPTER 3

Usage

GuessIt can be used from command line:

```
$ guessit
usage: guessit [-h] [-t TYPE] [-n] [-Y] [-D] [-L ALLOWED_LANGUAGES]
              [-C ALLOWED_COUNTRIES] [-E] [-T EXPECTED_TITLE] [-G EXPECTED_GROUP]
              [-f INPUT_FILE] [-v] [-P SHOW_PROPERTY] [-a] [-l] [-l] [-j] [-y]
              [-c CONFIG] [--no-embedded-config] [-p] [-V] [--version]
              [filename [filename ...]]

positional arguments:
  filename              Filename or release name to guess

optional arguments:
  -h, --help            show this help message and exit

Naming:
  -t TYPE, --type TYPE  The suggested file type: movie, episode. If undefined,
                        type will be guessed.
  -n, --name-only       Parse files as name only, considering "/" and "\" like
                        other separators.
  -Y, --date-year-first
                        If short date is found, consider the first digits as
                        the year.
  -D, --date-day-first  If short date is found, consider the second digits as
                        the day.
  -L ALLOWED_LANGUAGES, --allowed-languages ALLOWED_LANGUAGES
                        Allowed language (can be used multiple times)
  -C ALLOWED_COUNTRIES, --allowed-countries ALLOWED_COUNTRIES
                        Allowed country (can be used multiple times)
  -E, --episode-prefer-number
                        Guess "serie.213.avi" as the episode 213. Without this
                        option, it will be guessed as season 2, episode 13
  -T EXPECTED_TITLE, --expected-title EXPECTED_TITLE
                        Expected title to parse (can be used multiple times)
  -G EXPECTED_GROUP, --expected-group EXPECTED_GROUP
```

```

Expected release group (can be used multiple times)

Input:
  -f INPUT_FILE, --input-file INPUT_FILE
                                Read filenames from an input text file. File should
                                use UTF-8 charset.

Output:
  -v, --verbose                 Display debug output
  -P SHOW_PROPERTY, --show-property SHOW_PROPERTY
                                Display the value of a single property (title, series,
                                video_codec, year, ...)
  -a, --advanced                Display advanced information for filename guesses, as
                                json output
  -s, --single-value            Keep only first value found for each property
  -l, --enforce-list            Wrap each found value in a list even when property has
                                a single value
  -j, --json                    Display information for filename guesses as json
                                output
  -y, --yaml                    Display information for filename guesses as yaml
                                output

Configuration:
  -c CONFIG, --config CONFIG
                                Filepath to the configuration file. Configuration
                                contains the same options as those command line
                                options, but option names have "-" characters replaced
                                with "_". If not defined, guessit tries to read a
                                configuration default configuration file at
                                ~/.guessit/options.(json|yaml|yaml) and
                                ~/.config/guessit/options.(json|yaml|yaml). Set to
                                "false" to disable default configuration file loading.
  --no-embedded-config         Disable default configuration.

Information:
  -p, --properties              Display properties that can be guessed.
  -V, --values                  Display property values that can be guessed.
  --version                     Display the guessit version.

```

It can also be used as a python module:

```

>>> from guessit import guessit
>>> guessit('Treme.1x03.Right.Place,.Wrong.Time.HDTV.XviD-NoTV.avi')
MatchesDict([('title', 'Treme'), ('season', 1), ('episode', 3), ('episode_title',
↪ 'Right Place, Wrong Time'), ('format', 'HDTV'), ('video_codec', 'XviD'), ('release_
↪ group', 'NoTV'), ('container', 'avi'), ('mimetype', 'video/x-msvideo'), ('type',
↪ 'episode')])

```

MatchesDict is a dict that keeps matches ordering.

Command line options can be given as dict or string to the second argument.

CHAPTER 4

REST API

A REST API will be available soon ...

Sources are available in a dedicated [guessit-rest repository](#).

CHAPTER 5

Support

This project is hosted on [GitHub](#). Feel free to open an issue if you think you have found a bug or something is missing in guessit.

GuessIt relies on [Rebulk](#) project for pattern and rules registration.

CHAPTER 6

License

GuessIt is licensed under the [LGPLv3](#) license.