
Trassir Script Framework Documentation

Выпуск 0.3a

AATrubilin

июл. 26, 2019

1	trassir_script_framework	3
1.1	BaseUtils	3
1.2	HTTPRequester	13
1.3	PokaYoke	16
1.4	ScriptObject	19
1.5	GUITemplate	20
1.6	TrObject	22
1.7	ShotSaver	37
1.8	VideoExporter	40
1.9	Senders	41
2	Changelog	53
2.1	Changelog	53
3	Indices and tables	57
	Алфавитный указатель	59

Фреймворк для скриптов автоматизации Trassir.

Для использования фреймворка скопируйте исходный код в свой скрипт

Примеры использования фреймворка доступны по [ссылке](#)

Дополнительная информация о скриптах доступна в [документации](#)

1.1 BaseUtils

1.1.1 run_as_thread

`static BaseUtils.run_as_thread(fn)`

Декоратор для запуска функций в отдельном потоке.

Результат Функция в отдельном потоке

Тип результата `threading.Thread`

Examples

```
>>> import time
>>>
>>>
>>> @BaseUtils.run_as_thread
>>> def run_count_timer():
...     time.sleep(1)
...     host.stats()["run_count"] += 1
>>>
>>>
>>> run_count_timer()
```

1.1.2 win_encode_path

`static BaseUtils.win_encode_path(path)`

Изменяет кодировку на "cp1251" для WinOS.

Параметры `path` (`str`) – Путь до файла или папки

Результат Декодированный путь до файла или папки

Тип результата `str`

Examples

```
>>> path = r"D:/Shots/Скриншот.jpeg"
>>> os.path.isfile(path)
False
>>> os.path.isfile(BaseUtils.win_encode_path(path))
True
```

1.1.3 is_file_exists

`static BaseUtils.is_file_exists(file_path, tries=1)`

Проверяет, существует ли файл.

Проверка происходит в течении `tries` секунд.

Предупреждение:

Запускайте функцию только в отдельном потоке если `tries > 1`
Вторая и последующие проверки производятся с `time.sleep(1)`

Параметры

- `file_path` (`str`) – Полный путь до файла
- `tries` (`int`, optional) – Количество проверок. По умолчанию `tries=1`

Результат `True` if file exists, `False` otherwise

Тип результата `bool`

Examples

```
>>> BaseUtils.is_file_exists("_t1server.settings")
True
```

1.1.4 is_folder_exists

`static BaseUtils.is_folder_exists(folder)`

Проверяет существование папки и доступ на запись.

Параметры `folder` (`str`) – Путь к папке.

Raises `IOError` – Если папка не существует

Examples

```
>>> BaseUtils.is_folder_exists("/test_path")
IOError: Folder '/test_path' is not exists
```


1.1.5 is_template_exists

`classmethod BaseUtils.is_template_exists(template_name)`

Проверяет существование шаблона

Параметры `template_name` (`str`) – Имя шаблона

Результат `True` если шаблон существует, иначе `False`

Тип результата `bool`

1.1.6 cat

`classmethod BaseUtils.cat(filepath, check_ext=True)`

Выводит на отображение текстовую информацию.

Совет:

- *WinOS*: открывает файл программой по умолчанию
 - *TrassirOS*: открывает файл в терминале с помощью утилиты *cat*
-

Примечание:

Доступные расширения файлов: `[".txt", ".csv", ".log"]`

Если открываете файл с другим расширением установите `check_ext=False`

Параметры

- `filepath` (`str`) – Полный путь до файла
- `check_ext` (`bool`, optional) – Если `True` - проверяет расширение файла.
По умолчанию `True`

Examples

```
>>> BaseUtils.cat("/home/trassir/ Trassir 3 License.txt")
```

```

x bash
LICENSE_STANDALONE      1
LICENSE_T_AGESEX        5
LICENSE_T_ANALYTICS     5
LICENSE_T_FACESEARCH    5
LICENSE_T_FACE_DETECTOR 5
LICENSE_T_FACE_RECOGNIZER 10

Signature:

jm+0IJKH/rM+wFsAvMEZz51a10hMqJn/H1wvqa6Bc/y8Hpduf2AqLs91+ychFT7z
3mUURy1D/01jWt7hgerekXq1wp53AckEadhFQ0LLat9gkZsYTYBMbIBWVJYiL9M5
dm+5AIjnyHnzjnt6XaLZjYmKmnMIHJcIYafhadu7urid1WU1rVqld6W0vVQe9Z6
diwqu4u0pWx0YJDa8S6+8N1QSBGy7tsfs7tD3ZNGRYPc6MGm1KTFzHKuilsj0fK
LSXa0Kks05Kbv0ZZG6S043zL2uRohPLLbpzfZw0QJ8GD28I1fBrN3BAxwobznvtJ
5mkSAFn1L19ws20ja0dsFW5HTufebKVQtJa1qtYUyCQjAmCLV1BxpgTYFU+7S6Xs
Bvh3hUSa+7WdKsvxzELmLJXcTyeUj4D8EsfbAIWiVFNKDclwG5CsNy66QdZxf7HU
frtaUAcpmfauXl694YBfTDQqX1x7Gd12dMQZK4j3S6rd5yATgGboCg9kItUvA2Ds
3MJ0mrI8jNF0LjM3hdw1a2yCFxN8bUJX8WcWD5SuHLi1HarLdpAIR4h1t/JJfxUu
7mxAXD01UTo642JkLWZydbRuiq1QDyt00ZNYCCsq8R0W2pUSvjMPksE6JI12KPwdD
2aLE4CGH+Xp+ZjsdYTW0cL3xg+fLNFZDLC32giSTUSR9osHTIVvi91uNaQjW4tI
x1i1spLkMDIAi9agHM/p30iDoi6Yc100LqGhrdfi6/bgGRC1TYGVV4hKLZefcR3
E21A53wuy2WqG/D4CXf92uDzKL3k6it6xYQRIY9XZorZ5ZT95fa3JATbZGsBeRRc
SMbxvnQ6opTgV1nrBvkukR30GVrF26V8umcy0ILwnwEM0ufHFIGQXn7LZy/wDAHR
8ctPg0IeeDvobjE05P8v2PakZuHpgE0+0igP×9ma6hekchj1a13CdbxJeaVom1NB
1SwL4orU9FVWY80B7A//yF5W0qrJI+JAY+Kzn2rxKXU7J5N1s3HwHXsQpQavdj1G
Nm9q/gFzFLYXcyJ3nqPVigZIZq2Rig6oi062o7GP+QBQNDjwmU71i8L3en9JPRF
mkJbJu19m6SeRiEdURC1Wbn6DEkRdHzklw+Qjk4R39RsDNkFNqW5XjJWMP+01KP1
7cbVGfW2g4YK6ighWuiYwdAVSepTboDE3ABX7mdzES9Zmn+HrtGGtryB8viXs/t3
sss1iIr9IEKXGm/BGNMyJccu4j0z7iNhKb3/cKu52MSUSU39qAJv250fjjgus8j/
m9RJ1+6EBLrx1+Cd2K4Gat7vdaBEtn20L8/WdvKdIsZiToiukWY0+rQ6E3a5QfDA
Cpx/ER+Bz9Icmpiz4wB7Q==

Press any key to exit

```

Raises TypeError – Если `check_ext=True` расширение файла нет в списке `_TEXT_FILE_EXTENSIONS`

1.1.7 to_json

`classmethod BaseUtils.to_json(data, **kwargs)`

Сериализация объекта в JSON строку

Примечание: Не вызывает ошибку при сериализации объектов `datetime`, `date`, `SE_Settings`, `SE_Object`

Параметры `data (obj)` – Объект для сериализации

Результат JSON строка

Тип результата `str`

Examples

```

>>> obj = {"now": datetime.now()}
>>> json.dumps(obj)
TypeError: datetime.datetime(2019, 4, 2, 18, 01, 33, 881000) is not JSON_
↳serializable
>>> BaseUtils.to_json(obj, indent=None)
'{"now": "2019-04-02T18:01:33.881000"}'

```

1.1.8 ts_to_dt

`static BaseUtils.ts_to_dt(ts)`

Конвертирует timestamp в `datetime` объект

Параметры `ts` (`int`) – Timestamp

Результат Datetime объект

Тип результата `datetime`

Examples

```
>>> BaseUtils.ts_to_dt(1564109694242000)
datetime.datetime(2019, 7, 26, 9, 54, 54, 242000)
```

1.1.9 dt_to_ts

`static BaseUtils.dt_to_ts(dt)`

Конвертирует `datetime` объект в trassir timestamp

Параметры `dt` (`datetime`) – Datetime

Результат Trassir timestamp

Тип результата `int`

Examples

```
>>> BaseUtils.ts_to_dt(datetime(2019, 7, 26, 9, 54, 54, 242000))
1564109694242000
```

1.1.10 lpr_flags_decode

`classmethod BaseUtils.lpr_flags_decode(flags)`

Преобразует флаги события AutoTrassir

Приводит флаги события человекочитаемый список

Примечание: Список доступных флагов:

- LPR_UP - Направление движения вверх
- LPR_DOWN - Направление движения вниз
- LPR_BLACKLIST - Номер в черном списке
- LPR_WHITELIST - Номер в черном списке
- LPR_INFO - Номер в информационном списке
- LPR_FIRST_LANE - Автомобиль движется по первой полосе
- LPR_SECOND_LANE - Автомобиль движется по второй полосе
- LPR_THIRD_LANE - Автомобиль движется по третьей полосе

- LPR_EXT_DB_ERROR - Ошибка во внешнем списке
 - LPR_CORRECTED - Номер исправлен оператором
-

Параметры `flags` (`int`) – Биты LPR события. Как правило аргумент `ev.flags` события `SE_LprEvent` `AutoTrassir`. Например 536870917

Результат Список флагов

Тип результата `List[str]`

Examples

```
>>> BaseUtils.lpr_flags_decode(536870917)
['LPR_UP', 'LPR_BLACKLIST']
```

1.1.11 event_type_encode

`classmethod BaseUtils.event_type_encode(event_type)`

Преобразует тип события `str` -> `int`

Примечание: События в БД хранятся в `int`, в скриптах приходят в человекочитаемом, строковом формате.

Параметры `event_type` (`str`) – Тип события как в скриптах.

Examples

```
>>> BaseUtils.event_type_encode("Border Crossed A -> B")
-2010220362
```

Результат Тип события как в БД

Тип результата `int`

1.1.12 event_type_decode

`classmethod BaseUtils.event_type_decode(event_type)`

Преобразует тип события `int` -> `str`

Примечание: События в БД хранятся в `int`, в скриптах приходят в человекочитаемом, строковом формате.

Параметры `event_type` (`int`) – Тип события как в БД.

Examples

```
>>> BaseUtils.event_type_encode(-2010220362)
"Border Crossed A -> B"
```

Результат Тип события как в скриптах

Тип результата `str`

1.1.13 image_to_base64

classmethod BaseUtils.image_to_base64(*image*)

Создает base64 из изображения

Параметры *image* (`str`) – Путь к изображению или изображение

Результат Base64 image

Тип результата `str`

Examples

```
>>> BaseUtils.image_to_base64(r"manual/en/cloud-devices-16.png")
'iVBORw0KGgoAAAANSUhEUgAAB1MAAAHOCAYAAABo5wRhAAAAACXBIXMAAC4jA...'
>>> BaseUtils.image_to_base64(open(r"manual/en/cloud-devices-16.png", "rb").read())
'iVBORw0KGgoAAAANSUhEUgAAB1MAAAHOCAYAAABo5wRhAAAAACXBIXMAAC4jA...'
```

1.1.14 base64_to_html_img

classmethod BaseUtils.base64_to_html_img(*image_base64*, ***kwargs*)

Возвращает base64 изображение в `` html теге

Параметры

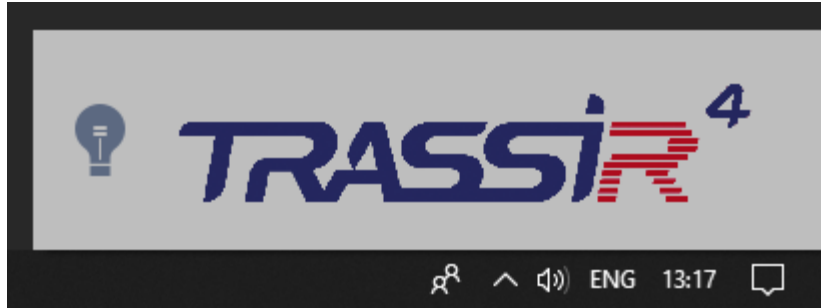
- *image_base64* (`str`) – Base64 image
- ***kwargs* – HTML `` tag attributes. Подробнее на html.com

Результат html image

Тип результата `str`

Examples

```
>>> base64_image = BaseUtils.image_to_base64(r"manual/en/cloud-devices-16.png")
>>> html_image = BaseUtils.base64_to_html_img(base64_image, width=280, height=75)
>>> html_image
''
>>> host.message(html_image)
```



1.1.15 save_pkl

`static BaseUtils.save_pkl(file_path, data)`

Сохраняет данные в *.pkl* файл

Параметры

- `file_path (str)` – Путь до файла
- `data` – Данные для сохранения

Результат Абсолютный путь до файла

Тип результата `str`

Examples

```
>>> data = {"key": "value"}
>>> BaseUtils.save_pkl("saved_data.pkl", data)
'D:\DSSL\Trassir-4.1-Client\saved_data.pkl'
```

1.1.16 load_pkl

`static BaseUtils.load_pkl(file_path, default_type=<type 'dict'>)`

Загружает данные из *.pkl* файла

Параметры

- `file_path (str)` – Путь до файла
- `default_type (optional)` – Тип данных, возвращаемый при неудачной загрузке данных из файла. По умолчанию `dict`

Результат Данные из файла или `default_type()`

Examples

```
>>> BaseUtils.load_pkl("fake_saved_data.pkl")
{}
>>> BaseUtils.load_pkl("fake_saved_data.pkl", default_type=list)
[]
>>> BaseUtils.load_pkl("fake_saved_data.pkl", default_type=int)
0
```

(continues on next page)

(продолжение с предыдущей страницы)

```
>>> BaseUtils.load_pkl("fake_saved_data.pkl", default_type=str)
''
>>> BaseUtils.load_pkl("saved_data.pkl")
{'key': 'value'}
```

1.1.17 get_operator_gui

classmethod BaseUtils.get_operator_gui()

Возвращает объект интерфейса оператора

Результат Объект интерфейса оператора

Тип результата OperatorGUI

Raises ScriptError – Если не удастся загрузить интерфейс

Examples

Открыть интерфейс Trassir а мониторе №1

```
>>> operator_gui = BaseUtils.get_operator_gui()
>>> operator_gui.raise_monitor(1)
```

1.1.18 get_server_guid

classmethod BaseUtils.get_server_guid()

Возвращает guid текущего сервера

Результат Guid сервера

Тип результата str

Examples

```
>>> BaseUtils.get_server_guid()
'client'
```

1.1.19 get_script_name

classmethod BaseUtils.get_script_name()

Возвращает имя текущего скрипта

Результат Имя скрипта

Тип результата str

Examples

```
>>> BaseUtils.get_script_name()
'Новый скрипт'
```

1.1.20 get_screenshot_folder

classmethod BaseUtils.get_screenshot_folder()

Возвращает путь до папки скриншотов

При этом производит проверку папки методом *BaseUtils.is_folder_exists()*

Результат Полный путь к папке скриншотов

Тип результата *str*

Examples

```
>>> BaseUtils.get_screenshot_folder()
'/home/trassir/shots'
```

1.1.21 get_logger

classmethod BaseUtils.get_logger(*host_log='WARNING', popup_log='ERROR',*
file_log=None, file_name=None,
file_max_bytes=5242880, file_backup_count=2)

Возвращает логгер с предустановленными хэндлерами

Доступные хэндлеры:

- *host_log*: Пишет сообщения в основной лог сервера *_tlserver.log*
- *popup_log*: Показывает всплывающие сообщения *message/alert/error*
- *file_log*: Пишет сообщения в отдельный файл в папку скриншотов

Для каждого хэндлера можно установить разный уровень логирования

По умолчанию *host_log="WARNING"* и *popup_log="ERROR"*

Примечание: Имя файла лога можно указать с расширением «.log» или без.

См.также:

[Logging levels](https://docs.python.org) на сайте docs.python.org

Параметры

- *host_log* (*str*, optional) – Уровень логирования в основной лог. По умолчанию "WARNING"
- *popup_log* (*str*, optional) – Уровень логирования во всплывающих сообщениях. По умолчанию "ERROR"
- *file_log* (*str*, optional) – Уровень логирования в отдельный файл По умолчанию *None*
- *file_name* (*str*, optional) – Имя файла для логирования. По умолчанию *None* и равно <имени скрипта>.log
- *file_max_bytes* (*int*, optional) – Максимальный размер файла лога в байтах. По умолчанию 5 * 1024 * 1024

- `file_backup_count` (`int`, optional) – Максимальное кол-во бэкапов лога.
По умолчанию 2

Результат `Логгер`

Тип результата `logging.logger`

Examples

```
>>> logger = BaseUtils.get_logger()
>>> logger.warning("My warning message")
>>> try:
...     # noinspection PyUnresolvedReferences
...     do_something()
... except NameError:
...     logger.error("Function is not defined", exc_info=True)
```

1.1.22 set_script_name

`classmethod BaseUtils.set_script_name(fmt=None, script_name=None)`

Автоматически изменяет имя скрипта

Новое имя скрипта создается на основе параметров скрипта. По желанию можно изменить шаблон имени. По умолчанию `"{title} v{version}"`

Примечание: Имя изменяется только если сейчас у скрипта стандартное имя, например "Новый скрипт" или "Unnamed Script" и др.

Параметры

- `fmt` (`str`, optional) – Шаблон имени скрипта. По умолчанию `None`
- `script_name` (`str`, optional) – Имя скрипта. Если не задано - парсит имя из параметров. По умолчанию `None`

Examples

```
>>> BaseUtils.set_script_name()
'trassir_script_framework v0.4'
```

```
>>> BaseUtils.set_script_name(fmt="{title}")
'trassir_script_framework'
```

1.2 HTTPRequester

`class HTTPRequester(opener=None, timeout=10)`

Framework for urllib2

См.также:

https://docs.python.org/2/library/urllib2.html#urllib2.build_opener

Параметры

- `opener` (`urllib2.OpenerDirector`, optional) – Обработчик запросов. По умолчанию `None`
- `timeout` (`int`, optional) – Время ожидания запроса, в секундах. По умолчанию `timeout=10`

Examples

Пример запроса к SDK Trassir

```
>>> # Отключение проверки сертификата
>>> context = ssl.create_default_context()
>>> context.check_hostname = False
>>> context.verify_mode = ssl.CERT_NONE
>>>
>>> handler = urllib2.HTTPSHandler(context=context)
>>> opener = urllib2.build_opener(handler)
>>>
>>> requests = HTTPRequester(opener, timeout=20)
>>> response = requests.get(
...     "https://172.20.0.101:8080/login",
...     params={"username": "Admin", "password": "12345"}
... )
>>>
>>> response.code
200
>>> response.text
'{\n  "sid" : "T6LAAcxg",\n  "success" : 1\n}\n'
>>> response.json
{'success': 1, u'sid': u'T6LAAcxg'}
```

```
class Response(*args)
```

Класс ответа от сервера

`code`

Код ответа сервера

Type `str` | `int`

`text`

Текст ответа

Type `str`

`json`

Создает объект из json ответа

Type `dict` | `list`

```
get(url, params=None, headers=None)
```

Создает GET запрос по указанному url

Параметры

- `url` (`str`) – Url для запроса
- `params` (`dict`, optional) – Параметры GET запроса
- `headers` (`dict`, optional) – Заголовки запроса

Examples

```
>>> requests = HTTPRequester()
>>> response = requests.get(
...     "http://httpbin.org/get",
...     params={"PARAMETER": "TEST"},
... )
>>> response.code
200
>>> response.text
'{"args": {"PARAMETER": "TEST"}, ...}'
>>> response.json
{'args': {'PARAMETER': 'TEST'}, ...}'
```

Результат Response instance

Тип результата *HTTPRequester.Response*

`post(url, data=None, headers=None)`

Создает POST запрос по указанному url

Параметры

- `url` (`str`) – Url для запроса
- `data` (`dict`, optional) – Данные POST запроса
- `headers` (`dict`, optional) – Заголовки запроса

Examples

```
>>> requests = HTTPRequester()
>>> response = requests.post(
...     "http://httpbin.org/post",
...     data={"PARAMETER": "TEST"},
...     headers={"Content-Type": "application/json"},
... )
>>> response.code
200
>>> response.text
'{"args": {"PARAMETER": "TEST"}, ...}'
>>> response.json
{'args': {'PARAMETER': 'TEST'}, ...}'
```

Результат Response instance

Тип результата *HTTPRequester.Response*

1.2.1 Response

`class Response(*args)`

Класс ответа от сервера

`code`

Код ответа сервера

Тип `str | int`

`text`

Текст ответа

Тип `str`

`json`
Создает объект из `json` ответа
Type `dict` | `list`

1.3 PokaYoke

`class PokaYoke`

Класс для защиты от дурака

Позволяет блокировать запуск скрипта на ПО, где это не предусмотрено (например, на клиенте или TOS). А также производить некоторые другие проверки.

1.3.1 ban_tos

`static PokaYoke.ban_tos()`

Блокирует запуск скрипта на *Trassir OS*

Raises `OSError` – Если скрипт запускается на *Trassir OS*

Examples

```
>>> PokaYoke.ban_tos()
OSError: Скрипт недоступен для TrassirOS
```

1.3.2 ban_win

`static PokaYoke.ban_win()`

Блокирует запуск скрипта на *Windows OS*

Raises `OSError` – Если скрипт запускается на *Windows OS*

Examples

```
>>> PokaYoke.ban_win()
OSError: Скрипт недоступен для WindowsOS
```

1.3.3 ban_client

`static PokaYoke.ban_client()`

Блокирует запуск скрипта на *Trassir Client*

Raises `TrassirError` – Если скрипт запускается на *Trassir Client*

Examples

```
>>> PokaYoke.ban_client()
TrassirError: Скрипт недоступен для клиентской версии Trassir
```

1.3.4 ban_daemon

`classmethod PokaYoke.ban_daemon()`

Блокирует запуск скрипта на сервере Trassir, который запущен как служба

Raises `TrassirError` – Если скрипт запускается на сервере Trassir, который запущен как служба

Examples

```
>>> PokaYoke.ban_daemon()
TrassirError: Скрипт недоступен для Trassir запущенным как служба
```

1.3.5 check_email_account

`static PokaYoke.check_email_account(account_name)`

Проверяет существование E-Mail аккаунта

Параметры `account_name` (`str`) – Имя E-Mail аккаунта

Результат Список объектов

Тип результата `List[TrObject]`

Raises

- `ParameterError` – Если аккаунт не выбран
- `ObjectsNotFoundError` – Если аккаунт не найден

Examples

```
>>> PokaYoke.check_email_account("")
ParameterError: 'EmailAccount' не выбраны
>>> PokaYoke.check_email_account("YourAccount")
ObjectsNotFoundError: Не найдены объекты EmailAccount: YourAccount
>>> PokaYoke.check_email_account("MyAccount")
[TrObject('MyAccount')]
```

1.3.6 parse_emails

`classmethod PokaYoke.parse_emails(mailing_list, regex=None)`

Парсит email дреса из строки

Каждый email проверяется с помощью regex `r"[^@]+@[^@]+\.[^@]+"`.

Параметры

- `mailing_list` (`str`) – Список email адресов, разделенный запятыми
- `regex` (`SRE_Pattern`, optional) – Новый regex шаблон для проверки. По умолчанию `None`

Результат Список адресов

Тип результата `List[str]`

Raises `ParameterError` – Если найден невалидный email

Examples

```
>>> PokaYoke.parse_emails("a.trubilil!dssl.ru,support@dssl.ru")
ParameterError: Email 'a.trubilil!dssl.ru' is not valid!
>>>
>>> PokaYoke.parse_emails("a.trubilil@dssl.ru,support@dssl.ru")
['a.trubilil@dssl.ru', 'support@dssl.ru']
```

1.3.7 check_phones

`classmethod PokaYoke.check_phones(phones, regex=None)`

Проверяет строку на валидность телефонных номеров

Строка проверяется с помощью regex `r"[^\d,;]"`.

Параметры

- `phones` (`str`) – Список телефонов, разделенный запятыми или точкой с запятой
- `regex` (`SRE_Pattern`, optional) – Новый regex шаблон для проверки. По умолчанию `None`

Результат Список номеров телефона

Тип результата `str`

Raises `ParameterError` – Если найден невалидный номер телефона

Examples

```
>>> PokaYoke.check_phones("79999999999,78888888888A")
ParameterError: Bad chars in phone list: `A`
>>>
>>> PokaYoke.check_phones("a.trubilil@dssl.ru,support@dssl.ru")
'79999999999,78888888888'
```

1.3.8 fire_recognizer_events

`classmethod PokaYoke.fire_recognizer_events(enable=True, server_guid=None)`

Проверяет «Режим для СКУД» настроек распознавания лиц.

По умолчанию проверяет активирован ли «Режим для СКУД» на сервере, где запущен скрипт. По желанию можно указать удаленный сервер для проверки.

Параметры

- `enable` (`bool`, optional) – Состояние параметра. По умолчанию `True`.
- `server_guid` (`str`, optional) – Guid сервера. По умолчанию `None`.

Raises

- `RuntimeError` – Если указанный сервер недоступен.

- `EnvironmentError` – Если модуль распознавания или режим для СКУД не доступны.
- `TrassirError` – Если текущее состояние не соответствует необходимому.

Examples

```
>>> PokaYoke.fire_recognizer_events()
TrassirError: Пожалуйста, активируйте 'Режим для СКУД' в настройках распознавания_
↳лиц
```

1.4 ScriptObject

```
class ScriptObject(name=None, guid=None, parent=None, host_api=<module 'host'
                        from '/home/docs/checkouts/readthedocs.org/user_builds/trassir-
                        script-framework/checkouts/stable/tests/host/__/init__.pyc'>)
    Создает объект для генерации событий
```

Параметры

- `name` (`str`, optional) – Имя объекта. По умолчанию `None`
- `guid` (`str`, optional) – Guid объекта. По умолчанию `None`
- `parent` (`str`, optional) – Guid родительского объекта. По умолчанию `None`

Примечание:

- Имя объекта по умолчанию - `BaseUtils.get_script_name()`
- Guid объекта по умолчанию строится по шаблону `"{script_guid}_object"`
- Guid родительского объекта по умолчанию - `BaseUtils.get_server_guid()`

Examples

```
>>> # Создаем объект
>>> scr_obj = ScriptObject()
```

```
>>> # Проверяем текущее состояние объекта
>>> scr_obj.health
'OK'
```

```
>>> # Установить флаг возле объекта
>>> scr_obj.check_me = True
```

```
>>> # Сгенерировать событие с текстом
>>> scr_obj.fire_event_v2("New event")
```

`health`
Состояние объекта
Type "OK" | "Error"

`check_me`
Флаг `check_me` объекта
Type `bool`

`name`
Имя объекта
Type `str`

`folder`
Папка объекта
Type `str`

`context_menu_button(text, callback)`
Добавляет кнопку в контекстное меню объекта

Параметры

- `text (str)` – Текст кнопки
- `callback (function)` – Функция, которая вызывается при нажатии на кнопку. В качестве единственного аргумента функция принимает текущий объект (`host.object(self.guid)`).

Результат Хендлер контекстного меню
Тип результата `SE_ContextCatcher`

Raises

- `ValueError` – Если пустой текст кнопки.
- `TypeError` – Если `callback` нельзя вызвать в качестве функции.

Examples

```
>>> scr = ScriptObject()
>>>
>>> def switch(obj):
...     check_me = scr.check_me
...     scr.check_me = not check_me
...     btn.set_name("ON" if check_me else "OFF")
>>>
>>> btn = scr.context_menu_button("ON", switch)
>>> btn
<host.SE_ContextCatcher object at 0x17B01A98>
>>> scr.context_menu
[('ON', 'switch', <host.SE_ContextCatcher object at 0x17B01A98>)]
```

`fire_event_v2(message, channel="", data="")`
Создает событие в Trassir

Параметры

- `message (str)` – Сообщение события (p1)
- `channel (str, optional)` – Ассоциированный с событием канал (p2)
- `data (str, optional)` – Дополнительные данные (p3)

1.5 GUITemplate

```
class GUITemplate(template_name, host_api=<module 'host' from
'/home/docs/checkouts/readthedocs.org/user_builds/trassir-script-
framework/checkouts/stable/tests/host/__init__.pyc'>)
    Класс для работы с шаблонами Trassir
```


При инициализации находит существующий шаблон по имени или создает новый.

Примечание: Если вручную создать два или большее шаблона с одинаковыми именами данный класс выберет первый попавшийся шаблон с заданным именем.

Предупреждение: Работа с контентом шаблона может привести к падениям трассира. Используйте данный класс на свой страх и риск!

Совет: Для понимания, как формируется контент отредактируйте любой шаблон вручную и посмотрите что получится в скрытых параметрах трассира (активируются нажатием клавиши F4 в настройках трассира) *Настройки/Шаблоны/<Имя шаблона>/content*

Ниже представлены некоторые примеры шаблонов

- Вывод одного канала `S0tE8nfg_0r3QZu4D` `gui7(DEWARP_SETTINGS,zwVj07w0,dewarp(),1,S0tE8nfg_0r3QZu4D)`
- Вывод шаблона 4x4 с каналами двумя `Kpid6EC0_0r3QZu4D`, `ZRtXLrgu_0r3QZu4D` `gui7(DEWARP_SETTINGS,zwVj07w0,dewarp(),4,Kpid6EC0_0r3QZu4D,ZRtXLrgu_0r3QZu4D,,)`
- Вывод шаблон с минибраузером и ссылкой на <https://www.google.com/> `minibrowser(0,htmltab(,https://www.google.com/))`

Параметры `template_name (str)` – Имя шаблон

Examples

```
>>> # Создаем шаблон с именем "New template" и получаем его guid
>>> template = GUITemplate("New template")
>>> template.guid
'Y2YFAkeZ'
```

```
>>> # Устанавливаем на шаблон минибраузер с ссылкой на google
>>> template.content = "minibrowser(0,htmltab(,https://www.google.com/))"
```

```
>>> # Изменяем имя шаблона на "Google search"
>>> template.name = "Google search"
```

```
>>> # Открываем шаблон на первом мониторе
>>> template.show(1)
```

```
guid
    Guid шаблона
    Type str
```

```
name
    Имя шаблона
    Type str
```

`content`
Контент шаблона
Type `str`

`delete()`
Удаляет шаблон

`show(monitor=1)`
Открывает шаблон на указанном мониторе
Параметры `monitor` (`int`, optional) – Номер монитора. По умолчанию `monitor=1`

1.6 TrObject

`class TrObject(obj, host_api=<module 'host' from '/home/docs/checkouts/readthedocs.org/user_builds/trassir-script-framework/checkouts/stable/tests/host/___init___pyc'>)`
Вспомогательный класс для работы с объектами Trassir

`obj`
Объект `trassir object('{guid}')` или `None`
Type `SE_Object`

`obj_methods`
Список методов объекта `TrObject.obj`
Type `List[str]`

`name`
Имя объекта или его `guid`
Type `str`

`guid`
Guid объекта
Type `str`

`full_guid`
Полный `guid {guid объекта}_{guid сервера}` или `None`
Type `str`

`type`
Тип объекта, например `"RemoteServer"`, `"Channel"`, `"Grabber"`, `"User"`, и др.
Type `str`

`path`
Путь в настройках или `None`
Type `str`

`parent`
Guid родительского объекта или `None`
Type `str`

`server`
Guid сервера или `None`
Type `str`

`settings`
Объект настроек `settings('{path}')` или `None`
Type `SE_Settings`

Raises

- `TypeError` – Если неправильные параметры объекта
- `ValueError` – Если в имени объекта есть запятые

1.6.1 Servers

`class Servers`

Класс для работы с серверами

Examples

```
>>> srvs = Servers()
>>> local_srv = srvs.get_local()
[TrObject('Клиент')]
>>> # Проверим "Здоровье" локального сервера
>>> local_srv[0].obj.state("server_health")
'Health Problem'
```

`get_local()`

Возвращает локальный сервер (на котором запущен скрипт)

Результат Список объектов

Тип результата `List[TrObject]`

`get_remote()`

Возвращает список удаленных серверов

Результат Список объектов

Тип результата `List[TrObject]`

`get_all()`

Возвращает список всех доступных серверов

Результат Список объектов

Тип результата `List[TrObject]`

1.6.2 Persons

`class Persons(server_guid=None)`

Класс для работы с персонами и их папками.

См.также:

Персоны - Руководство пользователя Trassir

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> persons = Persons()
>>> persons.get_folders()
[TrObject('Мошенники'), TrObject('DSSL'), TrObject('persons')]
>>> persons.get_persons()
```

(continues on next page)

(продолжение с предыдущей страницы)

```
[
  {
    'name': 'Leonardo',
    'guid': 'cJuJYAha',
    'gender': 0,
    'birth_date': '1980-01-01',
    'comment': 'Comment',
    'contact_info': 'Contact info',
    'folder_guid': 'n68LOBhG',
    'image': <image, str>,
    'image_guid': 'gBHZ2vpz',
    'effective_rights': 0,
  },
  ...
]
>>> persons.get_person_by_guid("cJuJYAha")
{
  'name': 'Leonardo',
  'guid': 'cJuJYAha',
  'gender': 0,
  'birth_date': '1980-01-01',
  'comment': 'Comment',
  'contact_info': 'Contact info',
  'folder_guid': 'n68LOBhG',
  'image': <image, str>,
  'image_guid': 'gBHZ2vpz',
  'effective_rights': 0,
}
```

`get_folders(names=None)`

Возвращает список папок персон

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_persons(folder_names=None, timeout=10)`

Возвращает список персон

Примечание: Данный метод работает только с локальной БД.

Параметры

- `folder_names` (`str` | `List[str]`, optional) – `str` - названия папок персон, разделенные запятыми или `list` - список папок персон. По умолчанию `None`
- `timeout` (`int`, optional) – Макс. время запроса к БД. По умолчанию `timeout=10`

Результат Список персон - если персоны найдены

Тип результата `List[dict]`

Raises

- `EnvironmentError` – Если произошла ошибка при запросе в БД.
- `TrassirError` – Если в данной сборке Trassir нет метода `host.service_persons_get`

`get_person_by_guid(person_guid, timeout=10)`

Возвращает информацию о персоне по его guid

Примечание: Для уменьшения кол-ва запросов к БД - метод создает локальную копию всех персон при первом запросе и обновляет ее вместе с последующими запросами не чаще чем 1 раз в 10 минут.

Параметры

- `person_guid` (`str`) – Guid персоны
- `timeout` (`int`, optional) – Макс. время запроса к БД. По умолчанию `timeout=10`

Результат Данные о персоне или `None` если персона не найдена

Тип результата `dict`

```
get_person_by_name(person_name, timeout=10)
```

Возвращает информацию о персоне по его имени

Примечание: Для уменьшения кол-ва запросов к БД - метод создает локальную копию всех персон при первом запросе и обновляет ее вместе с последующими запросами не чаще чем 1 раз в 10 минут.

Параметры

- `person_name` (`str`) – Имя персоны
- `timeout` (`int`, optional) – Макс. время запроса к БД. По умолчанию `timeout=10`

Результат Данные о персоне или `None` если персона не найдена

Тип результата `dict`

1.6.3 Users

```
class Users(server_guid=None)
```

Класс для работы с пользователями и их группами.

См.также:

Пользователи - Руководство пользователя Trassir

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> users = Users()
>>> users.get_groups()
[TrObject('TEST')]
```

```
get_groups(names=None)
```

Возвращает список групп пользователей

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

```
get_users(names=None)
```

Возвращает список пользователей

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

```
get_users_by_groups(group_names)
```

Возвращает список пользователей из указанных групп

Параметры `group_names` (`str` | `list`) – `str` - имена групп, разделенные запятыми или `list` - список имен.

Результат Список объектов

Тип результата `List[TrObject]`

1.6.4 Templates

```
class Templates(server_guid=None)
```

Класс для работы с существующими шаблонами.

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> templates = Templates(BaseUtils.get_server_guid())
>>> templates.get_all()
[TrObject('Parking'), TrObject('FR'), TrObject('AT'), TrObject('AD+')]
```

```
get_all(names=None)
```

Возвращает список шаблонов

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

1.6.5 PosTerminals

```
class PosTerminals(server_guid=None)
```

Класс для работы с POS Терминалами

См.также:

[Настройка POS-терминалов - Руководство пользователя Trassir](#)

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> pos_terminals = PosTerminals()
>>> pos_terminals.get_disabled()
[TrObject('Касса (1)')]
```

```
get_enabled(names=None)
    Возвращает список активных POS Терминалов
    Параметры names (str | list, optional) – str - имена, разделенные запя-
        тыми или list - список имен. По умолчанию None
    Результат Список объектов
    Тип результата List[TrObject]

get_disabled(names=None)
    Возвращает список неактивных POS Терминалов
    Параметры names (str | list, optional) – str - имена, разделенные запя-
        тыми или list - список имен. По умолчанию None
    Результат Список объектов
    Тип результата List[TrObject]

get_all(names=None)
    Возвращает список всех POS Терминалов
    Параметры names (str | list, optional) – str - имена, разделенные запя-
        тыми или list - список имен. По умолчанию None
    Результат Список объектов
    Тип результата List[TrObject]
```

1.6.6 Devices

```
class Devices(server_guid=None)
    Класс для работы с ip устройствами
```

См.также:

IP-устройства - Руководство пользователя Trassir

Параметры server_guid (**str** | List[**str**], optional) – Guid сервера или список guid. По умолчанию None, что соответствует всем доступным серверам.

Examples

```
>>> devices = Devices()
>>> enabled_devices = devices.get_enabled()
>>> enabled_devices
[TrObject('AC-D2121IR3W'), TrObject('AC-D5123IR32'), ...]
>>>
>>> # Перезагрузим все устройства
>>> for dev in enabled_devices:
...     dev.settings["reboot"] = 1
```

```
get_enabled(names=None)
    Возвращает список активных устройств
    Параметры names (str | list, optional) – str - имена, разделенные запя-
        тыми или list - список имен. По умолчанию None
    Результат Список объектов
    Тип результата List[TrObject]

get_disabled(names=None)
    Возвращает список неактивных устройств
    Параметры names (str | list, optional) – str - имена, разделенные запя-
        тыми или list - список имен. По умолчанию None
```

Результат Список объектов

Тип результата `List[TrObject]`

`get_all(names=None)`

Возвращает список всех устройств

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

1.6.7 Channels

`class Channels(server_guid=None)`

Класс для работы с каналами

См.также:

Каналы - Руководство пользователя Trassir

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> channels = Channels()
>>> selected_channels = channels.get_enabled("AC-D2121IR3W 2,AC-D9141IR2 1")
>>> selected_channels
[TrObject('AC-D2121IR3W 2'), TrObject('AC-D9141IR2 1')]
>>>
>>> # Включим ручную запись на выбранные каналы
>>> for channel in selected_channels:
...     channel.obj.manual_record_start()
>>>
>>> # Или добавим к имени канала его guid
>>> for channel in selected_channels:
...     channel.settings["name"] += " ({}).format(channel.guid)
```

`get_enabled(names=None)`

Возвращает список активных каналов

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_disabled(names=None)`

Возвращает список неактивных каналов

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_all(names=None)`

Возвращает список всех каналов

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

1.6.8 NetworkNodes

`class NetworkNodes(server_guid=None)`

Класс для работы с сетевыми подключениями

См.также:

Сеть - Руководство пользователя Trassir

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> network_nodes = NetworkNodes("client")
>>> network_nodes.get_enabled()
[TrObject('QuattroStationPro (172.20.0.101)'), TrObject('NSK-HD-01 (127.0.0.1)')]
```

`get_enabled(names=None)`

Возвращает список активных сетевых подключений

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_disabled(names=None)`

Возвращает список неактивных сетевых подключений

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_all(names=None)`

Возвращает список всех сетевых подключений

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

1.6.9 Rules

`class Rules(server_guid=None)`

Класс для работы с правилами

См.также:

Правила - Руководство пользователя Trassir

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> rules = Rules()
>>> all_rules = rules.get_all()
>>> all_rules
[TrObject('!Rule'), TrObject('NEW RULE'), TrObject('Новое правило')]
>>>
>>> # Отключим все правила
>>> for rule in all_rules:
...     rule.settings["enable"] = 0
```

`get_enabled(names=None)`

Возвращает список активных правил

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_disabled(names=None)`

Возвращает список неактивных правил

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_all(names=None)`

Возвращает список всех правил

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

1.6.10 Scripts

`class Scripts(server_guid=None)`

Класс для работы со скриптами

См.также:

Скрипты - Руководство пользователя Trassir

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> scripts = Scripts()
>>> all_scripts = scripts.get_all()
>>> all_scripts
[TrObject('Новый скрипт'), TrObject('HDD Health Monitor'), TrObject('Password_
↳Reminder')]
>>>
>>> # Отключим все скрипты
>>> for script in all_scripts:
...     script.settings["enable"] = 0
```

`get_enabled(names=None)`
 Возвращает список активных скриптов
Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`
Результат Список объектов
Тип результата `List[TrObject]`

`get_disabled(names=None)`
 Возвращает список неактивных скриптов
Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`
Результат Список объектов
Тип результата `List[TrObject]`

`get_all(names=None)`
 Возвращает список всех скриптов
Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`
Результат Список объектов
Тип результата `List[TrObject]`

1.6.11 StockScripts

`class StockScripts(server_guid=None)`
 Класс для работы со встроенными скриптами
Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> stock_scripts = StockScripts()
>>> all_scripts = stock_scripts.get_all()
>>> all_scripts
[TrObject('MegaRAID Monitor')]
>>>
>>> # Отключим все скрипты
>>> for script in all_scripts:
...     script.settings["enable"] = 0
```

`get_enabled(names=None)`
 Возвращает список активных скриптов
Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`
Результат Список объектов
Тип результата `List[TrObject]`

`get_disabled(names=None)`
 Возвращает список неактивных скриптов
Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`
Результат Список объектов
Тип результата `List[TrObject]`

```
get_all(names=None)
```

Возвращает список всех скриптов

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

1.6.12 Schedules

```
class Schedules(server_guid=None)
```

Класс для работы с расписаниями

См.также:

Расписания - Руководство пользователя Trassir

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> schedules = Schedules()
>>> my_schedule = schedules.get_enabled("!Schedule")[0]
>>> my_schedule.obj.state("color")
'Red'
```

```
on_load(schedule_name, callback, tries=5)
```

Вызывает `callback` после загрузки расписания

Примечание: При загрузке сервера, объект расписания становится не сразу доступен. Данный метод помогает предотвратить данную ошибку.

Параметры

- `schedule_name` (`str`) – Имя расписания
- `callback` (`function`) – Функция, которая вызывается после загрузки расписания.
- `tries` (`int`, optional) – Кол-во попыток загрузки расписания. Каждая попытка производится с интервалом 1 с. По умолчанию 5

Examples

```
>>> schedule = None
>>> # noinspection PyGlobalUndefined,PyUnresolvedReferences
>>> def on_schedule_loaded(schedule_obj):
...     global schedule
...     schedule = schedule_obj
...
...     message("Schedule '{obj.name}' ({obj.guid}) loaded".
↵format(obj=schedule))
...     schedule.activate_on_state_changes(lambda: alert(schedule.state(
↵"color"))))
>>>
>>> Schedules().on_load("Unnamed Schedule", on_schedule_loaded)
```

`get_enabled(names=None)`
 Возвращает список активных расписаний
Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`
Результат Список объектов
Тип результата `List[TrObject]`

`get_disabled(names=None)`
 Возвращает список неактивных расписаний
Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`
Результат Список объектов
Тип результата `List[TrObject]`

`get_all(names=None)`
 Возвращает список всех расписаний
Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`
Результат Список объектов
Тип результата `List[TrObject]`

1.6.13 EmailAccounts

`class EmailAccounts(server_guid=None)`
 Класс для работы с E-Mail аккаунтами

См.также:

Добавление учетной записи e-mail - Руководство пользователя Trassir

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> email_accounts = EmailAccounts()
>>> email_accounts.get_all()
[TrObject('Новая учетная запись e-mail'), TrObject('MyAccount')]
```

`get_all(names=None)`
 Возвращает список всех E-Mail аккаунтов
Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`
Результат Список объектов
Тип результата `List[TrObject]`

1.6.14 TemplateLoops

`class TemplateLoops(server_guid=None)`
 Класс для работы с циклическими просмотрами шаблонов

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> template_loops = TemplateLoops()
>>> template_loops.get_all()
[TrObject('Новый циклический просмотр')]
```

`get_enabled(names=None)`

Возвращает список активных циклических просмотров шаблонов

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_disabled(names=None)`

Возвращает список неактивных циклических просмотров шаблонов

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_all(names=None)`

Возвращает список всех циклических просмотров шаблонов

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

1.6.15 GPIO

`class GPIO(server_guid=None)`

Класс для работы с тревожными входами/выходами

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> gpio = GPIO()
>>> gpio_door = gpio.get_inputs("Door")[0]
>>> gpio_door.obj.state("gpio_input_level")
'Input Low (Normal High)'
>>> gpio_light = gpio.get_outputs("Light")[0]
>>> gpio_light.obj.set_output_high()
```

`get_inputs(names=None)`

Возвращает список тревожных входов

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_outputs(names=None)`

Возвращает список тревожных выходов

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов
Тип результата List[*TrObject*]

1.6.16 Zones

`class Zones(server_guid=None)`

Класс для работы с зонами

Параметры `server_guid` (`str` | List[`str`], optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> zones = Zones()
>>> zones.get_queues("Kacca 1")[0].obj.state("zone_queue")
'5+'
```

`get_people(names=None)`

Возвращает список PeopleZones

Параметры `names` (`str` | list, optional) – `str` - имена, разделенные запятыми или list - список имен. По умолчанию `None`

Результат Список объектов

Тип результата List[*TrObject*]

`get_simt(names=None)`

Возвращает список зон SIMT

Параметры `names` (`str` | list, optional) – `str` - имена, разделенные запятыми или list - список имен. По умолчанию `None`

Результат Список объектов

Тип результата List[*TrObject*]

`get_workplaces(names=None)`

Возвращает список рабочих зон

Параметры `names` (`str` | list, optional) – `str` - имена, разделенные запятыми или list - список имен. По умолчанию `None`

Результат Список объектов

Тип результата List[*TrObject*]

`get_queues(names=None)`

Возвращает список зон очередей

Параметры `names` (`str` | list, optional) – `str` - имена, разделенные запятыми или list - список имен. По умолчанию `None`

Результат Список объектов

Тип результата List[*TrObject*]

`get_shelves(names=None)`

Возвращает список зон полок

Параметры `names` (`str` | list, optional) – `str` - имена, разделенные запятыми или list - список имен. По умолчанию `None`

Результат Список объектов

Тип результата List[*TrObject*]

1.6.17 Borders

`class Borders(server_guid=None)`

Класс для работы с линиями пересечения

Параметры `server_guid` (`str` | `List[str]`, optional) – Guid сервера или список guid. По умолчанию `None`, что соответствует всем доступным серверам.

Examples

```
>>> borders = Borders()
>>> borders.get_simt()
[TrObject('DBOP')]
>>> borders.get_all()
[TrObject('Вход в офис'), TrObject('DBOP')]
```

`get_head(names=None)`

Возвращает список HeadBorders

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_people(names=None)`

Возвращает список PeopleBorders

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_simt(names=None)`

Возвращает список SIMT Borders

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_deep_people(names=None)`

Возвращает список DeepPeopleBorders

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

`get_all(names=None)`

Возвращает список всех линий пересечения

Параметры `names` (`str` | `list`, optional) – `str` - имена, разделенные запятыми или `list` - список имен. По умолчанию `None`

Результат Список объектов

Тип результата `List[TrObject]`

1.7 ShotSaver

```
class ShotSaver(shot_awaiting_time=5,                tries_to_make_shot=2,
                pool_size=10,                        host_api=<module 'host' from
                '/home/docs/checkouts/readthedocs.org/user_builds/trassir-script-
                framework/checkouts/stable/tests/host/__/__init__.pyc'>)
```

Класс для сохранения скриншотов

Параметры

- `shot_awaiting_time` (`int`, optional) – Время ожидания скриншота, с. По умолчанию 5.
- `tries_to_make_shot` (`int`, optional) – Кол-во попыток сохранить скриншот. Если в течении времени `shot_awaiting_time` скриншот не был сохранен - производится следующая попытка сохранить скриншот. По умолчанию 2
- `pool_size` (`int`) – Размер пула. По умолчанию 10

`pool_size`

Размер пула для метода `pool_shot`

Устанавливает размер пула (кол-во одновременно созданных задач сохранения скриншотов). По умолчанию 10.

Предупреждение: Изменить данный параметр можно только до первого вызова метода `pool_shot`. После вызовет `RuntimeError`

Raises `RuntimeError` – Если пул уже создан.

Type `int`

`pool_queue_size`

Размер текущей очереди в пуле

Возвращает текущий размер очереди в пуле.

Примечание: Если пул еще не был создан (метод `pool_shot` не вызывался) данный метод вернет `-1`

Type `int`

`pool_working`

`True` если в пуле есть не законченные задачи

Примечание: Если пул еще не был создан (метод `pool_shot` не вызывался) данный метод вернет `None`

Type `bool`

`screenshots_folder`

Папка для сохранения скриншотов по умолчанию

Устанавливает новый путь по умолчанию для сохранения скриншотов, если папка не существует - создает папку. Или возвращает текущий путь для сохранения скриншотов.

Примечание: По умолчанию `screenshots_folder` = `BaseUtils.get_screenshot_folder()`

Raises `OSError` – Если возникает ошибка при создании папки

Type `str`

`shot(channel_full_guid, dt=None, file_name=None, file_path=None)`

Делает скриншот с указанного канала

Примечание: По умолчанию:

- `dt=datetime.now()`
 - `file_name="{name} (%Y.%m.%d %H-%M-%S).jpg"`, где `{name}` - имя канала
-

Параметры

- `channel_full_guid (str)` – Полный guid анала. Например: `"CFsuNBzt_pV4ggECb"`
- `dt (datetime.datetime, optional)` – `datetime.datetime` для скриншота. По умолчанию `None`
- `file_name (str, optional)` – Имя файла с расширением. По умолчанию `None`
- `file_path (str, optional)` – Путь для сохранения скриншота. По умолчанию `None`

Результат Полный путь до скриншота

Тип результата `str`

Raises

- `ValueError` – Если в guid канала отсутствует guid сервера
- `TypeError` – Если `isinstance(dt, (datetime, date)) is False`

Examples

```
>>> ss = ShotSaver()
>>> ss.shot("e80kgBLh_pV4ggECb")
'/home/trassir/shots/AC-D2141IR3 Склад (2019.04.03 15-58-26).jpg'
```

`async_shot(channel_full_guid, dt=None, file_name=None, file_path=None, callback=None)`

Вызывает `callback` после сохранения скриншота

- Метод работает в отдельном потоке
- Вызывает функцию `ShotSaver.shot()`
- Ждет выполнения функции `BaseUtils.check_file()` `tries=10`
- Вызывает `callback` функцию

Параметры

- `channel_full_guid (str)` – Полный guid канала. Например: `"CFsuNBzt_pV4ggECb"`
- `dt (datetime.datetime, optional)` – `datetime.datetime` для скриншота. По умолчанию `None`
- `file_name (str, optional)` – Имя файла с расширением. По умолчанию `None`
- `file_path (str, optional)` – Путь для сохранения скриншота. По умолчанию `None`

- `callback` (function, optional) – Функция, которая вызывается после сохранения скриншота. В качестве аргументов должна принимать `success`, `shot_path`. По умолчанию `None`

Результат Thread object

Тип результата `threading.Thread`

Examples

```
>>> # noinspection PyUnresolvedReferences
>>> def callback(success, shot_path):
...     # Пример callback функции
...     # Args:
...     #     success (bool): True если скриншот успешно сохранен, иначе False
...     #     shot_path (str): Полный путь до скриншота
...     if success:
...         host.message("Скриншот успешно сохранен<br>%s" % shot_path)
...     else:
...         host.error("Ошибка сохранения скриншота <br>%s" % shot_path)
>>>
>>> ss = ShotSaver()
>>> ss.async_shot("e80kgBLh_pV4ggECb", callback=callback)
```

```
pool_shot(channel_full_guid, dt=None, file_name=None, file_path=None,
          callback=None)
```

Сохраняет скриншоты в пуле.

Одновременно в работе не более `ShotSaver.pool_size` задач.

Предупреждение: Данный метод создает `ShotSaver.pool_size` доп. потоков. Потоки удаляются при отключении скрипта.

Параметры

- `channel_full_guid` (str) – Полный guid канала. Например: "CFsuNBzt_pV4ggECb"
- `dt` (datetime.datetime, optional) – `datetime.datetime` для скриншота. По умолчанию `None`
- `file_name` (str, optional) – Имя файла с расширением. По умолчанию `None`
- `file_path` (str, optional) – Путь для сохранения скриншота. По умолчанию `None`
- `callback` (function, optional) – Функция, которая вызывается после сохранения скриншота. В качестве аргументов должна принимать `success`, `shot_path`. По умолчанию `None`

Examples

```
>>> ss = ShotSaver()
>>> ss.pool_size = 2
>>>
>>> ss.pool_shot("e80kgBLh_pV4ggECb")
>>> ss.pool_shot("e80kgBLh_pV4ggECb")
>>> ss.pool_shot("e80kgBLh_pV4ggECb")
>>> ss.pool_shot("e80kgBLh_pV4ggECb")
```

(continues on next page)

(продолжение с предыдущей страницы)

```
>>>
>>> ss.pool_queue_size
4
```

1.8 VideoExporter

```
class VideoExporter(host_api=<module 'host' from '/home/docs/checkouts/readthedocs.org/user_builds/trassir-script-framework/checkouts/stable/tests/host/___init___ .pyc'>)
```

Класс для экспорта видео

Examples

Смена папки экспорта видео по умолчанию

```
>>> ss = VideoExporter()
>>> ss.export_folder
'/home/trassir/shots'
>>> ss.export_folder += "/my_videos"
>>> ss.export_folder
'/home/trassir/shots/my_videos'
```

Экспорт видео с вызовом `callback` функции после выполнения.

Начало экспорта - 120 секунд назад, продолжительность 60 сек.

```
>>> # noinspection PyUnresolvedReferences
>>> def callback(success, file_path, channel_full_guid):
...     # Пример callback функции
...     # Args:
...     #     success (bool): True если видео экспортировано успешно, иначе False
...     #     file_path (str): Полный путь до видеофайла
...     #     channel_full_guid (str) : Полный guid канала
...     if success:
...         host.message("Экспорт успешно завершен<br>%s" % file_path)
...     else:
...         host.error("Ошибка экспорта<br>%s" % file_path)
```

```
>>> ss = VideoExporter()
>>> dt_start = datetime.now() - timedelta(seconds=120)
>>> ss.export(callback, "e80kgBLh_pV4ggECb", dt_start)
```

export_folder

Папка для экспорта видео по умолчанию

Устанавливает новый путь по умолчанию для экспорта видео, если папка не существует - создает папку. Или возвращает текущий путь для экспорта видео.

Примечание: По умолчанию `export_folder` = `BaseUtils.get_screenshot_folder()`

Raises `OSError` – Если возникает ошибка при создании папки

Type `str`

```
export(channel_full_guid, dt_start, dt_end=None, duration=60,
       prefer_substream=False, file_name=None, file_path=None, callback=None)
```

Запускает экспорт или добавляет задачу экспорта в очередь.

После завершения экспорта вызывает `callback` функцию а также очищает список задач экспорта в панели управления Trassir.

Примечание: Задача экспорта добавляется только после завершения предыдущей.

Совет:

- Вы можете задать время начала и окончания экспорта `dt_start`, `dt_end`.
 - Или можно задать время начала экспорта `dt_start` и продолжительность экспорта (в сек.) `duration`. По умолчанию `duration=60`.
 - Если `dt_end=None` функция использует `duration` для вычисления времени окончания `dt_end = dt_start + timedelta(seconds=duration)`.
-

Параметры

- `channel_full_guid` (`str`) – Полный `guid` канала. Например: `"CFsuNBzt_pV4ggECb"`
- `dt_start` (`datetime.datetime`) – `datetime.datetime` начала экспорта
- `dt_end` (`datetime.datetime`, optional) – `datetime.datetime` окончания экспорта. По умолчанию `None`
- `duration` (`int`, optional) – Продолжительность экспорта, в секундах. Используется если `dt_end` is `None`. По умолчанию 60
- `prefer_substream` (`bool`, optional) – Если `True` - Экспортирует субпоток. По умолчанию `False`
- `file_name` (`str`, optional) – Имя экспортируемого файла. По умолчанию `None`
- `file_path` (`str`, optional) – Путь для экспорта. По умолчанию `None`
- `callback` (`function`, optional) – Функция, которая вызывается после завершения экспорта. По умолчанию `None`

1.9 Senders

1.9.1 PopupSender

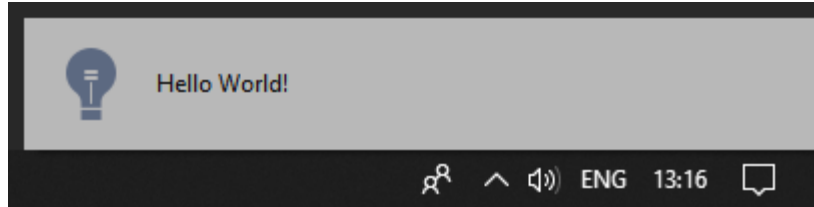
```
class PopupSender(width=400)
```

Класс для показа всплывающих окон в правом нижнем углу экрана

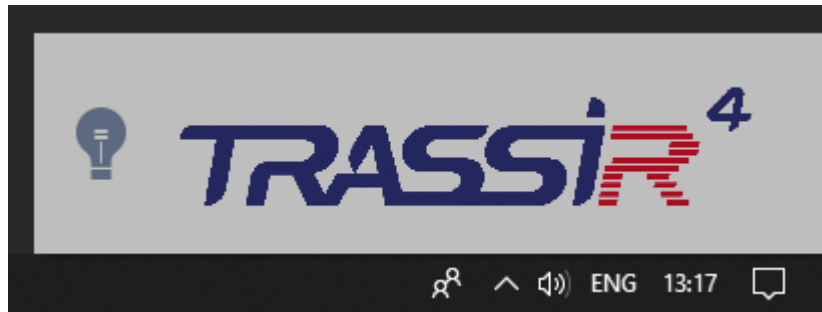
Параметры `width` (`int`, optional) – Ширина изображения, `px`. По умолчанию `width=400`

Examples

```
>>> sender = PopupSender(300)
>>> sender.text("Hello World!")
```



```
>>> sender.image(r"manual/en/cloud-devices-16.png")
```



```
text(text, popup_type='message', **kwargs)
```

Показывает текст во всплывающем окне

Вызывает один из методов Trassir `host.alert`, `host.message` или `host.error` с текстом

Параметры

- `text` (`str`) – Текст сообщения
- `popup_type` ("message" | "alert" | "error", optional) – Тип сообщения. По умолчанию "message"

```
image(image_path, text="", popup_type=None, **kwargs)
```

Показывает изображение во всплывающем окне

Параметры

- `image_path` (`str` | `List[str]`) – Полный путь до изображения или список путей.
- `text` (`str`, optional) – Текст сообщения. По умолчанию ""
- `popup_type` ("message" | "alert" | "error", optional) – Тип сообщения. По умолчанию "message"

1.9.2 PopupWithBtnSender

```
class PopupWithBtnSender(width=800)
```

Класс для показа всплывающих окон с кнопкой *Ok*

Примечание:

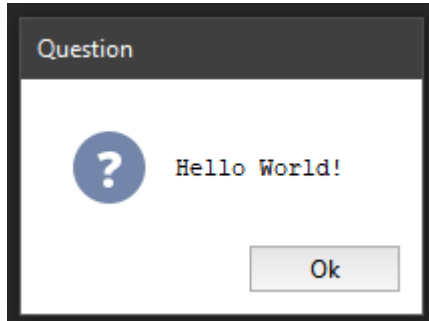
Для закрытия окна необходимо нажать кнопку *Ok* в течении 60 сек.

После 60 сек окно закрывается автоматически.

Параметры `width` (`int`, optional) – Ширина изображения, px. По умолчанию `width=800`

Examples

```
>>> sender = PopupWithBtnSender()
>>> sender.text("Hello World!")
```



```
>>> sender.image(r"manual/en/cloud-devices-16.png")
```



`text(text, **kwargs)`

Показывает текст во всплывающем окне

Вызывает метод `Trassir host.question` с текстом

Параметры `text (str)` – Текст сообщения

`image(image_path, text="", **kwargs)`

Показывает изображение во всплывающем окне

Параметры

- `image_path (str | List[str])` – Полный путь до изображения или список путей.
- `text (str, optional)` – Текст сообщения. По умолчанию ""

1.9.3 EmailSender

```
class EmailSender(account, mailing_list, subject=None, max_size=None)
```

Класс для отправки уведомлений, изображений и файлов на почту

Примечание: По умолчанию тема сообщений соответствует шаблону `{server_name}`
 -> `{script_name}`

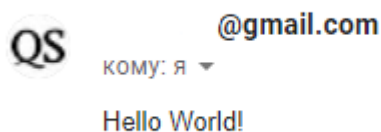
Совет: При отправке изображения с текстом предпочтительней использовать метод `EmailSender.image()` с необязательным аргументом `text` чем `EmailSender.text()` с необязательным аргументом `attachments`

Параметры

- `account (str)` – E-Mail аккаунт trassir. Проверяется методом `PokaYoke.check_email_account()`
- `mailing_list (str)` – Список email адресов для отправки писем разделенный запятыми. Проверяется и парсится в список методом `PokaYoke.parse_emails()`
- `subject (str, optional)` – Общая тема для сообщений. По умолчанию `None`
- `max_size (int, optional)` – Максимальный размер вложения, байт. По умолчанию `25 * 1024 * 1024`

Examples

```
>>> sender = EmailSender("MyAccount", "my_mail@google.com")
>>> sender.text("Hello World!")
```



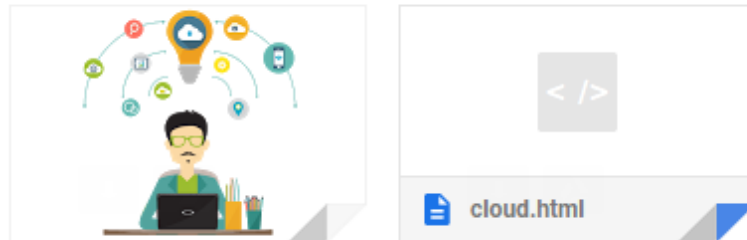
```
>>> sender.image(r"manual/en/cloud-devices-16.png")
```



```
>>> sender.files([r"manual/en/cloud.html", r"manual/en/cloud.png"])
```




2 прикрепленных файла



`text(text, subject=None, attachments=None, **kwargs)`

Отправка текстового сообщения

Параметры

- `text` (`str`) – Текст сообщения
- `subject` (`str`, optional) – Новая тема сообщения. По умолчанию `None`
- `attachments` (`list`, optional) – Список вложений. По умолчанию `None`

`image(image_path, text="", subject=None, **kwargs)`

Отправка изображения

Параметры

- `image_path` (`str` | `List[str]`) – Полный путь до изображения или список путей.
- `text` (`str`, optional) – Текст сообщения. По умолчанию ""
- `subject` (`str`, optional) – Новая тема сообщения. По умолчанию `None`

`files(file_paths, text="", subject=None, callback=None, **kwargs)`

Отправка файлов

Примечание: Если отправляется несколько файлов они могут быть разделены на несколько сообщений, основываясь на максимальном размере вложений.

Параметры

- `file_paths` (`str` | `list`) – Путь до файла или список файлов для отправки
- `text` (`str`, optional) – Текст сообщения. По умолчанию ""
- `subject` (`str`, optional) – Новая тема сообщения. По умолчанию `None`
- `callback` (`function`, optional) – Функция, которая вызывается после отправки частей

1.9.4 TelegramSender

`class TelegramSender(telegram_ids=None)`

Работа с телеграм ботом @trassirbot

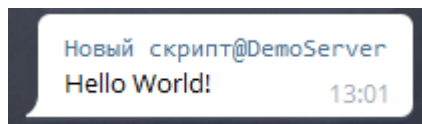
Предупреждение:

Скрипт должен быть запущен на **сервере** Trassir.
На Клиенте скрипт вызовет ошибку `ServerKeyError`

Параметры `telegram_ids (str)` – Id пользователей, через запятую.

Examples

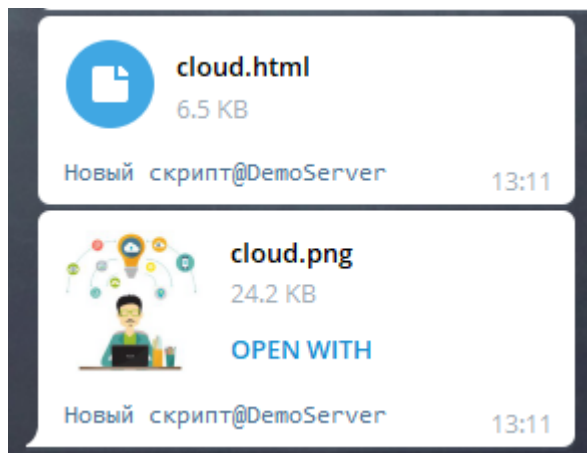
```
>>> # Можно указать id для рассылки при инициализации
>>> # класса, для всех уведомлений
>>> sender = TelegramSender("123456789")
>>> sender.text("Hello World!")
```



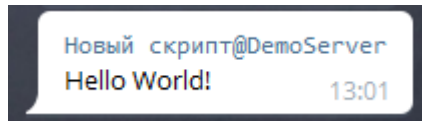
```
>>> sender.image(r"manual/en/cloud-devices-16.png")
```



```
>>> sender.files([r"manual/en/cloud.html", r"manual/en/cloud.png"])
```



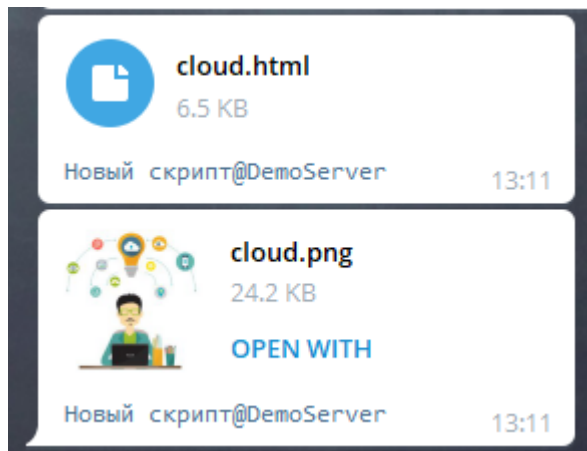
```
>>> # Или можно опередовать telegram id при вызове методов
>>> sender = TelegramSender()
>>> sender.text("Hello World!", tg_users=[123456789])
```



```
>>> sender.image(r"manual/en/cloud-devices-16.png", tg_users=[123456789])
```



```
>>> sender.files([r"manual/en/cloud.html", r"manual/en/cloud.png"], tg_
←users=[123456789])
```



text(text, tg_users=None, **kwargs)

Отправка текстового сообщения

Параметры

- **text** (str) – Текст сообщения.
- **tg_users** (List[int], optional) – Список id пользователей telegram для отправки отдельных сообщений. По умолчанию None

image(image_path, text="", tg_users=None, remove=False, **kwargs)

Отправка изображения

Параметры

- **image_path** (str | List[str]) – Полный путь до изображения или список путей
- **text** (str, optional) – Текст сообщения. По умолчанию ""

- `tg_users` (`List[int]`, optional) – Список id пользователей telegram для отправки отдельных сообщений. По умолчанию `None`
- `remove` (`bool`, optional) – Удалить файл после отправки или нет. По умолчанию `False`

```
files(file_paths, text="", tg_users=None, remove=False, **kwargs)
```

Отправка файлов

Параметры

- `file_paths` (`str` | `list`) – Путь до файла или список файлов для отправки
- `text` (`str`, optional) – Текст сообщения. По умолчанию ""
- `tg_users` (`List[int]`, optional) – Список id пользователей telegram для отправки отдельных сообщений. По умолчанию `None`
- `remove` (`bool`, optional) – Удалить файл после отправки или нет. По умолчанию `False`

1.9.5 SMSCSender

```
class SMSCSender(login, password, phones, translit=True)
```

Класс для отправки сообщений с помощью сервиса smsc.ru

См.также:

<https://smc.ru/api/http/>

Примечание:

Номера проверяются методом `PokaYoke.check_phones()`

Также при первом запуске скрипт проверяет данные авторизации

Предупреждение:

По умолчанию сервис smsc.ru отправляет сообщения от своего имени *SMSC.RU*. При этом отправка на номера Мегафон/Йота **недоступна** т.к. имя *SMSC.RU* заблокировано оператором.

Мы настоятельно **НЕ** рекомендуем использовать стандартное имя *SMSC.RU*.

Для отправки смс от вашего буквенного имени необходимо его создать в разделе - <https://smc.ru/senders/> и зарегистрировать для операторов в колонке Действия по кнопке Изменить (после заключения договора согласно инструкции - <https://smc.ru/contract/info/>) а также приложить гарантийное письмо на МТС в личный кабинет <http://smc.ru/documents/> и отправить на почту inna@smc.ru

Параметры

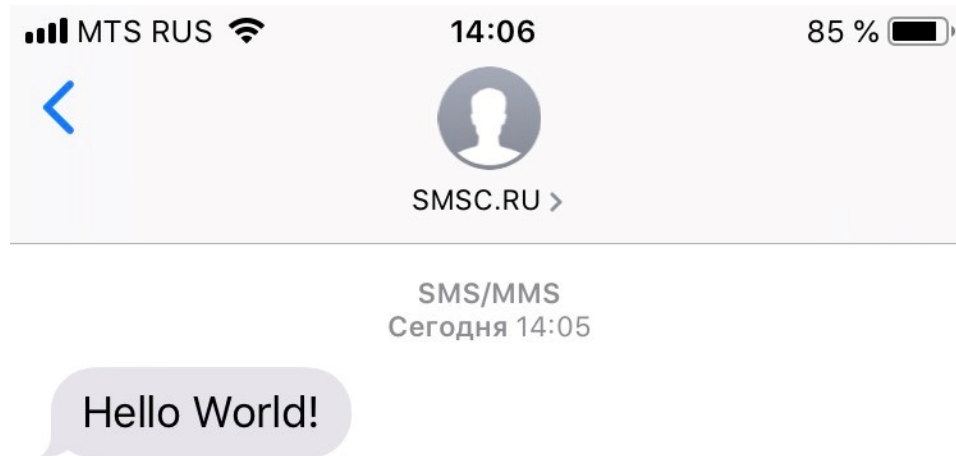
- `login` (`str`) – SMSC Логин
- `password` (`str`) – SMSC Пароль
- `phones` (`str`) – Список номеров для отправки смс разделенный запятыми или точкой с запятой

- `translit (bool, optional)` – Переводить сообщение в транслит. По умолчанию `True`

Raises `SMSCSenderError` – При любых ошибках с отправкой сообщения

Examples

```
>>> sender = SMSCSender("login", "password", "79999999999")
>>> sender.text("Hello World!")
```



`text(text, **kwargs)`

Отправка текстового сообщения

Параметры `text (str)` – Текст сообщения.

1.9.6 FTPSender

```
class FTPSender(host, port=21, user='anonymous', passwd="", work_dir=None,
               callback=None, queue_maxlen=1000)
```

Класс для отправки файлов на ftp сервер

При инициализации проверяют подключение к ftp серверу. Файлы отправляет по очереди. Максимальный размер очереди можно изменить. Во время выполнения передает текущий прогресс отправки файла в callback функцию.

Примечание:

Помимо прогресса в функцию `callback` может вернуться код ошибки.

- -1 Файл не существует.
 - -2 Ошибка отправки на ftp, файл будет повторно отправлен.
 - -3 Неизвестная ошибка.
-

Параметры

- `host (str)` – Адрес ftp сервера.
- `port (int, optional)` – Порт ftp сервера. По умолчанию `port=21`
- `user (str, optional)` – Имя пользователя. По умолчанию `"anonymous"`
- `passwd (str, optional)` – Пароль пользователя. По умолчанию `passwd=""`
- `work_dir (str, optional)` – Директория на сервере для сохранения файлов. По умолчанию `None`
- `callback (function, optional)` – Callable function. По умолчанию `None`
- `queue_maxlen (int, optional)` – Максимальная длина очереди на отправку. По умолчанию `queue_maxlen=1000`

Examples

```
>>> # noinspection PyUnresolvedReferences
>>> def callback(file_path, progress, error):
...     # Пример callback функции, которая отображает
...     # текущий прогресс в счетчике запуска скрипта
...     # Args:
...     #   file_path (str): Путь до файла
...     #   progress (int): Текущий прогресс передачи файла, %
...     #   error (str | Exception): Ошибка при отправке файла, если есть
...     host.stats()["run_count"] = progress
...     if error:
...         host.error(error)
...
...     if progress == 100:
...         host.timeout(3000, lambda: os.remove(BaseUtils.win_encode_path(file_
... path)))
>>>
>>> sender = FTPSender("172.20.0.10", 21, "trassir", "12345", work_dir="/test_dir/",
... callback=callback)
>>> sender.files(r"D:/Shots/export_video.avi")
```

`files(file_paths, *args, **kwargs)`
Отправка файлов

Примечание: Можно указать отдельный путь на ftp сервере для каждого файла. Для этого список файлов на отправку должен быть приведен к виду `[(shot_path, ftp_path), ...]` При этом так же будет учитываться глобальная папка `work_dir` заданная при инициализации класса.

Параметры `file_paths (str | list)` – Путь до файла или список файлов для отправки

1.9.7 SoundPlayer

```
class SoundPlayer(sound_file, host_api=<module 'host' from
'/home/docs/checkouts/readthedocs.org/user_builds/trassir-script-
framework/checkouts/stable/tests/host/__init__.pyc'>)
Класс для проигрывания выбранной мелодии.
```

Можно указать один из стандартных звуков или указать полный путь до своего файла.

Примечание: Список стандартных файлов

Параметры `sound_file` (`str`) – Имя файла с расширением

`play()`

Проигрывает выбранный файл

Examples

```
>>> player = SoundPlayer("alarm.wav")
>>> player.play()
```


2.1 Changelog

2.1.1 Version 0.64

Released 2019-07-25

New Features

- Add global *logger*. You need just set *DEBUG = True*
- Add `trassir_script_framework.BaseUtils.context_menu`
- Add `trassir_script_framework.BaseUtils.context_menu_button()`
- Add `trassir_script_framework.SoundPlayer`

Changes

- Refactoring `trassir_script_framework.BaseUtils.is_template_exists()`
- Fix `trassir_script_framework.BaseUtils.is_template_exists()` returns returns *1e15*
- Now you can pass list argument to senders.

2.1.2 Version 0.63

Released 2019-07-19

New Features

- Added `trassir_script_framework.BaseUtils.dt_to_ts()`
- Added `trassir_script_framework.StockScript`
- Added `trassir_script_framework.PokaYoke.fire_recognizer_events()`

Changes

- Rename `trassir_script_framework.BaseUtils.ts_to_datetime()` -> `trassir_script_framework.BaseUtils.dt_to_ts()`
- Update TBotAPI: Fix - Send image method with cyrillic symbols on WinOS; Add - send_image_album
- Fix docstrings

2.1.3 Version 0.61

Released 2019-07-12

New Features

- Added `trassir_script_framework.Schedules.on_load()`
- Added `trassir_script_framework.BaseUtils.event_type_encode()`
- Added `trassir_script_framework.BaseUtils.event_type_decode()`

Changes

- Exec encoded tbot_service in TelegramSender `__init__`
- Rebuild `trassir_script_framework.ShotSaver.pool_shot()`
- Rebuild `trassir_script_framework.BaseUtils.get_logger()`

2.1.4 Version 0.60

Released 2019-07-05

Changes

- Fix `trassir_script_framework.BaseUtils.win_encode_path()`
- Update tbot_api
- Other fixes

2.1.5 Version 0.50

Released 2019-05-16

New Features

- Added `trassir_script_framework.BaseUtils.save_pkl()`
- Added `trassir_script_framework.BaseUtils.load_pkl()`
- Added `trassir_script_framework.Templates`
- Added `trassir_script_framework.GUITemplate.delete()`

Changes

- Fix ShotSaver examples
- Rename Template to `trassir_script_framework.GUITemplate`
- Fix some errors in `trassir_script_framework.BaseUtils.image_to_base64()`
- Add extension to file_log in `trassir_script_framework.BaseUtils.get_logger()`

2.1.6 Version 0.4

Released 2019-04-30

New Features

- Added `trassir_script_framework.BaseUtils.set_script_name()`

Changes

- Fix `catch_request_exceptions` in `trassir_script_framework.HTTPRequester`
- Add `tg_users` argument to `trassir_script_framework.TelegramSender` methods

2.1.7 Version 0.3

Released 2019-04-26

New Features

- Added `trassir_script_framework.BaseUtils.is_template_exists()`
- Added `trassir_script_framework.HTTPRequester`
- Added `trassir_script_framework.ShotSaver.pool_shot()`
- Added `trassir_script_framework.Persons.get_person_by_name()`
- Added `trassir_script_framework.BaseUtils.image_to_base64()`
- Added `trassir_script_framework.BaseUtils.base64_to_html_img()`
- Added more examples

Changes

- Added data to string transform `trassir_script_framework.ScriptObject.fire_event_v2()`
- Added `raise_exception` in functions with `trassir_script_framework.BaseUtils.run_as_thread_v2()`
- Change argument structure in `trassir_script_framework.BaseUtils.to_json()`
- Change argument structure in `trassir_script_framework.ShotSaver.async_shot()`
- Fix: `trassir_script_framework.FTPSender` errors when send several files
- Fix: `trassir_script_framework.FTPSender` remove `self.logger`
- Fix: `trassir_script_framework.BaseUtils.get_logger()` permission denied when deleting log file

2.1.8 Version 0.2b

Released 2019-04-12

New Features

- Added `trassir_script_framework.FTPSender`
- Added `trassir_script_framework.BaseUtils.lpr_flags_decode()`
- Added more examples

Changes

- Transferring PokaYoke GET methods to independent classes and extended functionality:

```
PokaYoke.get_servers -> trassir_script_framework.Servers
PokaYoke.get_persons -> trassir_script_framework.Persons
PokaYoke.get_person_folders -> trassir_script_framework.Persons
PokaYoke.get_users -> trassir_script_framework.Users
PokaYoke.get_terminals -> trassir_script_framework.PosTerminals
PokaYoke.get_ip_cameras -> trassir_script_framework.Devices
PokaYoke.get_channels -> trassir_script_framework.Channels
PokaYoke.get_networks -> trassir_script_framework.NetworkNodes
PokaYoke.get_rules -> trassir_script_framework.Rules
PokaYoke.get_scripts -> trassir_script_framework.Scripts
PokaYoke.get_schedules -> trassir_script_framework.Schedules
PokaYoke.get_servers -> trassir_script_framework.EmailAccounts
PokaYoke.get_template_loops -> trassir_script_framework.TemplateLoops
PokaYoke.get_gpio_inputs -> trassir_script_framework.GPIO
PokaYoke.get_gpio_outputs -> trassir_script_framework.GPIO
PokaYoke.get_people_zones -> trassir_script_framework.Zones
PokaYoke.get_simt_zones -> trassir_script_framework.Zones
PokaYoke.get_workplaces -> trassir_script_framework.Zones
PokaYoke.get_queues -> trassir_script_framework.Zones
PokaYoke.get_shelves -> trassir_script_framework.Zones
PokaYoke.get_people_zones -> trassir_script_framework.Zones
PokaYoke.get_head_borders -> trassir_script_framework.Borders
PokaYoke.get_people_borders -> trassir_script_framework.Borders
PokaYoke.get_simt_borders -> trassir_script_framework.Borders
PokaYoke.get_deep_people_borders -> trassir_script_framework.Borders
PokaYoke.get_all_borders -> trassir_script_framework.Borders
PokaYoke.get_access_points -> trassir_script_framework.Sigur
```

- Rename some method and classes:

```
BaseUtils.check_file -> trassir_script_framework.BaseUtils.is_file_exists()
BaseUtils.check_folder -> trassir_script_framework.BaseUtils.is_folder_exists()
BaseUtils.pretty_json -> trassir_script_framework.BaseUtils.to_json()
PokaYokeObject -> trassir_script_framework.TrObject()
```

- Fix some typos

2.1.9 Version 0.1b

Released 2019-04-05

- Beta release

Indices and tables

- `genindex`
- `search`

A

`async_shot()` (метод *ShotSaver*), 38

B

`ban_client()` (статический метод *PokaYoke*), 16

`ban_daemon()` (метод класса *trassir_script_framework.PokaYoke*), 17

`ban_tos()` (статический метод *PokaYoke*), 16

`ban_win()` (статический метод *PokaYoke*), 16

`base64_to_html_img()` (метод класса *trassir_script_framework.BaseUtils*), 9

Borders (класс в *trassir_script_framework*), 36

C

`cat()` (метод класса *trassir_script_framework.BaseUtils*), 5

Channels (класс в *trassir_script_framework*), 28

`check_email_account()` (статический метод *PokaYoke*), 17

`check_me` (атрибут *ScriptObject*), 19

`check_phones()` (метод класса *trassir_script_framework.PokaYoke*), 18

`code` (атрибут *HTTPRequester.Response*), 14

`code` (атрибут *Response*), 15

`content` (атрибут *GUITemplate*), 21

`context_menu_button()` (метод *ScriptObject*), 20

D

`delete()` (метод *GUITemplate*), 22

Devices (класс в *trassir_script_framework*), 27

`dt_to_ts()` (статический метод *BaseUtils*), 7

E

EmailAccounts (класс в *trassir_script_framework*), 33

EmailSender (класс в *trassir_script_framework*), 43

`event_type_decode()` (метод класса *trassir_script_framework.BaseUtils*), 8

`event_type_encode()` (метод класса *trassir_script_framework.BaseUtils*), 8

`export()` (метод *VideoExporter*), 41

`export_folder` (атрибут *VideoExporter*), 40

F

`files()` (метод *EmailSender*), 45

`files()` (метод *FTPSender*), 50

`files()` (метод *TelegramSender*), 48

`fire_event_v2()` (метод *ScriptObject*), 20

`fire_recognizer_events()` (метод класса *trassir_script_framework.PokaYoke*), 18

`folder` (атрибут *ScriptObject*), 20

FTPSender (класс в *trassir_script_framework*), 49

`full_guid` (атрибут *TrObject*), 22

G

`get()` (метод *HTTPRequester*), 14

`get_all()` (метод *Borders*), 36

`get_all()` (метод *Channels*), 28

`get_all()` (метод *Devices*), 28

`get_all()` (метод *EmailAccounts*), 33

`get_all()` (метод *NetworkNodes*), 29

`get_all()` (метод *PosTerminals*), 27

`get_all()` (метод *Rules*), 30

`get_all()` (метод *Schedules*), 33

`get_all()` (метод *Scripts*), 31

`get_all()` (метод *Servers*), 23

`get_all()` (метод *StockScripts*), 31

`get_all()` (метод *TemplateLoops*), 34

`get_all()` (метод *Templates*), 26

`get_deep_people()` (метод *Borders*), 36

- `get_disabled()` (метод *Channels*), 28
 - `get_disabled()` (метод *Devices*), 27
 - `get_disabled()` (метод *NetworkNodes*), 29
 - `get_disabled()` (метод *PosTerminals*), 27
 - `get_disabled()` (метод *Rules*), 30
 - `get_disabled()` (метод *Schedules*), 33
 - `get_disabled()` (метод *Scripts*), 31
 - `get_disabled()` (метод *StockScripts*), 31
 - `get_disabled()` (метод *TemplateLoops*), 34
 - `get_enabled()` (метод *Channels*), 28
 - `get_enabled()` (метод *Devices*), 27
 - `get_enabled()` (метод *NetworkNodes*), 29
 - `get_enabled()` (метод *PosTerminals*), 26
 - `get_enabled()` (метод *Rules*), 30
 - `get_enabled()` (метод *Schedules*), 32
 - `get_enabled()` (метод *Scripts*), 30
 - `get_enabled()` (метод *StockScripts*), 31
 - `get_enabled()` (метод *TemplateLoops*), 34
 - `get_folders()` (метод *Persons*), 24
 - `get_groups()` (метод *Users*), 25
 - `get_head()` (метод *Borders*), 36
 - `get_inputs()` (метод *GPIO*), 34
 - `get_local()` (метод *Servers*), 23
 - `get_logger()` (метод класса *trassir_script_framework.BaseUtils*), 12
 - `get_operator_gui()` (метод класса *trassir_script_framework.BaseUtils*), 11
 - `get_outputs()` (метод *GPIO*), 34
 - `get_people()` (метод *Borders*), 36
 - `get_people()` (метод *Zones*), 35
 - `get_person_by_guid()` (метод *Persons*), 24
 - `get_person_by_name()` (метод *Persons*), 25
 - `get_persons()` (метод *Persons*), 24
 - `get_queues()` (метод *Zones*), 35
 - `get_remote()` (метод *Servers*), 23
 - `get_screenshot_folder()` (метод класса *trassir_script_framework.BaseUtils*), 12
 - `get_script_name()` (метод класса *trassir_script_framework.BaseUtils*), 11
 - `get_server_guid()` (метод класса *trassir_script_framework.BaseUtils*), 11
 - `get_shelves()` (метод *Zones*), 35
 - `get_simt()` (метод *Borders*), 36
 - `get_simt()` (метод *Zones*), 35
 - `get_users()` (метод *Users*), 25
 - `get_users_by_groups()` (метод *Users*), 26
 - `get_workplaces()` (метод *Zones*), 35
 - GPIO* (класс в *trassir_script_framework*), 34
 - `guid` (атрибут *GUITemplate*), 21
 - `guid` (атрибут *TrObject*), 22
 - GUITemplate* (класс в *trassir_script_framework*), 20
- ## H
- `health` (атрибут *ScriptObject*), 19
 - HTTPRequester* (класс в *trassir_script_framework*), 13
 - HTTPRequester.Response* (класс в *trassir_script_framework*), 14
- ## I
- `image()` (метод *EmailSender*), 45
 - `image()` (метод *PopupSender*), 42
 - `image()` (метод *PopupWithBtnSender*), 43
 - `image()` (метод *TelegramSender*), 47
 - `image_to_base64()` (метод класса *trassir_script_framework.BaseUtils*), 9
 - `is_file_exists()` (статический метод *BaseUtils*), 4
 - `is_folder_exists()` (статический метод *BaseUtils*), 4
 - `is_template_exists()` (метод класса *trassir_script_framework.BaseUtils*), 5
- ## J
- `json` (атрибут *HTTPRequester.Response*), 14
 - `json` (атрибут *Response*), 15
- ## L
- `load_pkl()` (статический метод *BaseUtils*), 10
 - `lpr_flags_decode()` (метод класса *trassir_script_framework.BaseUtils*), 7
- ## N
- `name` (атрибут *GUITemplate*), 21
 - `name` (атрибут *ScriptObject*), 20
 - `name` (атрибут *TrObject*), 22
 - NetworkNodes* (класс в *trassir_script_framework*), 29
- ## O
- `obj` (атрибут *TrObject*), 22
 - `obj_methods` (атрибут *TrObject*), 22
 - `on_load()` (метод *Schedules*), 32
- ## P
- `parent` (атрибут *TrObject*), 22
 - `parse_emails()` (метод класса *trassir_script_framework.PokaYoke*), 17

- path (ампубум TrObject), 22
 Persons (класс в trassir_script_framework), 23
 play() (метод SoundPlayer), 51
 PokaYoke (класс в trassir_script_framework), 16
 pool_queue_size (ампубум ShotSaver), 37
 pool_shot() (метод ShotSaver), 39
 pool_size (ампубум ShotSaver), 37
 pool_working (ампубум ShotSaver), 37
 PopupSender (класс в trassir_script_framework), 41
 PopupWithBtnSender (класс в trassir_script_framework), 42
 post() (метод HTTPRequester), 15
 PosTerminals (класс в trassir_script_framework), 26
- ## R
- Response (класс в trassir_script_framework.HTTPRequester), 15
 Rules (класс в trassir_script_framework), 29
 run_as_thread() (статический метод BaseUtils), 3
- ## S
- save_pkl() (статический метод BaseUtils), 10
 Schedules (класс в trassir_script_framework), 32
 screenshots_folder (ампубум ShotSaver), 37
 ScriptObject (класс в trassir_script_framework), 19
 Scripts (класс в trassir_script_framework), 30
 server (ампубум TrObject), 22
 Servers (класс в trassir_script_framework), 23
 set_script_name() (метод класса trassir_script_framework.BaseUtils), 13
 settings (ампубум TrObject), 22
 shot() (метод ShotSaver), 38
 ShotSaver (класс в trassir_script_framework), 37
 show() (метод GUITemplate), 22
 SMSCSender (класс в trassir_script_framework), 48
 SoundPlayer (класс в trassir_script_framework), 50
 StockScripts (класс в trassir_script_framework), 31
- ## T
- TelegramSender (класс в trassir_script_framework), 45
 TemplateLoops (класс в trassir_script_framework), 33
 Templates (класс в trassir_script_framework), 26
 text (ампубум HTTPRequester.Response), 14
 text (ампубум Response), 15
 text() (метод EmailSender), 45
 text() (метод PopupSender), 42
 text() (метод PopupWithBtnSender), 43
 text() (метод SMSCSender), 49
 text() (метод TelegramSender), 47
 to_json() (метод класса trassir_script_framework.BaseUtils), 6
 TrObject (класс в trassir_script_framework), 22
 ts_to_dt() (статический метод BaseUtils), 7
 type (ампубум TrObject), 22
- ## U
- Users (класс в trassir_script_framework), 25
- ## V
- VideoExporter (класс в trassir_script_framework), 40
- ## W
- win_encode_path() (статический метод BaseUtils), 3
- ## Z
- Zones (класс в trassir_script_framework), 35