

src

putils_logic package

Submodules

putils_logic.putils module

class **putils_logic.putils.PUtils**

Базовые классы: `object`

Вспомогательный класс для работы с путями и папками

static **bp(*args)**

Построение пути к файлу/директории на основе переданных частей пути

Параметры:

`args` – список частей пути

Результат:

относительный путь к файлу/директории

Тип результата:

`str`

Example:

```
bp('media', 'home', 'user') -> './media/home/user'
```

static **bp_abs(*args)**

Построение пути к файлу/директории на основе переданных частей пути

Параметры:

`args` – список частей пути

Результат:

абсолютный путь к файлу/директории

Тип результата:

`str`

Example:

```
bp_abs('media', 'project', '1.png') -> '/home/alien/git/media/project/1.png'
```

```
static mkdir(directory, parents=True, exists_ok=True)
```

Создание директории в файловой системе

Параметры:

- **directory** (`str`) – путь директории, которую нужно создать
- **parents** – создавать промежуточные подпапки, если не были созданы
- **exists_ok** – не вызывать exception в случае если папка уже создана

Тип результата:

`None`

Результат:

`None`

```
static delete_file(path, logger=None)
```

Удаление файла по переданному пути

Параметры:

- **path** (`str`) – путь файла для удаления
- **logger** – опциональная функция вывода ошибки

Тип результата:

`None`

Результат:

`None`

```
static delete_dir(path, logger=None)
```

Удаление директории

Параметры:

- **path** (`str`) – путь директории для удаления
- **logger** – опциональная функция вывода ошибки

Тип результата:

`None`

Результат:

`None`

```
static touch_file(path)
```

Создание пустого файла

Параметры:

path (`str`) – путь к файлу

Тип результата:

`None`

Результат:

`None`

```
static copy_dir(from_path, to_path)
```

Рекурсивное копирование файлов директории copytree

Параметры:

- **from_path** (`str`) – исходная папка
- **to_path** (`str`) – папка назначения

Тип результата:

`str`

Результат:

Папка назначения

```
static move_dir(from_path, to_path)
```

Рекурсивное перемещение файлов директории mv

Параметры:

- **from_path** (`str`) – исходная папка
- **to_path** (`str`) – папка назначения

Тип результата:

`str`

Результат:

Папка назначения

```
static copy_file(from_path, to_path)
```

Копирование файла с метаданными

Параметры:

- **from_path** (`str`) – исходный файл
- **to_path** (`str`) – папка или файл назначения

Тип результата:

`str`

Результат:

Путь до копии файла

```
static is_empty_file(filename)
```

Проверка на пустоту файла

Reference <https://stackoverflow.com/a/2507871>

Параметры:

filename (`str`) – путь к файлу

Тип результата:

`bool`

Результат:

`bool` размер файла равен 0

`static is_empty_dir(path)`

Проверка на пустоту директории. Директория не содержит файлов

Reference <https://stackoverflow.com/a/59050548>

Параметры:

`path (str)` – путь к директории

Тип результата:

`bool`

Результат:

`bool` кол-во файлов внутри директории равен 0

`static is_dir_exists(path)`

Проверка на существование директории

Reference <https://stackoverflow.com/a/44228213>

Параметры:

`path (str)` – путь к директории

Тип результата:

`bool`

Результат:

`bool` существует ли директория

`static is_file_exists(path)`

Проверка на существование файла

Reference <https://stackoverflow.com/a/44228213>

Параметры:

`path (str)` – путь к директории

Тип результата:

`bool`

Результат:

`bool` существует ли файл

`static zip_path(path, zip_name, *args, **kwargs)`

Запаковка в архив файлов директории

Параметры:

- `path (str)` – путь к директории
- `zip_name (str)` – имя архива (без .zip)
- `args` – параметры `shutil.make_archive`
- `kwargs` – параметры `shutil.make_archive`

Тип результата:

`str`

Результат:

имя архива

```
static unzip(filename, extract_path, *args, **kwargs)
```

Извлечение файлов из архива

Параметры:

- **filename** (`str`) – имя архива
- **extract_path** (`str`) – путь к папке назначения распаковки
- **args** – параметры `shutil.make_archive`
- **kwargs** – параметры `shutil.make_archive`

Тип результата:

`None`

Результат:

None

```
static size_of_file(file)
```

Размер файла в байтах

Параметры:

file (`str`) – путь к файлу

Тип результата:

`int`

Результат:

размер файла в байтах

```
static get_filename_from_path(path)
```

Извлечение файла из пути

Параметры:

path (`str`) – путь к файлу

Тип результата:

`str`

Результат:

имя файла

Example:

`get_filename_from_path(„/hello/world/file.txt“) -> „file.txt“`

```
static get_files(path)
```

Получение списка файлов и папок директории. Не рекурсивное

Параметры:

path (`str`) – путь к директории

Тип результата:

List [str]

Результат:

список файлов/папок с относительными путями

Example:

```
get_files('.') -> ['./.idea', './.git', './setup.py', './.gitignore', ...]
```

```
get_files('/home/Desktop') -> ['/home/p/file.drawio', '/home/Desktop/folder', ...]
```

static `get_files_recursive(path)`

Получение списка файлов и папок директории. Рекурсивное

Параметры:

`path` (str) – путь к директории

Тип результата:

List [str]

Результат:

список файлов/папок с относительными путями

Example:

```
get_files('.') -> ['./.idea/.gitignore', './.idea/vcs.xml', './.idea/sonarlint/
securityhotspotstore/index.pb', ...] get_files('/home') -> ['/home/
readable_serial.js', '/home/node_modules/async/lib/state.js', ...]
```

Module contents

class `putils_logic.PUtils`

Базовые классы: `object`

Вспомогательный класс для работы с путями и папками

static `bp(*args)`

Построение пути к файлу/директории на основе переданных частей пути

Параметры:

`args` – список частей пути

Результат:

относительный путь к файлу/директории

Тип результата:

str

Example:

```
bp('media', 'home', 'user') -> './media/home/user'
```

```
static bp_abs(*args)
```

Построение пути к файлу/директории на основе переданных частей пути

Параметры:

args – список частей пути

Результат:

абсолютный путь к файлу/директории

Тип результата:

str

Example:

```
bp_abs('media', 'project', '1.png') -> '/home/alien/git/media/project/1.png'
```

```
static mkdir(directory, parents=True, exists_ok=True)
```

Создание директории в файловой системе

Параметры:

- **directory** (`str`) – путь директории, которую нужно создать
- **parents** – создавать промежуточные подпапки, если не были созданы
- **exists_ok** – не вызывать exception в случае если папка уже создана

Тип результата:

None

Результат:

None

```
static delete_file(path, logger=None)
```

Удаление файла по переданному пути

Параметры:

- **path** (`str`) – путь файла для удаления
- **logger** – опциональная функция вывода ошибки

Тип результата:

None

Результат:

None

```
static delete_dir(path, logger=None)
```

Удаление директории

Параметры:

- **path** (`str`) – путь директории для удаления
- **logger** – опциональная функция вывода ошибки

Тип результата:

`None`

Результат:

None

static **touch_file**(*path*)

Создание пустого файла

Параметры:

path (`str`) – путь к файлу

Тип результата:

`None`

Результат:

None

static **copy_dir**(*from_path*, *to_path*)

Рекурсивное копирование файлов директории `copytree`

Параметры:

- **from_path** (`str`) – исходная папка
- **to_path** (`str`) – папка назначения

Тип результата:

`str`

Результат:

Папка назначения

static **move_dir**(*from_path*, *to_path*)

Рекурсивное перемещение файлов директории `mv`

Параметры:

- **from_path** (`str`) – исходная папка
- **to_path** (`str`) – папка назначения

Тип результата:

`str`

Результат:

Папка назначения

static **copy_file**(*from_path*, *to_path*)

Копирование файла с метаданными

Параметры:

- **from_path** (`str`) – исходный файл

- `to_path (str)` – папка или файл назначения

Тип результата:

`str`

Результат:

Путь до копии файла

`static is_empty_file(filename)`

Проверка на пустоту файла

Reference <https://stackoverflow.com/a/2507871>

Параметры:

`filename (str)` – путь к файлу

Тип результата:

`bool`

Результат:

`bool` размер файла равен 0

`static is_empty_dir(path)`

Проверка на пустоту директории. Директория не содержит файлов

Reference <https://stackoverflow.com/a/59050548>

Параметры:

`path (str)` – путь к директории

Тип результата:

`bool`

Результат:

`bool` кол-во файлов внутри директории равен 0

`static is_dir_exists(path)`

Проверка на существование директории

Reference <https://stackoverflow.com/a/44228213>

Параметры:

`path (str)` – путь к директории

Тип результата:

`bool`

Результат:

`bool` существует ли директория

`static is_file_exists(path)`

Проверка на существование файла

Reference <https://stackoverflow.com/a/44228213>

Параметры:

`path (str)` – путь к директории

Тип результата:

`bool`

Результат:

bool существует ли файл

```
static zip_path(path, zip_name, *args, **kwargs)
```

Запаковка в архив файлов директории

Параметры:

- `path (str)` – путь к директории
- `zip_name (str)` – имя архива (без .zip)
- `args` – параметры `shutil.make_archive`
- `kwargs` – параметры `shutil.make_archive`

Тип результата:

`str`

Результат:

имя архива

```
static unzip(filename, extract_path, *args, **kwargs)
```

Извлечение файлов из архива

Параметры:

- `filename (str)` – имя архива
- `extract_path (str)` – путь к папке назначения распаковки
- `args` – параметры `shutil.make_archive`
- `kwargs` – параметры `shutil.make_archive`

Тип результата:

`None`

Результат:

None

```
static size_of_file(file)
```

Размер файла в байтах

Параметры:

`file (str)` – путь к файлу

Тип результата:

`int`

Результат:

размер файла в байтах

```
static get_filename_from_path(path)
```

Извлечение файла из пути

Параметры:

`path (str)` – путь к файлу

Тип результата:

`str`

Результат:

имя файла

Example:

```
get_filename_from_path(„/hello/world/file.txt“) -> „file.txt“
```

static `get_files(path)`

Получение списка файлов и папок директории. Не рекурсивное

Параметры:

`path (str)` – путь к директории

Тип результата:

`List [str]`

Результат:

список файлов/папок с относительными путями

Example:

```
get_files('.') -> ['./.idea', './.git', './setup.py', './.gitignore', ...]
```

```
get_files('/home/Desktop') -> ['/home/p/file.drawio', '/home/Desktop/folder', ...]
```

static `get_files_recursive(path)`

Получение списка файлов и папок директории. Рекурсивное

Параметры:

`path (str)` – путь к директории

Тип результата:

`List [str]`

Результат:

список файлов/папок с относительными путями

Example:

```
get_files('.') -> ['./.idea/.gitignore', './.idea/vcs.xml', './.idea/sonarlint/
securityhotspotstore/index.pb', ...] get_files('/home') -> ['/home/
readable_serial.js', '/home/node_modules/async/async/lib/state.js', ...]
```