

---

# **GuessIt Documentation**

***Release 3.1.1***

**Remi Alvergnat**

**Oct 27, 2020**



---

## Contents

---

<b>1</b>	<b>Migration note</b>	<b>3</b>
<b>2</b>	<b>Install</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Configuration</b>	<b>11</b>
<b>5</b>	<b>REST API</b>	<b>13</b>
<b>6</b>	<b>Support</b>	<b>15</b>
<b>7</b>	<b>License</b>	<b>17</b>



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",
  "source": "HDTV",
  "video_codec": "XviD",
  "release_group": "NoTV",
  "container": "avi",
  "mimetype": "video/x-msvideo",
  "type": "episode"
}
```



# CHAPTER 1

---

## Migration note

---

In GuessIt 3, some properties and values were renamed in order to keep consistency and to be more intuitive.

To migrate from guessit 2.x to guessit 3.x, please read the migration page. To migrate from guessit 0.x or 1.x to guessit 2.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] [--includes INCLUDES]
               [--excludes EXCLUDES] [-f INPUT_FILE] [-v]
               [-P SHOW_PROPERTY] [-a] [-s] [-l] [-j] [-y] [-c CONFIG]
               [--no-user-config] [--no-default-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
```

(continues on next page)

(continued from previous page)

	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)
--includes INCLUDES	List of properties to be detected
--excludes EXCLUDES	List of properties to be ignored
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 configuration file. Configuration file contains the same options as those from command line options, but option names have "-" characters replaced with "_". This configuration will be merged with default and user configuration files.
--no-user-config	Disable user configuration. If not defined, guessit tries to read configuration files at ~/.guessit/options.(json yaml yaml) and ~/.config/guessit/options.(json yaml yaml)
--no-default-config	Disable default configuration. This should be done only if you are providing a full configuration through user configuration or --config option.
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'), ('source', '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

---

### Configuration

---

Find more about Guessit configuration at [configuration page](#).





## CHAPTER 5

---

### REST API

---

A REST API will be available soon . . .

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



## CHAPTER 6

---

### 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 7

---

### License

---

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