mwtp

NDKDD

Apr 15, 2024

CONTENTS:

`mwtp` is a parser for MediaWiki titles which relies on information provided by user (dependency injection) instead of using hard-coded data. Currently it only supports Python 3.10 and later.

It is authored and maintained by NguoiDungKhongDinhDanh (a.k.a. NDKDD). Please direct any suggestions and bug reports to GitHub.

CHAPTER

ONE

INSTALLATION

Since we are talking about Python packages, the only sane way to install `mwtp` is to use `pip`:

```
$ pip install mwtp
```

1.1 Usage

The parser works as simple as follows:

```
from mwtp import TitleParser as Parser

parser = Parser(namespaces_data, namespace_aliases)
title = parser.parse(' _ FoO: this/is A__/talk page _ ')

print(repr(title))  # Title('Talk:This/is A /talk page')
```

`namespaces_data` and `namespace_aliases` can be obtained by making a query to a wiki's API with `action=query&meta=siteinfo&siprop=namespaces|namespacealiases`. Here's how they might look like:

```
namespaces_data = {
  '0': {
    'id': 0,
    'case': 'first-letter',
    'name': '',
    'subpages': False,
    'content': True,
    'nonincludable': False
 },
 '1': {
    'id': 1,
    'case': 'first-letter',
    'name': 'Talk',
    'subpages': True,
    'canonical': 'Talk',
    'content': False,
    'nonincludable': False
  },
  ...: ...
}
namespace_aliases = [
```

(continues on next page)

(continued from previous page)

```
  { 'id': 1, 'alias': 'Foo' },
  ...
]
```

Note that the following format (*&formatversion=1*) is not supported. Always use *&formatversion=2* or *&formatversion=latest*.

```
namespaces_data = {
  '0': { 'id': 0, 'case': 'first-letter', '*': '',          ...: ... },
  '1': { 'id': 1, 'case': 'first-letter', '*': 'Tho lun', ...: ... },
  ...: ...
}
namespace_aliases = [
  { 'id': 1, '*': 'Foo' },
  ...
]
```

Parser.parse() returns a *Title* object which has a bunch of convenient properties for title manipulation:

```
title.namespace             # 1
title.in_content_namespace  # False
title.associated            # Title('This/is A /talk page')
```

A *Title* can be converted back to a `str` using either:

```
str(title)                  # 'Talk:This/is A /talk page'
title.full_name             # 'Talk:This/is A /talk page'
```

Path-like operations are also supported:

```
title + '/Foo'              # Title('Talk:This/is A /talk page/Foo')
title / 'Foo'               # Title('Talk:This/is A /talk page/Foo')
```

See the class's full method list for more information.

1.2 Limitations

Interwiki links (e.g. `w:`) are not supported. Neither do fragments (e.g. `#Foo`).

An interwiki title cannot be resolved completely, since there is no way to know what the other wiki's namespace configurations are.

A title is supposed to represent an "address" or a "path", not the real page. That said, the fragment is not considered a part of a title. However, the parser will strip out the fragment part, if any.

Note that *Title.fragments* returns something entirely different: a list of strings created by splitting the title by `/`.

1.3 API references

1.3.1 mwtp.parser

class mwtp.parser.**Parser**(*namespace_data: Mapping[str, NamespaceDataFromAPI]*, *alias_entries: Sequence[NamespaceAlias]*)

A parser that parse strings using (mostly) data provided by the user.

__init__(*namespace_data: Mapping[str, NamespaceDataFromAPI]*, *alias_entries: Sequence[NamespaceAlias]*) → None

Construct a new parser object from the given data.

Parameters

- **namespace_data** – A `Mapping` that maps string IDs to corresponding namespace data.
- **alias_entries** – A `Sequence` consisting of alias entries.

Attributes

_TITLE_MAX_BYTES: ClassVar[int] = 255

_ILLEGAL_TITLE_CHARACTER: ClassVar[Pattern[str]] = re.compile('[\\u0000-\\u001F#<>[\\]{|}\\u007F\\uFFFD]')

Methods

parse(*string: str*) → *Title*

The main parsing method. Raises a subclass of *InvalidTitle* if the string is not a valid title.

Parameters
string – The string to parse.

Returns
A *Title*, if parsed successfully.

Properties

property namespace_data: dict[str, mwtp._namespace_data.NamespaceData]

The data given to and sanitized by the parser.

1.3.2 mwtp.title

class mwtp.title.**Title**(*name: str*, *, *namespace: int*, *parser:* Parser)

Represents a MediaWiki title.

This class is not meant to be used directly. Use *Parser.parse* instead.

__init__(*name: str*, *, *namespace: int*, *parser:* Parser) → None

Construct a Title object.

Parameters

- **name** – The page name part of the title.
- **namespace** – The namespace of the title.
- **parser** – The parser which constructed the title.

Methods

__add__(*other: str*) → *Title*

Add a string to this title's full name and pass that to the parser.

A *Title* cannot be added to one another since there is no way to determine the namespace of the new title.

__truediv__(*other: str*) → *Title*

Add `/` and `other` to the title and pass that to the parser.

Properties

property associated: Self | None

The title associated to this title, or `None` if there is no such title.

property associated_namespace: int | None

The ID of the talk or subject namespace to which the title's namespace is associated with.

property associated_namespace_data: NamespaceData | None

An object containing all known information about the title's associated namespace or `None` if there is no such namespace. This is retrieved from the parser.

property associated_namespace_name: str | None

The localized name of the title's associated namespace.

property base: Self

A Title object representing the parent title of this title.

property canonical_namespace_name: str | None

The canonical name of the title's namespace.

property extension: str | None

The extension part of a file name, if any.

property fragments: tuple[str, ...]

If the namespace has `.subpages == True`, return a list of strings generated from splitting the title by `/`. Else, return the name wrapped in a list.

property full_name: str

The full title (i.e. `Namespace:Pagename` or `Pagename`).

property in_content_namespace: bool

Whether the namespace of the title is a content namespace.

property is_subpage: bool

Whether the title has a parent title.

property name: str

The title without the namespace.

property namespace: int

The title's namespace ID.

property namespace_data: NamespaceData

An object containing all known information about the title's namespace. This is retrieved from the parser.

property namespace_name: str

The localized name of the title's namespace.

property root: Self

A Title object representing the root title of this title.

property subject: Self

The subject title correspond to this title. Can be itself if it is a subject title.

property tail: str

The rightmost fragment of the title.

property talk: Self | None

The talk title correspond to this title, or `None` if there is no such title.

1.3.3 mwtp.namespace

class mwtp.namespace.**Namespace**(*value*, *names=None*, ***, *module=None*, *qualname=None*, *type=None*, *start=1*, *boundary=None*)

Bases: `IntEnum`

An `IntEnum` that contains all default namespace IDs, from Media (-2) to Category talk (15).

MEDIA = -2

SPECIAL = -1

MAIN = 0

TALK = 1

USER = 2

USER_TALK = 3

PROJECT = 4

PROJECT_TALK = 5

FILE = 6

FILE_TALK = 7

MEDIAWIKI = 8

MEDIAWIKI_TALK = 9

TEMPLATE = 10

TEMPLATE_TALK = 11

HELP = 12

HELP_TALK = 13

CATEGORY = 14

CATEGORY_TALK = 15

1.3.4 Exceptions

exception mwtp.exceptions.InvalidTitle

Bases: Exception

Umbrella exception for all kinds of exceptions a parser might raise.

exception mwtp.exceptions.TitleContainsHTMLEntity

Bases: *InvalidTitle*

Raised if the title contains an HTML entity or something that looks like one.

exception mwtp.exceptions.TitleContainsIllegalCharacter

Bases: *InvalidTitle*

Raised if the title contains illegal characters.

exception mwtp.exceptions.TitleContainsSignatureComponent

Bases: *InvalidTitle*

Raised if the title contains ~~~.

exception mwtp.exceptions.TitleContainsURLEncodedCharacter

Bases: *InvalidTitle*

Raised if the title contains a URL-encoded character.

exception mwtp.exceptions.TitleHasRelativePathComponent

Bases: *InvalidTitle*

Raised if the title contains a relative path component or only consists of either one or two dots.

exception mwtp.exceptions.TitleHasSecondLevelNamespace

Bases: *InvalidTitle*

Raised if the title is determined to be in the `Talk:` namespace while also contains a second valid namespace.

exception mwtp.exceptions.TitleIsBlank

Bases: *InvalidTitle*

Raised if the title contains nothing but whitespaces and/or a leading colon.

exception mwtp.exceptions.TitleIsTooLong

Bases: *InvalidTitle*

Raised if the title's length exceed the maximum length of a title.

exception mwtp.exceptions.TitleStartsWithColon

Bases: *InvalidTitle*

Raised if the page name starts with a colon, or the namespace part starts with more than one colon.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- search

PYTHON MODULE INDEX

m

INDEX

Symbols

A

B

C

E

F

H

I

M

N

P

R

S

T

U