

src

src package

Subpackages

src.browser package

Submodules

src.browser.browser module

src.browser.config module

Модуль содержит настройки для класса FirefoxBrowser.

src.browser.samsung_browser module

Module contents

src.config package

Submodules

src.config.app module

Модуль содержит настройки приложения, который меняются редко и только при внесении изменений в код приложения

```
class src.config.app.ConfigApp
```

Базовые классы: `object`

Содержит настройки приложения

```
MAX_PAGE_LOAD_TIMEOUT= 20
```

```
MAX_ERRORS_PERCENT= 30
```

```
auth
```

alias of `ConfigAuth`

person

alias of `ConfigPerson`

name

alias of `ConfigName`

name_person

alias of `ConfigNamePerson`

src.config.settings module

Module contents

src.exceptions package

Submodules

src.exceptions.parser_exceptions module

Module contents

src.interfaces package

Submodules

src.interfaces.abstract_proxy module

class `src.interfaces.abstract_proxy.AbstractProxy`

Базовые классы: `ABC`

Интерфейс для сервиса прокси

abstract
async **get_proxy()**

Тип результата:

`dict` | `None`

src.interfaces.abstract_result_parser module

class `src.interfaces.abstract_result_parser.AbstractResultParser`

Базовые классы: `ABC`

Интерфейс для парсера ответа сайта

```
abstract parse(response)
```

Обрабатывает ответ сайта, возвращает словарь с информацией по использованной сессии.

Параметры:

response (`Response`) – Ответ сайта в формате `requests.Response`.

Тип результата:

`dict`

Результат:

Словарь {«status»: «blocked»} или {«status»: «success»}, в случае ошибок возбуждает исключение или `SourceError`.

src.interfaces.abstract_samsung_source module

Модуль содержит интерфейс для получения данных с сайта Samsung.

```
class src.interfaces.abstract_samsung_source.AbstractSamsungSource(*args,  
**kwargs)
```

Базовые классы: `ABC`

Интерфейс для получения данных с сайта Samsung

```
abstract request(*args, **kwargs)
```

Тип результата:

`Response`

src.interfaces.abstract_seleniumwire_proxy module

```
class src.interfaces.abstract_seleniumwire_proxy.AbstractSeleniumWireProxy
```

Базовые классы: `ABC`

Интерфейс адаптера сервиса прокси, для использования в библиотеке `seleniumwire`

```
abstract  
async get_proxy()
```

Тип результата:

`dict` | `None`

src.interfaces.session_maker module

```
class src.interfaces.session_maker.AbstractSessionMaker
```

Базовые классы: `ABC`

Интерфейс для класса, который генерирует сессии

```
abstract  
async         prepare()
```

Выполняет подготовку браузера для получения сессии.

Тип результата:

`AbstractSessionMaker`

Результат:

SessionMaker

```
abstract  
async         make(search_data)
```

Генерирует сессию

Параметры:

`search_data` (`str` | `dict`) – Данные для поиска (заведомо не существующий аккаунт)

Тип результата:

`dict`

Результат:

Сессия

Module contents

```
class src.interfaces.AbstractProxy
```

Базовые классы: `ABC`

Интерфейс для сервиса прокси

```
abstract  
async         get_proxy()
```

Тип результата:

`dict` | `None`

```
class src.interfaces.AbstractSeleniumWireProxy
```

Базовые классы: `ABC`

Интерфейс адаптера сервиса прокси, для использования в библиотеке seleniumwire

```
abstract  
async         get_proxy()
```

Тип результата:

`dict` | `None`

```
class src.interfaces.AbstractSamsungSource(*args, **kwargs)
```

Базовые классы: `ABC`

Интерфейс для получения данных с сайта Samsung

```
abstract request(*args, **kwargs)
```

Тип результата:

`Response`

src.logger package

Submodules

src.logger.context_logger module

```
class src.logger.context_logger.ContextLogger
```

Базовые классы: `object`

Логер с контекстом

loggers

= ('seleniumwire.handler', 'seleniumwire.server', 'seleniumwire.storage', 'seleniumwire.backend', 'mongo', 'pymongo.s

```
static get_logger(name)
```

Возвращает логер по переданному имени.

Параметры:

name (`str`) – Имя логера, который требуется вернуть.

Тип результата:

`LoggingSearchKeyAdapter`

Результат:

Логер.

```
static get_root_logger()
```

Возвращает корневой логер (с именем root)

Тип результата:

`LoggingSearchKeyAdapter`

Результат:

Logger

src.logger.logger module

src.logger.logger_adapter module

```
class src.logger.logger_adapter.LoggingSearchKeyAdapter(logger, wrapper="||")
```

Базовые классы: `LoggerAdapter`

Адаптер для логера.

`process(msg, kwargs)`

Обработывает сообщение логера, добавляет контекст в сообщение, если он задан.

Параметры:

- `msg` – Сообщение логера.
- `kwargs` – Аргументы сообщения.

Результат:

Обработанное сообщение, аргументы сообщения.

`static get_context_message()`

Возвращает контекст из текущего запроса.

Тип результата:

`str`

Результат:

Контекст.

Module contents

`src.logic` package

Subpackages

`src.logic.adapters` package

Submodules

`src.logic.adapters.session_adapter` module

Module contents

`src.logic.handlers` package

Subpackages

src.logic.handlers.prolong package

Submodules

src.logic.handlers.prolong.prolong_auth module

src.logic.handlers.prolong.prolong_common module

src.logic.handlers.prolong.prolong_handler_auth module

src.logic.handlers.prolong.prolong_handler_common module

src.logic.handlers.prolong.prolong_handler_name module

src.logic.handlers.prolong.prolong_handler_person module

src.logic.handlers.prolong.prolong_name module

src.logic.handlers.prolong.prolong_person module

Module contents

Submodules

src.logic.handlers.placeholder module

src.logic.handlers.recovery module

src.logic.handlers.session_maker_mixin module

src.logic.handlers.unnecessary module

Module contents

src.logic.randomizer package

Submodules

src.logic.randomizer.randomizer module

src.logic.randomizer.randomizer.random_string(*length*)

Возвращает рандомную строку длиной length

Параметры:

length (`int`) – Длина строки

Тип результата:

`str`

Результат:

Рандомная строка

```
src.logic.randomizer.randomizer.random_fake_email(*args)
```

Возвращает не существующую почту, которая содержит ошибки в написании и с большой вероятностью никому не принадлежит

Тип результата:

str

Результат:

Не существующая почта

```
src.logic.randomizer.randomizer.get_random_string_month(*args)
```

Возвращает случайный месяц в формате строки: «Январь» - «Декабрь»

return: Месяц в формате строки

Тип результата:

str

```
src.logic.randomizer.randomizer.get_random_int_month(*args)
```

Возвращает случайный месяц в формате числа: 1 - 12

return: Месяц в формате числа

Тип результата:

int

```
src.logic.randomizer.randomizer.random_birthdate(month_strategy=<function  
get_random_string_month>)
```

Возвращает случайную дату рождения в диапазоне 1970-2004 гг.

Тип результата:

dict

Результат:

Дата рождения в формате {«day»: 1-28, «month»: Январь-Декабрь, «year»: 1970-2004}.

```
src.logic.randomizer.randomizer.random_fake_person(*args)
```

Возвращает не существующего человека для получения сессии.

Тип результата:

dict

Результат:

Словарь {«first_name»: <случайная строка длиной 10 символов>, «last_name»: <случайная строка длиной 12 символов>, «birthdate»: <дата рождения>}


```
src.logic.randomizer.randomizer.random_fake_person_to_prolong(*args)
```

Возвращает не существующего человека для пролонгации сессии person.

Тип результата:

```
dict
```

Результат:

Словарь {«first_name»: <случайная строка длиной 10 символов>, «last_name»: <случайная строка длиной 12 символов>, «birthdate»: <дата рождения>}

```
src.logic.randomizer.randomizer.random_fake_person_with_mail(*args)
```

Возвращает не существующего человека получения сессии в обработчике name.

Тип результата:

```
dict
```

Результат:

Словарь {«account_login»: <не существующая почта>, «first_name»: <случайная строка длиной 10 символов>, «last_name»: <случайная строка длиной 12 символов>, «birthdate»: <дата рождения>}

```
src.logic.randomizer.randomizer.random_fake_person_to_prolong_with_email(*args)
```

Возвращает не существующего человека для пролонгации сессии name.

Тип результата:

```
dict
```

Результат:

Словарь {«account_login»: <не существующая почта>, «first_name»: <случайная строка длиной 10 символов>, «last_name»: <случайная строка длиной 12 символов>, «birthdate»: <дата рождения>}

Module contents

```
src.logic.randomizer.random_string(length)
```

Возвращает рандомную строку длиной length

Параметры:

length (`int`) – Длина строки

Тип результата:

```
str
```

Результат:

Рандомная строка

```
src.logic.randomizer.random_fake_email(*args)
```

Возвращает не существующую почту, которая содержит ошибки в написании и с большой вероятностью никому не принадлежит

Тип результата:

str

Результат:

Не существующая почта

```
src.logic.randomizer.random_fake_person(*args)
```

Возвращает не существующего человека для получения сессии.

Тип результата:

dict

Результат:

Словарь {«first_name»: <случайная строка длиной 10 символов>, «last_name»: <случайная строка длиной 12 символов>, «birthdate»: <дата рождения>}

```
src.logic.randomizer.random_birthdate(month_strategy=<function  
get_random_string_month>)
```

Возвращает случайную дату рождения в диапазоне 1970-2004 гг.

Тип результата:

dict

Результат:

Дата рождения в формате {«day»: 1-28, «month»: Январь-Декабрь, «year»: 1970-2004}.

```
src.logic.randomizer.random_fake_person_to_prolong(*args)
```

Возвращает не существующего человека для пролонгации сессии person.

Тип результата:

dict

Результат:

Словарь {«first_name»: <случайная строка длиной 10 символов>, «last_name»: <случайная строка длиной 12 символов>, «birthdate»: <дата рождения>}

```
src.logic.randomizer.random_fake_person_to_prolong_with_email(*args)
```

Возвращает не существующего человека для пролонгации сессии name.

Тип результата:

dict

Результат:

Словарь {«account_login»: <не существующая почта>, «first_name»: <случайная строка длиной 10 символов>, «last_name»: <случайная строка длиной 12 символов>, «birthdate»: <дата рождения>}

src.logic.samsung package

Submodules

src.logic.samsung.samsung_manager module

src.logic.samsung.screen_definitions_auth module

Модуль содержит определения для экранов сайта account.samsung.com. Используются для определения страниц (главная страницы, страница с результатом) и ключевых элементов на странице (поле ввода, кнопка поиска).

src.logic.samsung.screen_definitions_name module

Модуль содержит определения для экранов сайта account.samsung.com. Используются для определения страниц (главная страницы, страница с результатом) и ключевых элементов на странице (поле ввода, кнопка поиска).

src.logic.samsung.screen_definitions_person module

Модуль содержит определения для экранов сайта account.samsung.com. Используются для определения страниц (главная страницы, страница с результатом) и ключевых элементов на странице (поле ввода, кнопка поиска).

src.logic.samsung.session_maker_auth module

src.logic.samsung.session_maker_common module

src.logic.samsung.session_maker_name module

src.logic.samsung.session_maker_person module

src.logic.samsung.session_manager_auth module

src.logic.samsung.session_manager_common module

src.logic.samsung.session_manager_name module

src.logic.samsung.session_manager_person module

Module contents

src.logic.session_storage package

Submodules

src.logic.session_storage.filter_constructor_auth module

src.logic.session_storage.filter_constructor_common module

src.logic.session_storage.filter_constructor_person module

src.logic.session_storage.storage module

Module contents

Module contents

src.parsers package

Submodules

src.parsers.result_parser_auth module

src.parsers.result_parser_name module

src.parsers.result_parser_person module

Module contents

src.proxy package

Submodules

src.proxy.proxy module

src.proxy.proxy_adapter module

Module contents

src.request_params package

Subpackages

src.request_params.api package

Submodules

src.request_params.api.samsung_auth module

src.request_params.api.samsung_name module

src.request_params.api.samsung_person module

Module contents

src.request_params.interfaces package

Submodules

src.request_params.interfaces.base module

Module contents

Module contents

src.utils package

Submodules

src.utils.utils module

```
class src.utils.utils.ExtStr
```

Базовые классы: `str`

Расширенный класс `str` с дополнительными методами.

```
inline()
```

Заменяет переносы строки пробелами.

Тип результата:

```
str
```

Результат:

Строка.

```
short(size=100)
```

Удаляет переносы строк и сокращает длину получившейся строки до указанного значения `size`.

Параметры:

size (`int`) – Максимальный размер строки

Тип результата:

`str`

Результат:

Строка

src.utils.utils.informer(*step_number, step_message*)

Выводит логи до начала работы функции и после ее завершения.

Параметры:

- **step_number** (`int` | `float`) – Номер шага
- **step_message** (`str`) – Сообщение

Тип результата:

`Callable`

Результат:

Callable

src.utils.utils.elapsed_time(*start_time*)

Возвращает разницу между переданным значением и текущим значением времени.

Параметры:

start_time (`float`) – Начало отсчета.

Тип результата:

`float`

Результат:

Разница в секундах, округленная до трех знаков после запятой.

src.utils.utils.str_to_dict(*data, separator='; '*)

Парсит строку формата <ключ>=<значение>; в словарь.

Параметры:

- **data** (`str`) – Строка формата <ключ>=<значение>.
- **separator** (`str`) – Разделитель пар <ключ>=<значение>.

Тип результата:

`dict` | `None`

Результат:

Словарь.

src.utils.utils.get_params_from_url(*url*)

Извлекает параметры запроса из URL адреса и возвращает их в виде словаря.

Параметры:

url (`str`) – URL адрес.

Тип результата:

`dict` | `None`

Результат:

Словарь из параметров запроса.

Module contents

`class src.utils.ExtStr`

Базовые классы: `str`

Расширенный класс `str` с дополнительными методами.

`inline()`

Заменяет переносы строки пробелами.

Тип результата:

`str`

Результат:

Строка.

`short(size=100)`

Удаляет переносы строк и сокращает длину получившейся строки до указанного значения `size`.

Параметры:

`size` (`int`) – Максимальный размер строки

Тип результата:

`str`

Результат:

Строка

`src.utils.informer(step_number, step_message)`

Выводит логи до начала работы функции и после ее завершения.

Параметры:

- `step_number` (`int` | `float`) – Номер шага
- `step_message` (`str`) – Сообщение

Тип результата:

`Callable`

Результат:

Callable

`src.utils.elapsed_time(start_time)`

Возвращает разницу между переданным значением и текущим значением времени.

Параметры:

`start_time (float)` – Начало отсчета.

Тип результата:

`float`

Результат:

Разница в секундах, округленная до трех знаков после запятой.

```
src.utils.str_to_dict(data, separator=';')
```

Парсит строку формата <ключ>=<значение>; в словарь.

Параметры:

- `data (str)` – Строка формата <ключ>=<значение>.
- `separator (str)` – Разделитель пар <ключ>=<значение>.

Тип результата:

`dict` | `None`

Результат:

Словарь.

```
src.utils.get_params_from_url(url)
```

Извлекает параметры запроса из URL адреса и возвращает их в виде словаря.

Параметры:

`url (str)` – URL адрес.

Тип результата:

`dict` | `None`

Результат:

Словарь из параметров запроса.

Module contents

start module

start_auth module

start_name module

start_person module