
Yandex Music API

февр. 04, 2020

1	[Pre-Alpha] Неофициальная Python библиотека для API Yandex Music	1
1.1	Содержание	1
1.2	Введение	2
1.3	Установка	2
1.4	Начало работы	2
1.5	Документация	5
1.6	Получение помощи	5
1.7	Благодарность	5
1.8	Лицензия	5
2	Пакет yandex_music	7
2.1	yandex_music.Client	7
2.2	yandex_music.YandexMusicObject	31
2.3	yandex_music.Cover	31
2.4	yandex_music.DownloadInfo	33
2.5	yandex_music.Experiments	35
2.6	yandex_music.Icon	36
2.7	yandex_music.InvocationInfo	37
2.8	yandex_music.Pager	37
2.9	yandex_music.PermissionAlerts	38
2.10	yandex_music.PromoCodeStatus	39
2.11	yandex_music.Settings	40
2.12	yandex_music.TrackShort	41
2.13	yandex_music.TracksList	43
2.14	yandex_music.Video	44
2.15	Пакет yandex_music.account	45
2.16	Пакет yandex_music.album	60
2.17	Пакет yandex_music.artist	66
2.18	Пакет yandex_music.feed	75
2.19	Пакет yandex_music.genre	82
2.20	Пакет yandex_music.landing	86
2.21	Пакет yandex_music.likes	95
2.22	Пакет yandex_music.playlist	95
2.23	Пакет yandex_music.rotor	103
2.24	Пакет yandex_music.search	116
2.25	Пакет yandex_music.shot	120

2.26	Пакет <code>yandex_music.supplement</code>	125
2.27	Пакет <code>yandex_music.track</code>	129
2.28	Пакет <code>yandex_music.utils</code>	132
3	Список изменений	135
3.1	Версия 0.0.16	135
3.2	Версия 0.0.15	136
3.3	Версия 0.0.14	137
	Алфавитный указатель	139

[Pre-Alpha] Неофициальная Python библиотека для API Yandex Music

Делаю то, что по определённым причинам не сделала компания Yandex.

Маленькое сообщество разработчиков общаются и помогают друг другу в [Telegram чатике](#), присоединяйтесь!

чат-blue.svg

1.1 Содержание

- *Введение*
 1. *Доступ к вашим данным на Яндексе*
- *Установка*
- *Начало работы*
 1. *Изучение по примерам*
 2. *Логирование*
 3. *Документация*
- *Получение помощи*

- *Благодарность*
- *Лицензия*

1.2 Введение

Эта библиотека предоставляется Python интерфейс для никем незадокументированного и сделанного только для себя API Яндекс Музыка.

Она совместима с версиями Python 3.6+.

В дополнение к реализации чистого API данная библиотека имеет ряд классов-обёрток объектов высокого уровня дабы сделать разработку клиентов и скриптов простой и понятной.

1.2.1 Доступ к вашим данным на Яндексе

Значения констант `CLIENT_ID` и `CLIENT_SECRET` позаимствовано у официального приложения-клиента сервиса Яндекс.Музыка из магазина Microsoft Store. Так как API является закрытым и используется только внутри компании Яндекс сейчас невозможно зарегистрировать своё собственное приложение на `oauth.yandex.ru`, а следовательно, использовать свои значения констант.

1.3 Установка

Библиотека находится в стадии разработки

Вы можете установить или обновить `yandex-music-api` при помощи:

```
$ pip install yandex-music --upgrade
```

Или Вы можете установить из исходного кода с помощью:

```
$ git clone https://github.com/MarshalX/yandex-music-api --recursive
$ cd yandex-music-api
$ python setup.py install
```

1.4 Начало работы

Приступив к работе первым делом необходимо создать экземпляр клиента.

Инициализация клиента:

```
from yandex_music.client import Client

client = Client()
```

Для доступа к своим личным данным следует авторизоваться. Это можно осуществить через OAuth токен или логин с паролем.

Авторизация по логину и паролю:

```
from yandex_music.client import Client

client = Client.from_credentials('example@yandex.com', 'password')
```

Авторизация по токену:

```
from yandex_music.client import Client

client = Client.from_token('token')
# или
client = Client('token')
```

После успешного создания клиента Вы вольны в выборе необходимого метода из API. Все они доступны у объекта класса Client. Подробнее в методах клиента в [документации](#).

Пример получения первого трека из плейлиста «Мне нравится» и его загрузка:

```
from yandex_music.client import Client

client = Client.from_credentials('example@yandex.com', 'password')
client.users_likes_tracks()[0].track.download('example.mp3')
```

В примере выше клиент получает список треков которые были отмечены как понравившиеся. API возвращает объект `TracksList` в котором содержится список с треками класса `TrackShort`. Данный класс содержит самую важную информацию о треке и никаких подробностей, поэтому для получения полной версии трека со всей информацией необходимо обратиться к свойству `track`. Затем можно скачать трек методом `download()`.

Пример получения треков по ID:

```
from yandex_music.client import Client

client = Client()
client.tracks(['10994777:1193829', '40133452:5206873', '48966383:6693286', '51385674:7163467'])
```

В качестве ID трека выступает его уникальный номер и номер альбома. Первым треком из примера является следующий трек: music.yandex.ru/album/1193829/track/10994777

Выполнение запросов с использованием прокси:

```
from yandex_music.utils.request import Request
from yandex_music.client import Client

request = Request(proxy_url='socks5://user:password@host:port')
client = Client(request=request)
```

Примеры proxy url:

- socks5://user:password@host:port
- http://host:port
- https://host:port
- http://user:password@host

Больше примеров тут: [proxies - advanced usage - requests](#)

Пример инициализации клиента с обработкой капчи:

```
def init_client():
    client = captcha_key = captcha_answer = None
    while not client:
        try:
            client = Client.from_credentials('login', 'pass', captcha_answer, captcha_key)
        except Captcha as e:
            e.captcha.download('captcha.png')

            captcha_key = e.captcha.x_captcha_key
            captcha_answer = input('Число с картинки: ')

    return client
```

Пример инициализации клиента с обработкой капчи при помощи callback-функции:

```
def proc_captcha(captcha):
    captcha.download('captcha.png')
    return input('Число с картинки: ')

client = Client.from_credentials('login', 'pass', captcha_callback=proc_captcha)
```

1.4.1 Изучение по примерам

Вот несколько примеров для обзора. Даже если это не Ваш подход к обучению, пожалуйста, возьмите и бегло просмотрите их.

Код примеров опубликован в открытом доступе, поэтому Вы можете взять его и начать писать вокруг своё.

Посетите [эту страницу](#) чтобы изучить официальные примеры.

1.4.2 Логирование

Данная библиотека использует logging модуль. Чтобы настроить логирование на стандартный вывод, поместите

```
import logging
logging.basicConfig(level=logging.DEBUG,
                    format='%(asctime)s - %(name)s - %(levelname)s - %(message)s')
```

в начало вашего скрипта.

Вы также можете использовать логирование в вашем приложении, вызвав `logging.getLogger()` и установить уровень какой Вы хотите:

```
logger = logging.getLogger()
logger.setLevel(logging.INFO)
```

Если Вы хотите DEBUG логирование:

```
logger.setLevel(logging.DEBUG)
```


1.5 Документация

Документация `yandex-music-api` находится в стадии написания и расположена на readthedocs.io.

1.6 Получение помощи

Получить помощь можно несколькими путями:

- Задать вопрос в [Telegram чатике](#), где мы помогаем друг другу, присоединяйтесь!
- Сообщить о баге, предложить новую фичу или задать вопрос можно [создав issue](#).
- Найти ответ на вопрос в [документации библиотеки](#).

1.7 Благодарность

Спасибо разработчикам `python-telegram-bot`. Выбрал Вас в качестве примера.

1.8 Лицензия

Вы можете копировать, распространять и модифицировать программное обеспечение при условии, что модификации описаны и лицензированы бесплатно в соответствии с [LGPL-3](#). Произведения производных (включая модификации или что-либо статически связанное с библиотекой) могут распространяться только в соответствии с [LGPL-3](#), но приложения, которые используют библиотеку, необязательно.

2.1 `yandex_music.Client`

```
class yandex_music.Client(token: str = None, fetch_account_status: bool = True,
                        base_url: str = None, oauth_url: str = None, request:
                        yandex_music.utils.request.Request = None)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий клиент Yandex Music.

Примечание: При `fetch_account_status = False` многие сокращения перестанут работать в связи с тем, что неоткуда будет взять `uid` аккаунта для отправки запроса. Так же в большинстве методов придётся передавать `uid` явно.

`logger`

Объект логера.

Type `logging.Logger`

`token`

Уникальный ключ для аутентификации.

Type `str`

`base_url`

Ссылка на API Yandex Music.

Type `str`

`oauth_url`

Ссылка на OAuth Yandex Music.

Type `str`

me

Объект класса `yandex_music.Status` представляющего основную информацию об аккаунте.

Типе `yandex_music.Status`

Параметры

- `token` (`str`, optional) – Уникальный ключ для аутентификации.
- `fetch_account_status` (`bool`, optional) – Получить ли информацию об аккаунте при инициализации объекта.
- `base_url` (`str`, optional) – Ссылка на API Yandex Music.
- `oauth_url` (`str`, optional) – Ссылка на OAuth Yandex Music.
- `request` (`yandex_music.utils.request.Request`, optional) – Пре-инициализация `yandex_music.utils.request.Request`.

`accountExperiments(timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.experiments.Experiments]`
 Псевдоним для `account_experiments`

`accountStatus(timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.account.status.Status]`
 Псевдоним для `account_status`

`account_experiments(timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.experiments.Experiments]`
 Получение значений экспериментальных функций аккаунта.

Параметры

- `timeout` (`int | float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Experiments` представляющий состояния экспериментальных функций, иначе `None`.

Тип результата `yandex_music.Experiments`

Raises `yandex_music.YandexMusicError`

`account_settings(timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.account.user_settings.UserSettings]`
 Получение настроек текущего пользователя.

Параметры

- `timeout` (`int | float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.UserSettings` представляющий настройки пользователя, иначе `None`.

Тип результата `yandex_music.UserSettings`

Raises `yandex_music.YandexMusicError`

```
account_settings_set(param: str = None, value: Union[str, int, bool] =
                    None, data: Dict[str, Union[str, int, bool]] = None,
                    timeout: Union[int, float] = None, *args, **kwargs) →
                    Optional[yandex_music.account.user_settings.UserSettings]
```

Изменение настроек текущего пользователя.

Примечание: Доступные названия параметров есть поля в классе `yandex_music.UserSettings`, только в CamelCase.

Параметры

- `param (str)` – Название параметра для изменения.
- `value (str | int | bool)` – Значение параметра.
- `data (dict)` – Словарь параметров и значений для множественного изменения.
- `timeout (int | float, optional)` – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs (dict, optional)` – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.UserSettings` представляющий настройки пользователя, иначе `None`.

Тип результата `yandex_music.UserSettings`

Raises `yandex_music.YandexMusicError`

```
account_status(timeout: Union[int, float] = None, *args, **kwargs) →
                Optional[yandex_music.account.status.Status]
```

Получение статуса аккаунта. Нет обязательных параметров.

Параметры

- `timeout (int | float, optional)` – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs (dict, optional)` – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Status` представляющий информацию об аккаунте если валиден, иначе `None`.

Тип результата `yandex_music.Status`

Raises `yandex_music.YandexMusicError`

```
afterTrack(next_track_id: Union[str, int], context_item: str, prev_track_id: Union[str, int]
           = None, context: str = 'playlist', types: str = 'shot', from_: str = 'mobile-
           landing-origin-default', timeout: Union[int, float] = None, *args, **kwargs) →
           Optional[yandex_music.shot.shot_event.ShotEvent]
```

Псевдоним для `after_track`

```
after_track(next_track_id: Union[str, int], context_item: str, prev_track_id: Union[str, int]
            = None, context: str = 'playlist', types: str = 'shot', from_: str = 'mobile-
            landing-origin-default', timeout: Union[int, float] = None, *args, **kwargs) →
            Optional[yandex_music.shot.shot_event.ShotEvent]
```

Получение рекламы или шота от Алисы после трека.

Примечание: При получении шота от Алисы *prev_track_id* можно не указывать.

Если *context* = „playlist“, то в *context_item* необходимо передать *{OWNER_PLAYLIST}:{ID_PLAYLIST}*. Плейлист с Алисой имеет владельца с *id* = 940441070.

ID плейлиста можно получить из блоков landing’a. Получить шот чужого плейлиста нельзя.

Известные значения *context*: *playlist*.

Известные значения *types*: *shot*, *ad*.

Параметры

- *prev_track_id* (*str* | *int*) – Уникальный идентификатор предыдущего трека.
- *next_track_id* (*str* | *int*) – Уникальный идентификатор следующего трека.
- *context_item* (*str*) – Уникальный идентификатор контекста.
- *context* (*str*, optional) – Место, откуда было вызвано получение.
- *types* (*str*, optional) – Тип того, что вернуть после трека.
- *from* (*str*, optional) – Место, с которого попали в контекст.
- *timeout* (*int* | *float*, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- ***kwargs* (*dict*, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса *yandex_music.ShotEvent* представляющий шоты от Алисы, иначе *None*.

Тип результата *yandex_music.ShotEvent*

Raises *yandex_music.YandexMusicError*

albums(*album_ids*: *Union[List[Union[str, int]], int, str]*, *timeout*: *Union[int, float]* = *None*, **args*, ***kwargs*) → *List[yandex_music.album.album.Album]*

albumsWithTracks(*album_id*: *Union[str, int]*, *timeout*: *Union[int, float]* = *None*, **args*, ***kwargs*) → *Optional[yandex_music.album.album.Album]*

Псевдоним для *albums_with_tracks*

albums_with_tracks(*album_id*: *Union[str, int]*, *timeout*: *Union[int, float]* = *None*, **args*, ***kwargs*) → *Optional[yandex_music.album.album.Album]*

Получение альбома по его уникальному идентификатору вместе с треками.

Параметры

- *album_id* (*str* | *int*) – Уникальный идентификатор альбома.
- *timeout* (*int* | *float*, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- ***kwargs* (*dict*, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объект класса *yandex_music.Album* представляющий альбом, иначе *None*.

Тип результата *list* из *yandex_music.Album*

Raises `yandex_music.YandexMusicError`

`artists(artist_ids: Union[List[Union[str, int]], int, str], timeout: Union[int, float] = None, *args, **kwargs) → List[yandex_music.artist.artist.Artist]`

`artistsBriefInfo(artist_id: Union[str, int], timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.artist.brief_info.BriefInfo]`

Псевдоним для `artists_brief_info`

`artistsDirectAlbums(artist_id: Union[str, int], page: int = 0, page_size: int = 20, sort_by: str = 'year', timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.artist.artist_albums.ArtistAlbums]`

Псевдоним для `artists_direct_albums`

`artistsTracks(artist_id: Union[str, int], page: int = 0, page_size: int = 20, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.artist.artist_tracks.ArtistTracks]`

Псевдоним для `artists_tracks`

`artists_brief_info(artist_id: Union[str, int], timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.artist.brief_info.BriefInfo]`

`artists_direct_albums(artist_id: Union[str, int], page: int = 0, page_size: int = 20, sort_by: str = 'year', timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.artist.artist_albums.ArtistAlbums]`

Получение альбомов артиста.

Примечание: Известные значения для `sort_by`: `year`, `rating`.

Параметры

- `artist_id` (`str` | `int`) – Уникальный идентификатор артиста.
- `page` (`int`, optional) – Номер страницы.
- `page_size` (`int`, optional) – Количество альбомов на странице.
- `sort_by` (`str`, optional) – Параметр для сортировки.
- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.ArtistsTracks` представляющий страницу списка альбомов артиста, иначе `None`.

Тип результата `yandex_music.ArtistAlbums`

Raises `yandex_music.YandexMusicError`

`artists_tracks(artist_id: Union[str, int], page: int = 0, page_size: int = 20, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.artist.artist_tracks.ArtistTracks]`

Получение треков артиста.

Параметры

- `artist_id` (`str` | `int`) – Уникальный идентификатор артиста.
- `page` (`int`, optional) – Номер страницы.

- `page_size` (int, optional) – Количество треков на странице.
- `timeout` (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.ArtistsTracks` представляющий страницу списка треков артиста, иначе `None`.

Тип результата `yandex_music.ArtistsTracks`

Raises `yandex_music.YandexMusicError`

```
consumePromoCode(code: str, language: str = 'en', timeout:
                  Union[int, float] = None, *args, **kwargs) →
                  Optional[yandex_music.promo_code_status.PromoCodeStatus]
```

Псевдоним для `consume_promo_code`

```
consume_promo_code(code: str, language: str = 'en', timeout:
                   Union[int, float] = None, *args, **kwargs) →
                   Optional[yandex_music.promo_code_status.PromoCodeStatus]
```

Активация промо-кода.

Параметры

- `code` (str) – Промо-код.
- `language` (str, optional) – Язык ответа API в ISO 639-1.
- `timeout` (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.PromoCodeStatus` представляющий информацию об активации промо-кода, иначе `None`.

Тип результата `yandex_music.PromoCodeStatus`

Raises `yandex_music.YandexMusicError`

```
feed(timeout: Union[int, float] = None, *args, **kwargs) →
      Optional[yandex_music.feed.feed.Feed]
```

Получение потока информации (фида) подобранного под пользователя. Содержит умные плейлисты.

Параметры

- `timeout` (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Feed` представляющий умные плейлисты пользователя, иначе `None`.

Тип результата `yandex_music.Feed`

Raises `yandex_music.YandexMusicError`

```
feedWizardIsPassed(timeout: Union[int, float] = None, *args, **kwargs) → bool
```

Псевдоним для `feed_wizard_is_passed`


```
feed_wizard_is_passed(timeout: Union[int, float] = None, *args, **kwargs) → bool
classmethod fromCredentials(username: str, password: str, x_captcha_answer: str =
    None, x_captcha_key: str = None, captcha_callback:
    Callable[[yandex_music.exceptions.Captcha], str] = None,
    *args, **kwargs) → yandex_music.client.Client
```

Псевдоним для `from_credentials`

```
classmethod fromToken(token: str, *args, **kwargs) → yandex_music.client.Client
```

Псевдоним для `from_token`

```
classmethod from_credentials(username: str, password: str, x_captcha_answer: str =
    None, x_captcha_key: str = None, captcha_callback:
    Callable[[yandex_music.exceptions.Captcha], str] = None,
    *args, **kwargs) → yandex_music.client.Client
```

Инициализация клиента по логину и паролю.

Примечание: Данный метод получает токен каждый раз при вызове. Рекомендуется сгенерировать его самостоятельно, сохранить и использовать при следующих инициализациях клиента. Не храните логины и пароли!

Параметры

- `username` (`str`) – Логин клиента (идентификатор).
- `password` (`str`) – Пароль клиента (аутентификатор).
- `x_captcha_answer` (`str`, optional) – Ответ на капчу (цифры с картинки).
- `x_captcha_key` (`str`, optional) – Уникальный ключ капчи.
- `captcha_callback` (`function`, optional) – Функция обратного вызова для обработки капчи, должна принимать объект класса `yandex_music.exceptions.Captcha` и возвращать проверочный код.
- `**kwargs` (`dict`, optional) – Аргументы для конструктора клиента.

Результат `yandex_music.Client`.

Raises `yandex_music.YandexMusicError`

```
classmethod from_token(token: str, *args, **kwargs) → yandex_music.client.Client
```

Инициализация клиента по токenu.

Примечание: Ничем не отличается от `Client(token)`. Так исторически сложилось.

Параметры

- `token` (`str`, optional) – Уникальный ключ для аутентификации.
- `**kwargs` (`dict`, optional) – Аргументы для конструктора клиента.

Результат `yandex_music.Client`.

```
generateTokenByUsernameAndPassword(username: str, password: str, grant_type: str
    = 'password', x_captcha_answer: str = None,
    x_captcha_key: str = None, timeout: Union[int, float]
    = None, *args, **kwargs) → str
```

Псевдоним для `generate_token_by_username_and_password`

```
generate_token_by_username_and_password(username: str, password: str, grant_type: str
    = 'password', x_captcha_answer: str = None,
    x_captcha_key: str = None, timeout: Union[int,
    float] = None, *args, **kwargs) → str
```

Метод получения OAuth токена по логину и паролю.

Параметры

- `username` (str) – Логин клиента (идентификатор).
- `password` (str) – Пароль клиента (аутентификатор).
- `grant_type` (str, optional) – Тип разрешения OAuth.
- `x_captcha_answer` (str, optional) – Ответ на капчу (цифры с картинки).
- `x_captcha_key` (str, optional) – Уникальный ключ капчи.
- `timeout` (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат OAuth токен.

Тип результата str

Raises `yandex_music.YandexMusicError`

```
genres(timeout: Union[int, float] = None, *args, **kwargs) →
    List[yandex_music.genre.genre.Genre]
```

Получение жанров музыки.

Параметры

- `timeout` (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Список объектов класса `yandex_music.Genre` представляющих жанры музыки, иначе None.

Тип результата list из `yandex_music.Genre`

Raises `yandex_music.YandexMusicError`

```
landing(blocks: Union[str, List[str]], timeout: Union[int, float] = None, *args, **kwargs) →
    Optional[yandex_music.landing.landing.Landing]
```

Получение лендинг-страницы содержащий блоки с новыми релизами, чартами, плейлистами с новинками и т.д.

Примечание: Поддерживаемые типы блоков: `personalplaylists`, `promotions`, `new-releases`, `new-playlists`, `mixes`, `chart`, `artists`, `albums`, `playlists`, `play_contexts`.

Параметры

- `blocks` (`str` | `list` из `str`) – Блок или список блоков необходимых для выдачи.
- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Landing` представляющий лендинг-страницу, иначе `None`.

Тип результата `yandex_music.Landing`

Raises `yandex_music.YandexMusicError`

```
permissionAlerts(timeout: Union[int, float] = None, *args, **kwargs) →
Optional[yandex_music.permission_alerts.PermissionAlerts]
Псевдоним для permission_alerts
```

```
permission_alerts(timeout: Union[int, float] = None, *args, **kwargs) →
Optional[yandex_music.permission_alerts.PermissionAlerts]
Получение оповещений. Нет обязательных параметров.
```

Параметры

- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.PermissionAlerts` представляющий оповещения, иначе `None`.

Тип результата `yandex_music.PermissionAlerts`

Raises `yandex_music.YandexMusicError`

```
playAudio(track_id: Union[str, int], from_: str, album_id: Union[str, int], playlist_id: str
= None, from_cache: bool = False, play_id: str = None, uid: int = None,
timestamp: str = None, track_length_seconds: int = 0, total_played_seconds: int
= 0, end_position_seconds: int = 0, client_now: str = None, timeout: Union[int,
float] = None, *args, **kwargs) → bool
Псевдоним для play_audio
```

```
play_audio(track_id: Union[str, int], from_: str, album_id: Union[str, int], playlist_id: str
= None, from_cache: bool = False, play_id: str = None, uid: int = None,
timestamp: str = None, track_length_seconds: int = 0, total_played_seconds: int
= 0, end_position_seconds: int = 0, client_now: str = None, timeout: Union[int,
float] = None, *args, **kwargs) → bool
```

Метод для отправки текущего состояния прослушиваемого трека.

Параметры

- `track_id` (`str` | `int`) – Уникальный идентификатор трека.
- `from` (`str`) – Наименования клиента с которого происходит прослушивание.
- `album_id` (`str` | `int`) – Уникальный идентификатор альбома.
- `playlist_id` (`str`, optional) – Уникальный идентификатор плейлиста, если таковой прослушивается.

- `from_cache` (bool, optional) – Проигрывается ли трек с кеша.
- `play_id` (str, optional) – Уникальный идентификатор проигрывания.
- `uid` (int, optional) – Уникальный идентификатор пользователя.
- `timestamp` (str, optional) – Текущая дата и время в ISO.
- `track_length_seconds` (int, optional) – Продолжительность трека в секундах.
- `total_played_seconds` (int, optional) – Сколько было всего воспроизведено трека в секундах.
- `end_position_seconds` (int, optional) – Окончательное значение воспроизведенных секунд.
- `client_now` (str, optional) – Текущая дата и время клиента в ISO.
- `timeout` (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат True при успешном выполнении запроса, иначе False.

Тип результата bool

Raises `yandex_music.YandexMusicError`

`playlistsList`(*playlist_ids*: Union[List[Union[str, int]], int, str], *timeout*: Union[int, float] = None, *args, **kwargs) → List[yandex_music.playlist.playlist.Playlist]
 Псевдоним для `playlists_list`

`playlists_list`(*playlist_ids*: Union[List[Union[str, int]], int, str], *timeout*: Union[int, float] = None, *args, **kwargs) → List[yandex_music.playlist.playlist.Playlist]

`request`

Объект вспомогательного класса для отправки запросов.

Типе `yandex_music.utils.request.Request`

`rotorAccountStatus`(*timeout*: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.account.status.Status]
 Псевдоним для `rotor_account_status`

`rotorStationFeedback`(*station*: str, *type_*: str, *timestamp*: Union[str, float, int] = None, *from_*: str = None, *batch_id*: str = None, *total_played_seconds*: Union[int, float] = None, *track_id*: Union[str, int] = None, *timeout*: Union[int, float] = None, *args, **kwargs) → bool
 Псевдоним для `rotor_station_feedback`

`rotorStationFeedbackRadioStarted`(*station*: str, *from_*: str, *batch_id*: str = None, *timestamp*: Union[str, float, int] = None, *timeout*: Union[int, float] = None, *args, **kwargs) → bool
 Псевдоним для `rotor_station_feedback_radio_started`

`rotorStationFeedbackSkip`(*station*: str, *track_id*: Union[str, int], *total_played_seconds*: float, *batch_id*: str = None, *timestamp*: Union[str, float, int] = None, *timeout*: Union[int, float] = None, *args, **kwargs) → bool
 Псевдоним для `rotor_station_feedback_skip`

```

rotorStationFeedbackTrackFinished(station: str, track_id: Union[str, int],
                                   total_played_seconds: float, batch_id: str = None,
                                   timestamp: Union[str, float, int] = None, timeout:
                                   Union[int, float] = None, *args, **kwargs) → bool
    Псевдоним для rotor_station_feedback_track_finished

rotorStationFeedbackTrackStarted(station: str, track_id: Union[str, int], batch_id: str =
                                   None, timestamp: Union[str, float, int] = None, timeout:
                                   Union[int, float] = None, *args, **kwargs) → bool
    Псевдоним для rotor_station_feedback_track_started

rotorStationInfo(station: str, timeout: Union[int, float] = None, *args, **kwargs) →
    List[yandex_music.rotor.station_result.StationResult]
    Псевдоним для rotor_station_info

rotorStationSettings2(station: str, mood_energy: str, diversity: str, language: str = 'not-
    russian', timeout: Union[int, float] = None, *args, **kwargs) → bool
    Псевдоним для rotor_station_settings2

rotorStationTracks(station: str, settings2: bool = True, queue: Union[str, int] =
    None, timeout: Union[int, float] = None, *args, **kwargs) →
    Optional[yandex_music.rotor.station_tracks_result.StationTracksResult]
    Псевдоним для rotor_station_tracks

rotorStationsDashboard(timeout: Union[int, float] = None, *args, **kwargs) →
    Optional[yandex_music.rotor.dashboard.Dashboard]
    Псевдоним для rotor_stations_dashboard

rotorStationsList(language: str = 'ru', timeout: Union[int, float] = None, *args, **kwargs) →
    List[yandex_music.rotor.station_result.StationResult]
    Псевдоним для rotor_stations_list

rotor_account_status(timeout: Union[int, float] = None, *args, **kwargs) →
    Optional[yandex_music.account.status.Status]
    Получение статуса пользователя с дополнительными полями.

```

Примечание: Данный статус отличается от обычного наличием дополнительных полей, например, `skips_per_hour`.

Параметры

- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Status` представляющий статус пользователя с дополнительными полями от радио, иначе `None`.

Тип результата `yandex_music.Status`

Raises `yandex_music.YandexMusicError`

```

rotor_station_feedback(station: str, type_: str, timestamp: Union[str, float, int] = None,
                       from_: str = None, batch_id: str = None, total_played_seconds:
                       Union[int, float] = None, track_id: Union[str, int] = None, timeout:
                       Union[int, float] = None, *args, **kwargs) → bool
    Отправка ответной реакции на происходящее при прослушивании радио.

```

Примечание: Сообщения о начале прослушивания радио, начале и конце трека, его пропуса.

Известные типы фидбека: *radioStarted*, *trackStarted*, *trackFinished*, *skip*.

Пример *station*: *user:onyourwave*, *genre:allrock*.

Пример *from_*: *mobile-radio-user-123456789*.

Параметры

- `station (str)` – Станция.
- `type (str)` – Тип отправляемого фидбека.
- `timestamp (str | float | int, optional)` – Текущее время и дата.
- `from (str, optional)` – Откуда начато воспроизведение радио.
- `batch_id (str, optional)` – Уникальный идентификатор партии треков. Возвращается при получении треков.
- `total_played_seconds (int | float, optional)` – Сколько было проиграно секунд трека перед действием.
- `track_id (int | str, optional)` – Уникальный идентификатор трека.
- `timeout (int | float, optional)` – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs (dict, optional)` – Произвольные аргументы (будут переданы в запрос).

Результат True при успешном выполнении запроса, иначе False.

Тип результата bool

Raises `yandex_music.YandexMusicError`

```
rotor_station_feedback_radio_started(station: str, from_: str, batch_id: str = None,
                                     timestamp: Union[str, float, int] = None, timeout:
                                     Union[int, float] = None, *args, **kwargs) → bool
```

Сокращение для:

```
client.rotor_station_feedback(station, 'radioStarted', timestamp, from, *args, **kwargs)
```

Результат True при успешном выполнении запроса, иначе False.

Тип результата bool

Raises `yandex_music.YandexMusicError`

```
rotor_station_feedback_skip(station: str, track_id: Union[str, int], total_played_seconds:
                             float, batch_id: str = None, timestamp: Union[str, float, int]
                             = None, timeout: Union[int, float] = None, *args, **kwargs)
                             → bool
```

Сокращение для:

```
client.rotor_station_feedback(station, 'skip', timestamp, track_id, total_played_seconds,
                               *args, **kwargs)
```

Результат True при успешном выполнении запроса, иначе False.

Тип результата bool

Raises `yandex_music.YandexMusicError`

```
rotor_station_feedback_track_finished(station: str, track_id: Union[str, int],
                                     total_played_seconds: float, batch_id: str =
                                     None, timestamp: Union[str, float, int] = None,
                                     timeout: Union[int, float] = None, *args, **kwargs)
                                     → bool
```

Сокращение для:

```
client.rotor_station_feedback(station, 'trackFinished', timestamp, track_id, total_
↳played_seconds,
*args, **kwargs)
```

Результат True при успешном выполнении запроса, иначе False.

Тип результата bool

Raises `yandex_music.YandexMusicError`

```
rotor_station_feedback_track_started(station: str, track_id: Union[str, int], batch_id: str
= None, timestamp: Union[str, float, int] = None,
timeout: Union[int, float] = None, *args, **kwargs)
→ bool
```

Сокращение для:

```
client.rotor_station_feedback(station, 'trackStarted', timestamp, track_id, *args,
↳**kwargs)
```

Результат True при успешном выполнении запроса, иначе False.

Тип результата bool

Raises `yandex_music.YandexMusicError`

```
rotor_station_info(station: str, timeout: Union[int, float] = None, *args, **kwargs) →
List[yandex_music.rotor.station_result.StationResult]
```

Получение информации о станции и пользовательских настроек на неё.

Параметры

- `station` (`str`) – Станция.
- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Список из одного объекта класса `yandex_music.StationResult` представляющего информацию о станции, иначе `None`.

Тип результата list из `yandex_music.StationResult`

Raises `yandex_music.YandexMusicError`

```
rotor_station_settings2(station: str, mood_energy: str, diversity: str, language: str = 'not-
russian', timeout: Union[int, float] = None, *args, **kwargs) → bool
```

Изменение настроек определённой станции.

Примечание: Доступные значения для *mood_energy*: *fun, active, calm, sad, all*.

Доступные значения для *diversity*: *favorite, popular, discover, default*.

Доступные значения для *language*: *not-russian, russian, any*.

У станций в *restrictions* есть Enum'ы, а в них *possible_values* - доступные значения для поля.

Параметры

- `station (str)` – Станция.
- `mood_energy (str)` – Настроение.
- `diversity (str)` – Треки.
- `language (str)` – Язык.
- `timeout (int | float, optional)` – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs (dict, optional)` – Произвольные аргументы (будут переданы в запрос).

Результат `True` при успешном выполнении запроса, иначе `False`.

Тип результата `bool`

Raises `yandex_music.YandexMusicError`

```
rotor_station_tracks(station: str, settings2: bool = True, queue: Union[str, int] =  
    None, timeout: Union[int, float] = None, *args, **kwargs) →  
    Optional[yandex_music.rotor.station_tracks_result.StationTracksResult]
```

Получение цепочки треков определённой станции.

Примечание: Для продолжения цепочки треков необходимо:

1. Передавать *ID* трека, что был до этого (первый в цепочки).
2. Отправить фидбек о конце или скипе трека, что был передан в *queue*.
3. Отправить фидбек о начале следующего трека (второй в цепочки).
4. Выполнить запрос получения треков. В ответе придёт новые треки или произойдёт сдвиг цепочки на 1 элемент.

Проход по цепочке до конца не изучен. Часто встречаются дубликаты.

Все официальные клиенты выполняют запросы с *settings2 = True*.

Параметры

- `station (str)` – Станция.
- `settings2 (bool, optional)` – Использовать ли второй набор настроек.
- `queue (str | int, optional)` – Уникальный идентификатор трека, который только что был.
- `timeout (int | float, optional)` – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.

- **kwargs** (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.StationTracksResult` представляющий последовательность треков станции, иначе `None`.

Тип результата `yandex_music.StationTracksResult`

Raises `yandex_music.YandexMusicError`

`rotor_stations_dashboard(timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.rotor.dashboard.Dashboard]`

Получение рекомендованных станций текущего пользователя.

Параметры

- **timeout** (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- **kwargs** (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Dashboard` представляющий рекомендованные станции, иначе `None`.

Тип результата `yandex_music.Dashboard`

Raises `yandex_music.YandexMusicError`

`rotor_stations_list(language: str = 'ru', timeout: Union[int, float] = None, *args, **kwargs) → List[yandex_music.rotor.station_result.StationResult]`

Получение всех радиостанций с настройками пользователя.

Примечание: Чтобы определить что за тип станции (жанры, настроения, занятие и т.д.) необходимо смотреть в поле `id_for_from`.

Параметры

- **language** (str) – Язык, на котором будет информация о станциях.
- **timeout** (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- **kwargs** (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Список объектов класса `yandex_music.StationResult` представляющих станцию, иначе `None`.

Тип результата list из `yandex_music.StationResult`

Raises `yandex_music.YandexMusicError`

`search(text: str, nocorrect: bool = False, type_: str = 'all', page: int = 0, playlist_in_best: bool = True, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.search.search.Search]`

Осуществление поиска по запросу и типу, получение результатов.

Параметры

- **text** (str) – Текст запроса.

- `nocorrect` (bool) – Без исправлений ли TODO.
- `type` (str) – Среди какого типа искать (трек, плейлист, альбом, исполнитель).
- `page` (int) – Номер страницы.
- `playlist_in_best` (bool) – Выдавать ли плейлисты лучшим вариантом поиска.
- `timeout` (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Search` представляющий результаты поиска, иначе `None`.

Тип результата `yandex_music.Search`

Raises `yandex_music.YandexMusicError`

`searchSuggest(part: str, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.search.suggestions.Suggestions]`

Псевдоним для `search_suggest`

`search_suggest(part: str, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.search.suggestions.Suggestions]`

Получение подсказок по введенной части поискового запроса.

Параметры

- `part` (str) – Часть поискового запроса.
- `timeout` (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Suggestions` представляющий подсказки для запроса, иначе `None`.

Тип результата `yandex_music.Suggestions`

Raises `yandex_music.YandexMusicError`

`settings(timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.settings.Settings]`

Получение предложений по покупке. Нет обязательных параметров.

Параметры

- `timeout` (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Settings` представляющий информацию о предлагаемых продуктах, иначе `None`.

Тип результата `yandex_music.Settings`

Raises `yandex_music.YandexMusicError`

```
trackSupplement(track_id: Union[str, int], timeout: Union[int, float] = None, *args, **kwargs)
    → Optional[yandex_music.supplement.supplement.Supplement]
```

Псевдоним для `track_supplement`

```
track_supplement(track_id: Union[str, int], timeout: Union[int, float] = None, *args, **kwargs)
    → Optional[yandex_music.supplement.supplement.Supplement]
```

Получение дополнительной информации о треке.

Параметры

- `track_id` (str) – Уникальный идентификатор трека.
- `timeout` (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объект класса `yandex_music.Supplement` представляющий дополнительную информацию о треке.

Тип результата `yandex_music.Supplement`

Raises `yandex_music.YandexMusicError`

```
tracks(track_ids: Union[List[Union[str, int]], int, str], with_positions: bool = True, timeout:
    Union[int, float] = None, *args, **kwargs) → List[yandex_music.track.track.Track]
```

```
tracksDownloadInfo(track_id: Union[str, int], get_direct_links: bool = False,
    timeout: Union[int, float] = None, *args, **kwargs) →
    List[yandex_music.download_info.DownloadInfo]
```

Псевдоним для `tracks_download_info`

```
tracksSimilar(track_id: Union[str, int], timeout: Union[int, float] = None, *args, **kwargs)
    → Optional[yandex_music.track.tracks_similar.SimilarTracks]
```

Псевдоним для `tracks_similar`

```
tracks_download_info(track_id: Union[str, int], get_direct_links: bool = False,
    timeout: Union[int, float] = None, *args, **kwargs) →
    List[yandex_music.download_info.DownloadInfo]
```

Получение информации о доступных вариантах загрузки трека.

Параметры

- `track_id` (str | list из str) – Уникальный идентификатор трека или треков.
- `get_direct_links` (bool, optional) – Получить ли при вызове метода прямую ссылку на загрузку.
- `timeout` (int | float, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (dict, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Список объектов класса `yandex_music.DownloadInfo` представляющих варианты загрузки трека, иначе `None`.

Тип результата list из `yandex_music.DownloadInfo`

Raises `yandex_music.YandexMusicError`

```
tracks_similar(track_id: Union[str, int], timeout: Union[int, float] = None, *args, **kwargs)
    → Optional[yandex_music.track.tracks_similar.SimilarTracks]
```

Получение похожих треков.

Параметры

- `track_id` (`str`) – Уникальный идентификатор трека.
- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объект класса `yandex_music.SimilarTracks` представляющий список похожих треков на другой трек.

Тип результата `yandex_music.SimilarTracks`

Raises `yandex_music.YandexMusicError`

```
usersDislikesTracks(user_id: Union[str, int] = None, if_modified_since_revision: int = 0, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.tracks_list.TracksList]
```

Псевдоним для `users_dislikes_tracks`

```
usersDislikesTracksAdd(track_ids: Union[List[Union[str, int]], str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → bool
```

Псевдоним для `users_dislikes_tracks_add`

```
usersDislikesTracksRemove(track_ids: Union[List[Union[str, int]], str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → bool
```

Псевдоним для `users_dislikes_tracks_remove`

```
usersLikesAlbums(user_id: Union[str, int] = None, rich: bool = True, timeout: Union[int, float] = None, *args, **kwargs) → List[yandex_music.like.Like]
```

Псевдоним для `users_likes_albums`

```
usersLikesAlbumsAdd(album_ids: Union[List[Union[str, int]], str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → bool
```

Псевдоним для `users_likes_albums_add`

```
usersLikesAlbumsRemove(album_ids: Union[List[Union[str, int]], str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → bool
```

Псевдоним для `users_likes_albums_remove`

```
usersLikesArtists(user_id: Union[str, int] = None, with_timestamps: bool = True, timeout: Union[int, float] = None, *args, **kwargs) → List[yandex_music.like.Like]
```

Псевдоним для `users_likes_artists`

```
usersLikesArtistsAdd(artist_ids: Union[List[Union[str, int]], str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → bool
```

Псевдоним для `users_likes_artists_add`

```
usersLikesArtistsRemove(artist_ids: Union[List[Union[str, int]], str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → bool
```

Псевдоним для `users_likes_artists_remove`

```
usersLikesPlaylists(user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → List[yandex_music.like.Like]
```

Псевдоним для `users_likes_playlists`

`usersLikesPlaylistsAdd(playlist_ids: Union[List[Union[str, int]], str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → bool`
 Псевдоним для `users_likes_playlists_add`

`usersLikesPlaylistsRemove(playlist_ids: Union[List[Union[str, int]], str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → bool`
 Псевдоним для `users_likes_playlists_remove`

`usersLikesTracks(user_id: Union[str, int] = None, if_modified_since_revision: int = 0, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.tracks_list.TracksList]`
 Псевдоним для `users_likes_tracks`

`usersLikesTracksAdd(track_ids: Union[List[Union[str, int]], str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → bool`
 Псевдоним для `users_likes_tracks_add`

`usersLikesTracksRemove(track_ids: Union[List[Union[str, int]], str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → bool`
 Псевдоним для `users_likes_tracks_remove`

`usersPlaylists(kind: Union[List[Union[str, int]], str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → List[yandex_music.playlist.playlist.Playlist]`
 Псевдоним для `users_playlists`

`usersPlaylistsChange(kind: Union[str, int], diff: str, revision: int = 1, user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.playlist.playlist.Playlist]`
 Псевдоним для `users_playlists_change`

`usersPlaylistsCreate(title: str, visibility: str = 'public', user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.playlist.playlist.Playlist]`
 Псевдоним для `users_playlists_create`

`usersPlaylistsDelete(kind: Union[str, int], user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → bool`
 Псевдоним для `users_playlists_delete`

`usersPlaylistsDeleteTrack(kind: Union[str, int], from_: int, to: int, revision: int = 1, user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.playlist.playlist.Playlist]`
 Псевдоним для `users_playlists_delete_track`

`usersPlaylistsInsertTrack(kind: Union[str, int], track_id: Union[str, int], album_id: Union[str, int], at: int = 0, revision: int = 1, user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.playlist.playlist.Playlist]`
 Псевдоним для `users_playlists_insert_track`

`usersPlaylistsList(user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → List[yandex_music.playlist.playlist.Playlist]`
 Псевдоним для `users_playlists_list`

`usersPlaylistsName(kind: Union[str, int], name: str, user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args, **kwargs) → Optional[yandex_music.playlist.playlist.Playlist]`

Псевдоним для `users_playlists_name`

```
usersPlaylistsVisibility(kind: Union[str, int], visibility: str, user_id: Union[str, int] =
    None, timeout: Union[int, float] = None, *args, **kwargs) →
    Optional[yandex_music.playlist.playlist.Playlist]
```

Псевдоним для `users_playlists_visibility`

```
users_dislikes_tracks(user_id: Union[str, int] = None, if_modified_since_revision: int
    = 0, timeout: Union[int, float] = None, *args, **kwargs) →
    Optional[yandex_music.tracks_list.TracksList]
```

```
users_dislikes_tracks_add(track_ids: Union[List[Union[str, int]], str, int], user_id:
    Union[str, int] = None, timeout: Union[int, float] = None,
    *args, **kwargs) → bool
```

```
users_dislikes_tracks_remove(track_ids: Union[List[Union[str, int]], str, int], user_id:
    Union[str, int] = None, timeout: Union[int, float] = None,
    *args, **kwargs) → bool
```

```
users_likes_albums(user_id: Union[str, int] = None, rich: bool = True, timeout: Union[int,
    float] = None, *args, **kwargs) → List[yandex_music.like.Like]
```

```
users_likes_albums_add(album_ids: Union[List[Union[str, int]], str, int], user_id: Union[str,
    int] = None, timeout: Union[int, float] = None, *args, **kwargs) →
    bool
```

```
users_likes_albums_remove(album_ids: Union[List[Union[str, int]], str, int], user_id:
    Union[str, int] = None, timeout: Union[int, float] = None, *args,
    **kwargs) → bool
```

```
users_likes_artists(user_id: Union[str, int] = None, with_timestamps: bool =
    True, timeout: Union[int, float] = None, *args, **kwargs) →
    List[yandex_music.like.Like]
```

```
users_likes_artists_add(artist_ids: Union[List[Union[str, int]], str, int], user_id: Union[str,
    int] = None, timeout: Union[int, float] = None, *args, **kwargs) →
    bool
```

```
users_likes_artists_remove(artist_ids: Union[List[Union[str, int]], str, int], user_id:
    Union[str, int] = None, timeout: Union[int, float] = None, *args,
    **kwargs) → bool
```

```
users_likes_playlists(user_id: Union[str, int] = None, timeout: Union[int, float] = None,
    *args, **kwargs) → List[yandex_music.like.Like]
```

```
users_likes_playlists_add(playlist_ids: Union[List[Union[str, int]], str, int], user_id:
    Union[str, int] = None, timeout: Union[int, float] = None, *args,
    **kwargs) → bool
```

```
users_likes_playlists_remove(playlist_ids: Union[List[Union[str, int]], str, int], user_id:
    Union[str, int] = None, timeout: Union[int, float] = None,
    *args, **kwargs) → bool
```

```
users_likes_tracks(user_id: Union[str, int] = None, if_modified_since_revision: int
    = 0, timeout: Union[int, float] = None, *args, **kwargs) →
    Optional[yandex_music.tracks_list.TracksList]
```

```
users_likes_tracks_add(track_ids: Union[List[Union[str, int]], str, int], user_id: Union[str,
    int] = None, timeout: Union[int, float] = None, *args, **kwargs) →
    bool
```

```
users_likes_tracks_remove(track_ids: Union[List[Union[str, int]], str, int], user_id:
    Union[str, int] = None, timeout: Union[int, float] = None,
    *args, **kwargs) → bool
```

```
users_playlists(kind: Union[List[Union[str, int]], str, int], user_id: Union[str, int]
                = None, timeout: Union[int, float] = None, *args, **kwargs) →
                List[yandex_music.playlist.playlist.Playlist]
```

Получение плейлиста или списка плейлистов по уникальным идентификаторам.

Параметры

- `kind` (`str | int | list` из `str | int`) – Уникальный идентификатор плейлиста или их список.
- `user_id` (`str | int`, optional) – Уникальный идентификатор пользователя владеющим плейлистом.
- `timeout` (`int | float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Список объектов класса `yandex_music.Playlist` представляющих плейлист, иначе `None`.

Тип результата `list` из `yandex_music.Playlist`

Raises `yandex_music.YandexMusicError`

```
users_playlists_change(kind: Union[str, int], diff: str, revision: int = 1, user_id: Union[str,
                                                int] = None, timeout: Union[int, float] = None, *args, **kwargs) →
                        Optional[yandex_music.playlist.playlist.Playlist]
```

Изменение плейлиста.

Примечание: Для получения отличий есть вспомогательный класс `yandex_music.utils.difference.Difference`.

Так же существуют уже готовые методы-обёртки над операциями.

Параметры

- `kind` (`str | int`) – Уникальный идентификатор плейлиста.
- `revision` (`int`) – TODO.
- `diff` (`str`) – JSON представления отличий старого и нового плейлиста.
- `user_id` (`str | int`, optional) – Уникальный идентификатор пользователя владеющим плейлистом.
- `timeout` (`int | float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Playlist` представляющий изменённый плейлист, иначе `None`.

Тип результата `yandex_music.Playlist`

Raises `yandex_music.YandexMusicError`

```
users_playlists_create(title: str, visibility: str = 'public', user_id: Union[str, int] =
    None, timeout: Union[int, float] = None, *args, **kwargs) →
    Optional[yandex_music.playlist.playlist.Playlist]
```

Создание плейлиста.

Параметры

- `title` (`str`) – Название.
- `visibility` (`str`, optional) – Модификатор доступа.
- `user_id` (`str` | `int`, optional) – Уникальный идентификатор пользователя владеющим плейлистом.
- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Playlist` представляющий созданный плейлист, иначе `None`.

Тип результата `yandex_music.Playlist`

Raises `yandex_music.YandexMusicError`

```
users_playlists_delete(kind: Union[str, int], user_id: Union[str, int] = None, timeout:
    Union[int, float] = None, *args, **kwargs) → bool
```

Удаление плейлиста.

Параметры

- `kind` (`str` | `int`) – Уникальный идентификатор плейлиста.
- `user_id` (`str` | `int`, optional) – Уникальный идентификатор пользователя владеющим плейлистом.
- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат `True` при успешном выполнении запроса, иначе `False`.

Тип результата `bool`

Raises `yandex_music.YandexMusicError`

```
users_playlists_delete_track(kind: Union[str, int], from_: int, to: int, revision:
    int = 1, user_id: Union[str, int] = None, timeout:
    Union[int, float] = None, *args, **kwargs) →
    Optional[yandex_music.playlist.playlist.Playlist]
```

Удаление треков из плейлиста.

Примечание: Для удаление необходимо указать границы с какого по какой элемент (трек) удалить.

Параметры

- `kind` (`str` | `int`) – Уникальный идентификатор плейлиста.

- `from` (`int`) – С какого индекса.
- `to` (`int`) – По какой индекс.
- `revision` (`int`) – TODO.
- `user_id` (`str` | `int`, optional) – Уникальный идентификатор пользователя владеющим плейлистом.
- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Playlist` представляющий изменённый плейлист, иначе `None`.

Тип результата `yandex_music.Playlist`

Raises `yandex_music.YandexMusicError`

```
users_playlists_insert_track(kind: Union[str, int], track_id: Union[str, int],
                             album_id: Union[str, int], at: int = 0, revision:
                             int = 1, user_id: Union[str, int] = None, timeout:
                             Union[int, float] = None, *args, **kwargs) →
                             Optional[yandex_music.playlist.playlist.Playlist]
```

Добавление трека в плейлист.

Примечание: Трек можно вставить с любое место плейлиста задав индекс вставки (аргумент `at`).

Параметры

- `kind` (`str` | `int`) – Уникальный идентификатор плейлиста.
- `track_id` (`str` | `int`) – Уникальный идентификатор трека.
- `album_id` (`str` | `int`) – Уникальный идентификатор альбома.
- `at` (`int`) – Индекс для вставки.
- `revision` (`int`) – TODO.
- `user_id` (`str` | `int`, optional) – Уникальный идентификатор пользователя владеющим плейлистом.
- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Playlist` представляющий изменённый плейлист, иначе `None`.

Тип результата `yandex_music.Playlist`

Raises `yandex_music.YandexMusicError`

```
users_playlists_list(user_id: Union[str, int] = None, timeout: Union[int, float] = None,
                     *args, **kwargs) → List[yandex_music.playlist.playlist.Playlist]
```

```
users_playlists_name(kind: Union[str, int], name: str, user_id: Union[str, int] =
                    None, timeout: Union[int, float] = None, *args, **kwargs) →
                    Optional[yandex_music.playlist.playlist.Playlist]
```

Изменение названия плейлиста.

Параметры

- `kind` (`str` | `int`) – Уникальный идентификатор плейлиста.
- `name` (`str`) – Новое название.
- `user_id` (`str` | `int`, optional) – Уникальный идентификатор пользователя владеющим плейлистом.
- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Playlist` представляющий изменённый плейлист, иначе `None`.

Тип результата `yandex_music.Playlist`

Raises `yandex_music.YandexMusicError`

```
users_playlists_visibility(kind: Union[str, int], visibility: str, user_id: Union[str, int] =
                          None, timeout: Union[int, float] = None, *args, **kwargs) →
                          Optional[yandex_music.playlist.playlist.Playlist]
```

Изменение видимости плейлиста.

Примечание: Видимость (`visibility`) может быть задана только одним из двух значений: `private`, `public`.

Параметры

- `kind` (`str` | `int`) – Уникальный идентификатор плейлиста.
- `visibility` (`str`) – Новое название.
- `user_id` (`str` | `int`, optional) – Уникальный идентификатор пользователя владеющим плейлистом.
- `timeout` (`int` | `float`, optional) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict`, optional) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.Playlist` представляющий изменённый плейлист, иначе `None`.

Тип результата `yandex_music.Playlist`

Raises `yandex_music.YandexMusicError`

```
users_settings(user_id: Union[str, int] = None, timeout: Union[int, float] = None, *args,
               **kwargs) → Optional[yandex_music.account.user_settings.UserSettings]
```

Получение настроек пользователя.

Примечание: Для получения настроек пользователя нужно быть авторизованным или владеть *user_id*.

Параметры

- `user_id` (`str | int, optional`) – Уникальный идентификатор пользователя чьи настройки хотим получить.
- `timeout` (`int | float, optional`) – Если это значение указано, используется как время ожидания ответа от сервера вместо указанного при создании пула.
- `**kwargs` (`dict, optional`) – Произвольные аргументы (будут переданы в запрос).

Результат Объекта класса `yandex_music.UserSettings` представляющий настройки пользователя, иначе `None`.

Тип результата `yandex_music.UserSettings`

Raises `yandex_music.YandexMusicError`

2.2 `yandex_music.YandexMusicObject`

```
class yandex_music.YandexMusicObject
```

Базовые классы: `object`

```
classmethod de_json(data: dict, client: Optional[Client]) → Optional[dict]
```

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.YandexMusicObject`.

Тип результата `yandex_music.YandexMusicObject`

```
to_dict() → dict
```

```
to_json() → str
```

2.3 `yandex_music.Cover`

```
class yandex_music.Cover(type_: Optional[str] = None, uri: Optional[str] = None, items_uri:
    Optional[str] = None, dir_: Optional[str] = None, version: Optional[str]
    = None, custom: Optional[bool] = None, prefix: Optional[str] = None,
    error: Optional[str] = None, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий обложку.

```
type
```

Тип обложки.

Type str

`uri`
Ссылка на изображение.

Type str

`items_uri`
Список ссылок на изображения.

Type str

`dir`
Директория хранения изображения на сервере.

Type str

`version`
Версия.

Type str

`custom`
Является ли обложка пользовательской.

Type bool

`prefix`
Уникальный идентификатор.

Type str

`error`
Сообщение об ошибке.

Type str

`client`
Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Type `yandex_music.Client`

Параметры

- `type` (`str`, optional) – Тип обложки.
- `uri` (`str`, optional) – Ссылка на изображение.
- `items_uri` (`str`, optional) – Список ссылок на изображения.
- `dir` (`str`, optional) – Директория хранения изображения на сервере.
- `version` (`str`, optional) – Версия.
- `custom` (`bool`, optional) – Является ли обложка пользовательской.
- `prefix` (`str`, optional) – Уникальный идентификатор.
- `error` (`str`, optional) – Сообщение об ошибке.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[Cover]`
Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Cover`.

Тип результата `yandex_music.Cover`

```
classmethod de_list(data: dict, client: Client) → List[Cover]
```

Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.Cover`.

Тип результата `list` из `yandex_music.Cover`

```
download(filename: str, index: int = 0, size: str = '200x200') → None
```

Загрузка обложки.

Параметры

- `filename` (`str`) – Путь для сохранения файла с названием и расширением.
- `index` (`int`, optional) – Индекс элемента в списке ссылок на обложки если нет `self.uri`.
- `size` (`str`, optional) – Размер изображения.

2.4 yandex_music.DownloadInfo

```
class yandex_music.DownloadInfo(codec: str, bitrate_in_kbps: int, gain: bool, preview: bool,
                               download_info_url: str, client: Optional[Client] = None,
                               **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий информацию о вариантах загрузки трека.

`codec`

Кодек аудиофайла.

Тип `str`

`bitrate_in_kbps`

Битрейт аудиофайла в кбит/с.

Тип `int`

`gain`

Усиление TODO.

Тип `bool`

`preview`

Предварительный просмотр TODO.

Type `bool`

`download_info_url`

Ссылка на XML документ содержащий данные для загрузки трека.

Type `str`

`direct_link`

Прямая ссылка на загрузку. Доступна после получения ссылки.

Type `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Type `yandex_music.Client`

Параметры

- `codec (str)` – Кодек аудиофайла.
- `bitrate_in_kbps (int)` – Битрейт аудиофайла в кбит/с.
- `gain (bool)` – Усиление TODO.
- `preview (bool)` – Предварительный просмотр TODO.
- `download_info_url (str)` – Ссылка на XML документ содержащий данные для загрузки трека.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[DownloadInfo]`

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.DownloadInfo`.

Тип результата `yandex_music.DownloadInfo`

`classmethod de_list(data: dict, client: Client, get_direct_links: bool = False) → List[DownloadInfo]`

Десериализация списка объектов.

Параметры

- `data (list)` – Список словарей с полями и значениями десериализуемого объекта.
- `get_direct_links (bool)` – Получать ли сразу прямые ссылки на загрузку.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.DownloadInfo*.

Тип результата *list* из *yandex_music.DownloadInfo*

`download(filename: str) → None`

Загрузка трека.

Параметры `filename (str)` – Путь и(или) название файла вместе с расширением.

`getDirectLink() → str`

Псевдоним для *get_direct_link*

`get_direct_link() → str`

Получение прямой ссылки на загрузку из XML ответа.

Метод доступен только одну минуту с момента получения информации о загрузке, иначе 410 ошибка!

Результат Прямая ссылка на загрузку трека.

Тип результата *str*

2.5 *yandex_music.Experiments*

`class yandex_music.Experiments(client: Optional[Client] = None, **kwargs)`

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий какие-то свистелки-перделки, флажки, режимы экспериментальных функций.

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

`название_эксперимента`

Содержит режим или состояние, или функцию, или ещё что угодно.

Тип *str, optional*

Параметры

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Собственно тут и передаются все эти свистелки.

`classmethod de_json(data: dict, client: Client) → Optional[Experiments]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Experiments*.

Тип результата *yandex_music.Experiments*

2.6 yandex_music.Icon

```
class yandex_music.Icon(background_color: str, image_url: str, client: Optional[Client] = None,
                       **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий иконку.

`background_color`

Цвет заднего фона в HEX.

Тип `str`

`image_url`

Ссылка на изображение.

Тип `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `background_color (str)` – Цвет заднего фона в HEX.
- `image_url (str)` – Ссылка на изображение.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Icon]
```

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Icon`.

Тип результата `yandex_music.Icon`

```
download(filename: str, size: str = '200x200') → None
```

Загрузка иконки.

Параметры

- `filename (str)` – Путь для сохранения файла с названием и расширением.
- `size (str, optional)` – Размер иконки.

2.7 yandex_music.InvocationInfo

```
class yandex_music.InvocationInfo(hostname: str, req_id: str, exec_duration_millis:
    Optional[int] = None, client: Optional[Client] = None,
    **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий информацию о запросе.

`hostname`

Имя удалённого сервера.

Тип `str`

`req_id`

Номер запроса.

Тип `str`

`exec_duration_millis`

Время выполнения в миллисекундах.

Тип `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `hostname` (`str`) – Имя удалённого сервера.
- `req_id` (`str`) – Номер запроса.
- `exec_duration_millis` (`str`, optional) – Время выполнения в миллисекундах.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[InvocationInfo]
```

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.InvocationInfo`.

Тип результата `yandex_music.InvocationInfo`

2.8 yandex_music.Pager

```
class yandex_music.Pager(total: int, page: int, per_page: int, client: Optional[Client] = None,
    **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий пагинатор.

`total`

Всего треков.

Тип `int`

`page`

Номер страницы.

Тип `int`

`per_page`

Количество треков на странице.

Тип `int`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `total (int)` – Всего треков.
- `page (int)` – Номер страницы.
- `per_page (int)` – Количество треков на странице.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Pager]`

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Pager`.

Тип результата `yandex_music.Pager`

2.9 `yandex_music.PermissionAlerts`

class `yandex_music.PermissionAlerts(alerts: List[str], client: Optional[Client] = None, **kwargs)`

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий оповещения.

`alerts`

Список оповещений.

Тип `list` из `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `alerts` (`list` из `str`) – Список оповещений.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[PermissionAlerts]`

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.PermissionAlerts`.

Тип результата `yandex_music.PermissionAlerts`

2.10 `yandex_music.PromoCodeStatus`

```
class yandex_music.PromoCodeStatus(status: str, status_desc: str, account_status:
    Optional[Status], client: Optional[Client] = None,
    **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий статус активации промо-кода.

`status`

Статус операции.

Типе `str`

`status_desc`

Описание статуса.

Типе `str`

`account_status`

Объект класса `yandex_music.Status` представляющий подробную информацию об аккаунте пользователя.

Типе `yandex_music.Status`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `status` (`str`) – Статус операции.

- `status_desc` (`str`) – Описание статуса.
- `account_status` (`yandex_music.Status`) – Объект класса `yandex_music.Status` представляющий подробную информацию об аккаунте пользователя.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[PromoCodeStatus]`
 Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.PromoCodeStatus`.

Тип результата `yandex_music.PromoCodeStatus`

2.11 `yandex_music.Settings`

```
class yandex_music.Settings(in_app_products: List[Product], native_products: List[Product],
                           web_payment_url: str, promo_codes_enabled: bool,
                           web_payment_month_product_price: Optional[Price] = None,
                           client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий предложения по покупке.

`in_app_products`

Список объектов класса `yandex_music.Product` представляющий продаваемые продукты внутри приложения.

Тип list из `yandex_music.Product`

`native_products`

Список объектов класса `yandex_music.Product` представляющий продаваемые продукты всплывающими окнами.

Тип list из `yandex_music.Product`

`web_payment_url`

Ссылка для осуществления платежа.

Тип str

`web_payment_month_product_price`

Объект класса `yandex_music.Price` представляющий цену продукта за месяц.

Тип `yandex_music.Price`

`promo_codes_enabled`

Доступно ли использование промо-кодов.

Тип bool

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `in_app_products` (list из `yandex_music.Product`) – Список объектов класса `yandex_music.Product` представляющий продаваемые продукты внутри приложения.
- `native_products` (list) – Список объектов класса `yandex_music.Product` представляющий продаваемые продукты всплывающими окнами.
- `web_payment_url` (str) – Ссылка для осуществления платежа.
- `web_payment_month_product_price` (`yandex_music.Price`, optional) – Объект класса `yandex_music.Price` представляющий цену продукта за месяц.
- `promo_codes_enabled` (bool) – Доступно ли использование промо-кодов.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Settings]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Settings`.

Тип результата `yandex_music.Settings`

2.12 `yandex_music.TrackShort`

```
class yandex_music.TrackShort(id_: Union[str, int], timestamp: str, album_id: Optional[str] = None, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий укороченную версию трека с неполными данными.

`id`

Уникальный идентификатор трека.

Типе `str`

`timestamp`

Дата TODO.

Типе `str`

`album_id`

Уникальный идентификатор альбома.

Типе `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `id (str)` – Уникальный идентификатор трека.
- `timestamp (str)` – Дата TODO.
- `album_id (str, optional)` – Уникальный идентификатор альбома.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[TrackShort]`

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.TrackShort`.

Тип результата `yandex_music.TrackShort`

`classmethod de_list(data: dict, client: Client) → List[TrackShort]`

Десериализация списка объектов.

Параметры

- `data (list)` – Список словарей с полями и значениями десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.TrackShort`.

Тип результата `list` из `yandex_music.TrackShort`

`track`

Объект класса `yandex_music.Track` представляющий полную версию трека.

Типе `yandex_music.Track`

`trackId`

Псевдоним для `track_id`

`track_id`

Уникальный идентификатор трека состоящий из его номера и номера альбома или просто из номера.

Типе `str`

2.13 `yandex_music.TracksList`

```
class yandex_music.TracksList(
    uid: int, revision: int, tracks: List[TrackShort], client:
    Optional[Client] = None, **kwargs)

```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий список треков.

`uid`

Уникальный идентификатор пользователя.

Тип `int`

`revision`

Актуальность данных TODO.

Тип `int`

`tracks`

Список треков.

Тип `list` из `yandex_music.TrackShort`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `uid` (`int`) – Уникальный идентификатор пользователя.
- `revision` (`int`) – Актуальность данных TODO.
- `tracks` (`list` из `yandex_music.TrackShort`) – Список треков.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[TracksList]

```

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.TracksList`.

Тип результата `yandex_music.TracksList`

`tracksIds`

Псевдоним для `tracks_ids`

`tracks_ids`

Список уникальных идентификаторов треков.

Тип `list` из `str`

2.14 yandex_music.Video

```
class yandex_music.Video(title: str, cover: Optional[str] = None, embed_url: Optional[str]
    = None, provider: Optional[str] = None, provider_video_id:
    Union[str, int, None] = None, youtube_url: Optional[str] =
    None, thumbnail_url: Optional[str] = None, duration=None,
    text=None, html_auto_play_video_player=None, regions=None,
    client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Video]
```

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Video`.

Тип результата `yandex_music.Video`

```
classmethod de_list(data: dict, client: Client) → List[Video]
```

Десериализация списка объектов.

Параметры

- `data` (list) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.Video`.

Тип результата list из `yandex_music.Video`

2.15 Пакет `yandex_music.account`

2.15.1 `yandex_music.Status`

```
class yandex_music.Status(account: Optional[Account], permissions: Optional[Permissions],
    subscription: Optional[Subscription] = None, cache_limit: Optional[int] = None,
    subeditor: Optional[bool] = None, subeditor_level: Optional[int] = None, plus: Optional[Plus] = None,
    default_email: Optional[str] = None, skips_per_hour: Optional[int] = None,
    station_exists: Optional[bool] = None, premium_region: Optional[int] = None, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий подробную информацию об аккаунте пользователя.

`account`

Объект класса `yandex_music.Account` представляющий основную информацию об аккаунте.

Type `yandex_music.Account`

`permissions`

Объект класса `yandex_music.Permissions` представляющий информацию о правах пользователя.

Type `yandex_music.Permissions`

`subscription`

Объект класса `yandex_music.Subscription` представляющий информацию о подписках

Type `yandex_music.Subscription`

`cache_limit`

Максимальное количество загруженных треков.

Type `int`

`subeditor`

Наличие статуса модератора проверки корректности информации.

Type `bool`

`subeditor_level`

Уровень статуса модератора.

Type `int`

`plus`

Объект класса `yandex_music.Plus` представляющий информацию о Plus подписке.

Type `yandex_music.Plus`

`default_email`

Основной e-mail адрес аккаунта.

Type `str`

`skips_per_hour`

Количество переключение треков на радио в час.

Type `int`

`station_exists`

Наличие станции TODO.

Типе bool

premium_region

Регион TODO.

Типе int

client

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Типе *yandex_music.Client*

Параметры

- **account** (*yandex_music.Account*) – Объект класса *yandex_music.Account* представляющий основную информацию об аккаунте.
- **permissions** (*yandex_music.Permissions*) – Объект класса *yandex_music.Permissions* представляющий информацию о правах пользователя.
- **subscription** (*yandex_music.Subscription*) – Объект класса *yandex_music.Subscription* представляющий информацию о подписках
- **cache_limit** (int, optional) – Максимальное количество загруженных треков.
- **subeditor** (bool, optional) – Наличие статуса модератора проверки корректности информации.
- **subeditor_level** (int, optional) – Уровень статуса модератора.
- **plus** (*yandex_music.Plus*, optional) – Объект класса *yandex_music.Plus* представляющий информацию о Plus подписке.
- **default_email** (str, optional) – Основной e-mail адрес аккаунта.
- **skips_per_hour** (int, optional) – Количество переключение треков на радио в час.
- **station_exists** (bool, optional) – Наличие станции TODO.
- **premium_region** (int, optional) – Регион TODO.
- **client** (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- ****kwargs** – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Status]`

Десериализация объекта.

Параметры

- **data** (dict) – Поля и значения десериализуемого объекта.
- **client** (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Status*.

Тип результата *yandex_music.Status*

2.15.2 yandex_music.Account

```
class yandex_music.Account(now: str, service_available: bool, region: Optional[int] = None, uid:
    Optional[int] = None, login: Optional[str] = None, full_name:
    Optional[str] = None, second_name: Optional[str] = None,
    first_name: Optional[str] = None, display_name: Optional[str] =
    None, hosted_user: Optional[bool] = None, birthday: Optional[str] =
    None, passport_phones: List[PassportPhone] = None, registered_at:
    Optional[str] = None, has_info_for_app_metrics: bool = False,
    client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий основную информацию об аккаунте пользователя.

`now`

Текущая дата и время.

Type `str`

`uid`

Уникальный идентификатор.

Type `int`

`login`

Виртуальное имя (обычно e-mail).

Type `str`

`full_name`

Полное имя (имя и фамилия).

Type `str`

`second_name`

Фамилия.

Type `str`

`first_name`

Имя.

Type `str`

`display_name`

Отображаемое имя.

Type `str`

`birthday`

Дата рождения.

Type `str`

`service_available`

Доступен ли сервис.

Type `bool`

`hosted_user`

Является ли пользователем чьим-то другим.

Type `bool`

`region`

Регион.

Type `int`

`passport_phones`

Список объектов класса `yandex_music.PassportPhone` представляющие мобильные номера.

Type `list` из `yandex_music.PassportPhone`

`registered_at`

Дата создания аккаунта.

Type `str`

`has_info_for_app_metrika`

Наличие информации для App Metrika.

Type `bool`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Type `yandex_music.Client`

Параметры

- `now` (`str`) – Текущая дата и время.
- `uid` (`int`, optional) – Уникальный идентификатор.
- `login` (`str`, optional) – Виртуальное имя (обычно e-mail).
- `full_name` (`str`, optional) – Полное имя (имя и фамилия).
- `second_name` (`str`, optional) – Фамилия.
- `first_name` (`str`, optional) – Имя.
- `display_name` (`str`, optional) – Отображаемое имя.
- `service_available` (`bool`) – Доступен ли сервис.
- `hosted_user` (`bool`, optional) – Является ли пользователем чьим-то другим.
- `birthday` (`str`, optional) – Дата рождения.
- `region` (`int`, optional) – Регион.
- `passport_phones` (`list` из `yandex_music.PassportPhone`) – Список объектов класса `yandex_music.PassportPhone` представляющие мобильные номера.
- `registered_at` (`str`, optional) – Дата создания учётной записи.
- `has_info_for_app_metrika` (`bool`, optional) – Наличие информации для App Metrika.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[Account]`

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Account`.

Тип результата `yandex_music.Account`

`downloadAvatar(filename: str, format_: str = 'normal') → None`

Псевдоним для `download_avatar`

`download_avatar(filename: str, format_: str = 'normal') → None`

Загрузка изображения пользователя.

Параметры

- `filename` (`str`) – Путь для сохранения файла с названием и расширением.
- `format` (`str`) – Формат желаемого изображения (`normal`, `orig`, `small`, `big`).

2.15.3 `yandex_music.PassportPhone`

`class yandex_music.PassportPhone(phone: str, client: Optional[Client] = None, **kwargs)`

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий номер телефона пользователя.

`phone`

Номер телефона.

Тип `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `phone` (`str`) – Номер телефона.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[PassportPhone]`

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.PassportPhone`.

Тип результата `yandex_music.PassportPhone`

`classmethod de_list(data: dict, client: Client) → List[PassportPhone]`

Десериализация списка объектов.

Параметры

- `data (list)` – Список словарей с полями и значениями десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.PassportPhone`.

Тип результата `list` из `yandex_music.PassportPhone`

2.15.4 `yandex_music.Plus`

```
class yandex_music.Plus(has_plus: bool, is_tutorial_completed: bool, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий *Plus* подписку.

`has_plus`

Наличие.

Тип `bool`

`is_tutorial_completed`

Закончено ли руководство.

Тип `bool`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `has_plus (bool)` – Наличие.
- `is_tutorial_completed (bool)` – Закончено ли руководство.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Plus]
```

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Plus`.

Тип результата `yandex_music.Plus`

2.15.5 yandex_music.Permissions

```
class yandex_music.Permissions(until: str, values: List[str], default: List[str], client:
    Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий информацию о правах пользователя, их изначальных значениях и даты окончания.

`until`

Дата окончания прав.

Type `str`

`values`

Список прав.

Type `list` из `str`

`default`

Список изначальных прав.

Type `list` из `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Type `yandex_music.Client`

Параметры

- `until` (`str`) – Дата окончания прав.
- `values` (`list` из `str`) – Список прав.
- `default` (`list` из `str`) – Список изначальных прав.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Permissions]
```

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Permissions`.

Тип результата `yandex_music.Permissions`

2.15.6 yandex_music.Subscription

```
class yandex_music.Subscription(auto_renewable: List[AutoRenewable] = None, can_start_trial:
    Optional[bool] = None, mcdonalds: Optional[bool] = None,
    end: Optional[str] = None, client: Optional[Client] = None,
    **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий информацию о подписках пользователя.

`auto_renewable`

Список объектов класса `yandex_music.AutoRenewable` представляющих автопродление.

Type list из `yandex_music.AutoRenewable`

`can_start_trial`

Есть ли возможность начать пробный период.

Type bool

`mcdonalds`

mcdonalds TODO.

Type bool

`end`

Дата окончания.

Type str

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Type `yandex_music.Client`

Параметры

- `auto_renewable` (list из `yandex_music.AutoRenewable`, optional) – Список объектов класса `yandex_music.AutoRenewable` представляющих автопродление.
- `can_start_trial` (bool, optional) – Есть ли возможность начать пробный период.
- `mcdonalds` (bool, optional) – mcdonalds TODO.
- `end` (str, optional) – Дата окончания.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[Subscription]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Subscription`.

Тип результата `yandex_music.Subscription`

2.15.7 yandex_music.AutoRenewable

```
class yandex_music.AutoRenewable(expires: str, vendor: str, vendor_help_url: str, product:
    Optional[Product], finished: bool, product_id: Optional[str] =
    None, order_id: Optional[int] = None, client: Optional[Client]
    = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий автопродление подписки.

`expires`

Дата истечения подписки.

Type `str`

`vendor`

Продавец.

Type `str`

`vendor_help_url`

Ссылка на страницу помощи продавца.

Type `str`

`product_id`

Уникальный идентификатор продукта.

Type `str`

`product`

Объект класса `yandex_music.Product` представляющий продукт.

Type `yandex_music.Product`

`order_id`

Уникальный идентификатор заказа.

Type `int`

`finished`

Завершенность автопродления.

Type `bool`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Type `yandex_music.Client`

Параметры

- `expires (str)` – Дата истечения подписки.
- `vendor (str)` – Продавец.
- `vendor_help_url (str)` – Ссылка на страницу помощи продавца.
- `product_id (str)` – Уникальный идентификатор продукта.
- `finished (bool)` – Завершенность автопродления.
- `product (yandex_music.Product, optional)` – Объект класса `yandex_music.Product` представляющий продукт.
- `order_id (int)` – Уникальный идентификатор заказа.

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[AutoRenewable]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.AutoRenewable*.

Тип результата *yandex_music.AutoRenewable*

classmethod `de_list(data: dict, client: Client) → List[AutoRenewable]`

Десериализация списка объектов.

Параметры

- `data` (list) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.AutoRenewable*.

Тип результата list из *yandex_music.AutoRenewable*

2.15.8 yandex_music.Product

```
class yandex_music.Product(product_id: str, type_: str, common_period_duration: str, duration:
    int, trial_duration: int, price: Optional[Price], feature: str, debug:
    bool, features: List[str] = None, description: Optional[str] = None,
    available: Optional[bool] = None, trial_available: Optional[bool] =
    None, vendor_trial_available: Optional[bool] = None, button_text:
    Optional[str] = None, button_additional_text: Optional[str] = None,
    payment_method_types: List[str] = None, client: Optional[Client] =
    None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий продаваемый продукт.

`product_id`

Уникальный идентификатор.

Тип str

`type`

Тип продаваемого.

Тип str

`common_period_duration`

Длительность общего периода.

Тип str

duration
Длительность.
Type int

trial_duration
Длительность испытательного срока.
Type int

price
Объект класса *yandex_music.Price* представляющий цену.
Type *yandex_music.Price*

feature
Предоставляемая возможность.
Type str

debug
Отладочный продукт.
Type bool

features
Список предоставляемых возможностей.
Type list из str

description
Описание.
Type str

available
Доступна ли покупка.
Type bool

trial_available
Доступен ли испытательный срок.
Type bool

vendor_trial_available
Доступен испытательный срок продавца TODO.
Type bool

button_text
Текст кнопки.
Type str

button_additional_text
Дополнительный текст кнопки.
Type str

payment_method_types
Способы оплаты.
Type list из str

client
Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Типе *yandex_music.Client*

Параметры

- `product_id` (`str`) – Уникальный идентификатор.
- `type` (`str`) – Тип продаваемого.
- `common_period_duration` (`str`) – Длительность общего периода.
- `duration` (`int`) – Длительность.
- `trial_duration` (`int`) – Длительность испытательного срока.
- `price` (*yandex_music.Price*) – Объект класса *yandex_music.Price* представляющий цену.
- `feature` (`str`) – Предоставляемая возможность.
- `debug` (`bool`) – Отладочный продукт.
- `features` (`list` из `str`, `optional`) – Список предоставляемых возможностей.
- `description` (`str`, `optional`) – Описание.
- `available` (`bool`, `optional`) – Доступна ли покупка.
- `trial_available` (`bool`, `optional`) – Доступен ли испытательный срок.
- `vendor_trial_available` (`bool`, `optional`) – Доступен испытательный срок продавца TODO.
- `button_text` (`str`, `optional`) – Текст кнопки.
- `button_additional_text` (`str`, `optional`) – Дополнительный текст кнопки.
- `payment_method_types` (`list` из `str`, `optional`) – Способы оплаты.
- `client` (*yandex_music.Client*, `optional`) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[Product]`
Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Product*.

Тип результата *yandex_music.Product*

`classmethod de_list(data: dict, client: Client) → List[Product]`
Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Product*.

Тип результата *list* из *yandex_music.Product*

2.15.9 *yandex_music.Price*

```
class yandex_music.Price(amount: int, currency: str, client: Optional[Client] = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий цену.

amount

Количество единиц.

Тип *int*

currency

Валюта.

Тип *str*

client

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- **amount** (*int*) – Количество единиц.
- **currency** (*str*) – Валюта.
- **client** (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- ****kwargs** – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Price]
```

Десериализация объекта.

Параметры

- **data** (*dict*) – Поля и значения десериализуемого объекта.
- **client** (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Price*.

Тип результата *yandex_music.Price*

2.15.10 yandex_music.UserSettings

```
class yandex_music.UserSettings(
    uid: int, last_fm_scrobbling_enabled: bool,
    shuffle_enabled: bool, volume_percents: int,
    modified: str, facebook_scrobbling_enabled: bool,
    add_new_track_on_playlist_top: bool, user_music_visibility:
    str, user_social_visibility: str, rbt_disabled: bool, theme: str,
    promos_disabled: bool, auto_play_radio: bool, ads_disabled:
    Optional[bool] = None, disk_enabled: Optional[bool] = None,
    show_disk_tracks_in_library: Optional[bool] = None, client:
    Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий настройки пользователя.

Примечание: Доступные значения для поля `theme`: `white`, `black`.

Доступные значения для полей `user_music_visibility` и `user_social_visibility`: `private`, `public`.

Notes

`promos_disabled`, `ads_disabled`, `rbt_disabled` устарели и не работают.

`last_fm_scrobbling_enabled`, `facebook_scrobbling_enabled` выглядят устаревшими.

`uid`

Уникальный идентификатор пользователя.

Type int

`last_fm_scrobbling_enabled`

Скробблинг lastfm.

Type bool

`shuffle_enabled`

Переключать треки в случайном порядке.

Type bool

`volume_percents`

Громкость звука в процентах.

Type int

`modified`

Дата изменения настроек.

Type str

`facebook_scrobbling_enabled`

Скробблинг facebook.

Type bool

`add_new_track_on_playlist_top`

Добавлять новые треки в начало плейлиста.

Type bool

`user_music_visibility`
Публичный доступ к моей фонотеке.
Type `str`

`user_social_visibility`
Показывать соцсети на странице.
Type `str`

`rbt_disabled`
TODO (неиспользуемая фича).
Type `bool`

`theme`
Тема оформления.
Type `str`

`promos_disabled`
Не показывать рекламируемый контент).
Type `bool`

`auto_play_radio`
Бесконечный поток музыки.
Type `bool`

`ads_disabled`
Не показывать рекламу.
Type `bool`

`disk_enabled`
TODO.
Type `bool`

`show_disk_tracks_in_library`
Показывать локальные треки в библиотеке.
Type `bool`

`client`
Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
Type `yandex_music.Client`

Параметры

- `uid (int)` – Уникальный идентификатор пользователя.
- `last_fm_scrobbling_enabled (bool)` – Скробблинг lastfm.
- `shuffle_enabled (bool)` – Переключать треки в случайном порядке.
- `volume_percents (int)` – Громкость звука в процентах.
- `modified (str)` – Дата изменения настроек.
- `facebook_scrobbling_enabled (bool)` – Скробблинг facebook.
- `add_new_track_on_playlist_top (bool)` – Добавлять новые треки в начало плейлиста.

- `user_music_visibility` (`str`) – Публичный доступ к моей фонотеке.
- `user_social_visibility` (`str`) – Показывать соцсети на странице.
- `rbt_disabled` (`bool`) – TODO (неиспользуемая фича).
- `theme` (`str`) – Тема оформления.
- `promos_disabled` (`bool`) – Не показывать рекламируемый контент).
- `auto_play_radio` (`bool`) – Бесконечный поток музыки.
- `ads_disabled` (`bool`, `optional`) – Не показывать рекламу.
- `disk_enabled` (`bool`, `optional`) – TODO.
- `show_disk_tracks_in_library` (`bool`, `optional`) – Показывать локальные треки в библиотеке.
- `client` (`yandex_music.Client`, `optional`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[UserSettings]`
 Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.UserSettings`.

Тип результата `yandex_music.UserSettings`

2.16 Пакет `yandex_music.album`

2.16.1 `yandex_music.Album`

```
class yandex_music.Album(id_: int, error: Optional[str] = None, title: Optional[str] =
    None, track_count: Optional[int] = None, artists: List[Artist]
    = None, labels: List[Label] = None, available: Optional[bool] =
    None, available_for_premium_users: Optional[bool] = None,
    version: Optional[str] = None, cover_uri: Optional[str] = None,
    content_warning: Optional[str] = None, original_release_year=None,
    genre: Optional[str] = None, og_image: Optional[str] = None,
    buy: Optional[list] = None, recent: Optional[bool] = None,
    very_important: Optional[bool] = None, available_for_mobile:
    Optional[bool] = None, available_partially: Optional[bool] = None,
    bests: Optional[List[int]] = None, prerolls: Optional[list] = None,
    volumes: Optional[List[List[Track]]] = None, year: Optional[int] =
    None, release_date: Optional[str] = None, type_: Optional[str] = None,
    track_position: Optional[TrackPosition] = None, regions=None, client:
    Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий альбом.

Примечание: Известные типы альбома: *single* - сингл, *compilation* - сборник.

Известные предупреждения о содержимом: *explicit* - ненормативная лексика.

Известные ошибки: *not-found* - альбом с таким ID не существует.

`id`

Идентификатор альбома.

Type int

`error`

Ошибка получения альбома.

Type str

`title`

Название альбома.

Type str

`track_count`

Количество треков.

Type int

`artists`

Список объектов класса *yandex_music.Artist* представляющие артистов.

Type list из *yandex_music.Artist*

`labels`

Список объектов класса *yandex_music.Label* представляющие лейблы.

Type list из *yandex_music.Label*

`available`

Доступен ли альбом.

Type bool

`available_for_premium_users`

Доступен ли альбом для пользователей с подпиской.

Type bool

`version`

Дополнительная информация об альбоме.

Type str

`cover_uri`

Ссылка на обложку.

Type str

`content_warning`

Предупреждение о содержимом альбома.

Type str

`genre`

Жанр музыки.

Type str

`og_image`
Ссылка на превью Open Graph.

Type str

`recent`
Является ли альбом новым.

Type bool

`very_important`
Популярен ли альбом у слушателей.

Type bool

`available_for_mobile`
Доступен ли альбом из приложения для телефона.

Type bool

`available_partially`
Доступен ли альбом частично для пользователей без подписки.

Type bool

`bests`
ID лучших треков альбома.

Type list из int

`volumes`
Треки альбома, разделенные по дискам.

Type list из list из *Track*

`year`
Год релиза.

Type int

`release_date`
Дата релиза в формате ISO 8601.

Type str

`type`
Тип альбома.

Type str

`track_position`
Объект класса *yandex_music.TrackPosition* представляющий позицию трека. Возвращается при получении альбома в составе трека.

Type *yandex_music.TrackPosition*

`client`
Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Type *yandex_music.Client*

Параметры

- `id (int)` – Идентификатор альбома.

- `error` (`str`, optional) – Ошибка получения альбома.
- `title` (`str`, optional) – Название альбома.
- `track_count` (`int`, optional) – Количество треков.
- `artists` (`list` из `yandex_music.Artist`, optional) – Список объектов класса `yandex_music.Artist` представляющие артистов.
- `labels` (`list` из `yandex_music.Label`, optional) – Список объектов класса `yandex_music.Label` представляющие лейблы.
- `available` (`bool`, optional) – Доступен ли альбом.
- `available_for_premium_users` (`bool`, optional) – Доступен ли альбом для пользователей с подпиской.
- `version` (`str`, optional) – Дополнительная информация об альбоме.
- `cover_uri` (`str`, optional) – Ссылка на обложку.
- `content_warning` (`str`, optional) – Предупреждение о содержимом альбома.
- `genre` (`str`, optional) – Жанр музыки.
- `og_image` (`str`, optional) – Ссылка на превью Open Graph.
- `recent` (`bool`, optional) – Является ли альбом новым.
- `very_important` (`bool`, optional) – Популярен ли альбом у слушателей.
- `available_for_mobile` (`bool`, optional) – Доступен ли альбом из приложения для телефона.
- `available_partially` (`bool`, optional) – Доступен ли альбом частично для пользователей без подписки.
- `bests` (`list` из `int`, optional) – ID лучших треков альбома.
- `volumes` (`list` из `list` из `Track`, optional) – Треки альбома, разделенные по дискам.
- `year` (`int`, optional) – Год релиза.
- `release_date` (`str`, optional) – Дата релиза в формате ISO 8601.
- `type` (`str`, optional) – Тип альбома.
- `track_position` (`yandex_music.TrackPosition`, optional) – Объект класса `yandex_music.TrackPosition` представляющий позицию трека. Возвращается при получении альбома в составе трека.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[Album]`

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Album*.

Тип результата *yandex_music.Album*

`classmethod de_list(data: dict, client: Client) → List[Album]`

Десериализация списка объектов.

Параметры

- `data (list)` – Список словарей с полями и значениями десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Album*.

Тип результата `list` из *yandex_music.Album*

`dislike(*args, **kwargs) → bool`

Сокращение для:

`client.users_likes_albums_remove(album.id, user.id *args, **kwargs)`

`downloadCover(filename: str, size: str = '200x200') → None`

Псевдоним для *download_cover*

`downloadOgImage(filename: str, size: str = '200x200') → None`

Псевдоним для *download_og_image*

`download_cover(filename: str, size: str = '200x200') → None`

Загрузка обложки.

Параметры

- `filename (str)` – Путь для сохранения файла с названием и расширением.
- `size (str, optional)` – Размер обложки.

`download_og_image(filename: str, size: str = '200x200') → None`

Загрузка обложки.

Предпочтительнее использовать *self.download_cover()*.

Параметры

- `filename (str)` – Путь для сохранения файла с названием и расширением.
- `size (str, optional)` – Размер обложки.

`like(*args, **kwargs) → bool`

Сокращение для:

`client.users_likes_albums_add(album.id, user.id *args, **kwargs)`

`withTracks(*args, **kwargs) → Optional[yandex_music.album.album.Album]`

Псевдоним для *with_tracks*

`with_tracks(*args, **kwargs) → Optional[yandex_music.album.album.Album]`

Сокращение для:

`client.albums_with_tracks(album.id, *args, **kwargs)`

2.16.2 `yandex_music.TrackPosition`

```
class yandex_music.TrackPosition(volume: int, index: int, client: Optional[Client] = None,
                               **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[TrackPosition]
```

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.TrackPosition`.

Тип результата `yandex_music.TrackPosition`

2.16.3 `yandex_music.Label`

```
class yandex_music.Label(id_: int, name: str, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий лейбл альбома.

`id`

Идентификатор альбома.

Типе `int`

`name`

Название альбома.

Типе `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `id` (int) – Идентификатор альбома.
- `name` (str) – Название альбома.

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Label]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Label*.

Тип результата *yandex_music.Label*

classmethod `de_list(data: dict, client: Client) → List[Label]`

Десериализация списка объектов.

Параметры

- `data` (list) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Label*.

Тип результата list из *yandex_music.Label*

2.17 Пакет `yandex_music.artist`

2.17.1 `yandex_music.Counts`

```
class yandex_music.Counts(tracks: int, direct_albums: int, also_albums: int, also_tracks: int,
                        client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий счётчик некоторых значений исполнителя.

Примечание: Под дополнительными подразумеваются треки и альбомы, в которых артист задействован, но не является его автором. Так же в дополнительные альбомы входят сборники.

`tracks`

Количество треков.

Тип int

`direct_albums`

Количество альбомов.

Тип int

`also_albums`

Количество дополнительных треков.

Тип `int`

`also_tracks`

Количество дополнительных альбомов.

Тип `int`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `tracks (int)` – Количество треков.
- `direct_albums (int)` – Количество альбомов.
- `also_albums (int)` – Количество дополнительных треков.
- `also_tracks (int)` – Количество дополнительных альбомов.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Counts]`

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Counts`.

Тип результата `yandex_music.Counts`

2.17.2 `yandex_music.Link`

```
class yandex_music.Link(title: str, href: str, type_: str, social_network: Optional[str] = None,
                       client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий ссылку на официальную страницу исполнителя.

Примечание: Известные типы страниц: *official* - официальный сайт и *social* - социальная сеть.

`title`

Название страницы.

Тип `str`

`href`

URL страницы.

Тип `str`

`type_`
Тип страницы.

Типе `str`

`social_network`
Название социальной сети.

Типе `str`

`client`
Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `title (str)` – Название страницы.
- `href (str)` – URL страницы.
- `type (str)` – Тип страницы.
- `social_network (str, optional)` – Название социальной сети.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

`classmethod de_json(data: dict, client: Client) → Optional[Link]`
Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Link`.

Тип результата `yandex_music.Link`

`classmethod de_list(data: dict, client: Client) → List[Link]`
Десериализация списка объектов.

Параметры

- `data (list)` – Список словарей с полями и значениями десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.Link`.

Тип результата `list` из `yandex_music.Link`

2.17.3 yandex_music.Artist

```
class yandex_music.Artist(id_: int, error: Optional[str] = None, name: Optional[str] =
    None, cover: Optional[Cover] = None, various: Optional[bool]
    = None, composer=None, genres=None, op_image=None,
    no_pictures_from_search=None, counts: Optional[Counts] = None,
    available: Optional[bool] = None, ratings: Optional[Ratings] = None,
    links: List[Link] = None, tickets_available: Optional[bool] = None,
    likes_count: Optional[int] = None, popular_tracks: List[Track]
    = None, regions=None, decomposed=None, full_names=None,
    description: Optional[Description] = None, countries=None,
    en_wikipedia_link=None, db_aliases=None, aliases=None, init_date:
    Optional[str] = None, end_date=None, client: Optional[Client] =
    None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Artist]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Artist`.

Тип результата `yandex_music.Artist`

classmethod `de_list(data: dict, client: Client) → List[Artist]`

Десериализация списка объектов.

Параметры

- `data` (list) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.Artist`.

Тип результата list из `yandex_music.Artist`

`dislike(*args, **kwargs) → bool`

Сокращение для:

`client.users_likes_artists_remove(artist.id, user.id *args, **kwargs)`

`downloadOpImage(filename: str, size: str = '200x200') → None`

Псевдоним для `download_op_image`

`download_op_image(filename: str, size: str = '200x200') → None`

Загрузка обложки.

Используйте это только когда нет `self.cover!`

Параметры

- `filename` (`str`) – Путь для сохранения файла с названием и расширением.
- `size` (`str`, optional) – Размер обложки.

`getAlbums(page=0, page_size=20, sort_by='year', *args, **kwargs) → Optional[ArtistAlbums]`

Псевдоним для `get_albums`

`getTracks(page=0, page_size=20, *args, **kwargs) → Optional[ArtistTracks]`

Псевдоним для `get_tracks`

`get_albums(page=0, page_size=20, sort_by='year', *args, **kwargs) → Optional[ArtistAlbums]`

Сокращение для:

`client.artists_direct_albums(artist.id, page, page_size, sort_by, *args, **kwargs)`

`get_tracks(page=0, page_size=20, *args, **kwargs) → Optional[ArtistTracks]`

Сокращение для:

`client.artists_tracks(artist.id, page, page_size, *args, **kwargs)`

`like(*args, **kwargs) → bool`

Сокращение для:

`client.users_likes_artists_add(artist.id, user.id *args, **kwargs)`

2.17.4 yandex_music.Ratings

```
class yandex_music.Ratings(month: int, week: Optional[int] = None, day: Optional[int] = None,
                           client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий рейтинг исполнителя.

`month`

Значение ежемесячного рейтинга.

Тип `int`

`week`

Значение еженедельного рейтинга.

Тип `int`

`day`

Значение дневного рейтинга.

Тип `int`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `month (int)` – Значение ежемесячного рейтинга.
- `week (int, optional)` – Значение еженедельного рейтинга.
- `day (int, optional)` – Значение дневного рейтинга.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[Ratings]`
 Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Ratings`.

Тип результата `yandex_music.Ratings`

2.17.5 yandex_music.Vinyl

```
class yandex_music.Vinyl(url: str, title: str, year: int, price: int, media: str, picture: Optional[str]
                        = None, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий виниловую пластинку.

`url`

Ссылка на пластинку в магазине.

Тип `str`

`title`

Заголовок.

Тип `str`

`year`

Год выпуска.

Тип `int`

`price`

Цена.

Тип `int`

`media`

Средство распространения.

Тип `str`

`picture`

Ссылка на обложку.

Тип `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `url (str)` – Ссылка на пластинку в магазине.
- `title (str)` – Заголовок.
- `year (int)` – Год выпуска.
- `price (int)` – Цена.
- `media (str)` – Средство распространения.
- `picture (str, optional)` – Ссылка на обложку.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Vinyl]`

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Vinyl`.

Тип результата `yandex_music.Vinyl`

classmethod `de_list(data: dict, client: Client) → List[Vinyl]`

Десериализация списка объектов.

Параметры

- `data (list)` – Список словарей с полями и значениями десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.Vinyl`.

Тип результата `list` из `yandex_music.Vinyl`

2.17.6 `yandex_music.Description`

class `yandex_music.Description(text: str, url: str, client: Optional[Client] = None, **kwargs)`

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий описание исполнителя из другого источника.

Примечание: Очень редкий объект, у минимального количества исполнителей. Обычно берётся информация из википедии.

`text`
Описание исполнителя.
Тип `str`

`url`
Ссылка на источник.
Тип `str`

`client`
Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
Тип `yandex_music.Client`

Параметры

- `text (str)` – Описание исполнителя.
- `url (str)` – Ссылка на источник.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Description]`
Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Descriptions`.

Тип результата `yandex_music.Description`

2.17.7 `yandex_music.BriefInfo`

```
class yandex_music.BriefInfo(artist: Optional[Artist], albums: List[Album], also_albums:
    List[Album], last_release_ids: List[int], popular_tracks:
    List[Track], similar_artists: List[Artist], all_covers: List[Cover],
    concerts, videos: List[Video], vinyls: List[Vinyl], has_promotions:
    bool, playlist_ids: List[PlaylistId], tracks_in_chart: List[Chart] =
    None, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`
Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
Тип `yandex_music.Client`

Параметры

- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

- ****kwargs** – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[BriefInfo]`
 Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.BriefInfo`.

Тип результата `yandex_music.BriefInfo`

2.17.8 `yandex_music.ArtistTracks`

class `yandex_music.ArtistTracks(tracks: List[Track], pager: Optional[Pager], client: Optional[Client] = None, **kwargs)`

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий страницу списка треков артиста.

`tracks`

Список треков артиста.

Тип list из `yandex_music.Track`

`pager`

Объект класса `yandex_music.Pager` представляющий пагинатор.

Тип `yandex_music.Pager`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `tracks` (list из `yandex_music.Track`) – Список треков артиста.
- `pager` (`yandex_music.Pager`) – Объект класса `yandex_music.Pager` представляющий пагинатор.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- ****kwargs** – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[ArtistTracks]`
 Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.ArtistsTracks`.

Тип результата `yandex_music.ArtistsTracks`

2.17.9 `yandex_music.ArtistAlbums`

```
class yandex_music.ArtistAlbums(albums: List[Album], pager: Optional[Pager], client:
    Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий страницу списка альбомов артиста.

`albums`

Список альбомов артиста.

Тип `list` из `yandex_music.Album`

`pager`

Объект класса `yandex_music.Pager` представляющий пагинатор.

Тип `yandex_music.Pager`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `albums` (`list` из `yandex_music.Album`) – Список альбомов артиста.
- `pager` (`yandex_music.Pager`) – Объект класса `yandex_music.Pager` представляющий пагинатор.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[ArtistAlbums]
```

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.ArtistAlbums`.

Тип результата `yandex_music.ArtistAlbums`

2.18 Пакет `yandex_music.feed`

2.18.1 `yandex_music.Feed`

```
class yandex_music.Feed(can_get_more_events: bool, pumpkin: bool, is_wizard_passed: bool,
    generated_playlists: List[GeneratedPlaylist], headlines: list, today:
    str, days: List[Day], next_revision: Optional[str] = None, client:
    Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Feed]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Feed`.

Тип результата `yandex_music.Feed`

2.18.2 `yandex_music.TrackWithAds`

```
class yandex_music.TrackWithAds(type_: str, track: Optional[Track], client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий трек с рекламой.

Примечание: Поле `type` встречалось только с значением `track`.

`type`

Тип TODO.

Типе `str`

`track`

Объект класса `yandex_music.Track`, представляющий трек.

Типе `yandex_music.Track`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `type` (str) – Тип TODO.
- `track` (`yandex_music.Track`) – Объект класса `yandex_music.Track`, представляющий трек.

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[TrackWithAds]`
 Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.TrackWithAds*.

Тип результата *yandex_music.TrackWithAds*

classmethod `de_list(data: dict, client: Client) → List[TrackWithAds]`
 Десериализация списка объектов.

Параметры

- `data` (list) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.TrackWithAds*.

Тип результата list из *yandex_music.TrackWithAds*

2.18.3 yandex_music.GeneratedPlaylist

```
class yandex_music.GeneratedPlaylist(type_: str, ready: bool, notify: bool, data:
    Optional[Playlist], client: Optional[Client] = None,
    **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий автоматически сгенерированный плейлист.

Примечание: Известные значения `type`: *playlistOfTheDay*, *origin*, *recentTracks*, *neverHeard*, *podcasts*, *missedLikes*.

`type`
 Тип сгенерированного плейлиста.

Тип str

`ready`
 Готовность плейлиста.

Тип bool

`notify`
 Уведомлён ли пользователь об обновлении содержания.

Тип bool

`data`

Объект класса `yandex_music.Playlist`, представляющий сгенерированный плейлист.

Типе `yandex_music.Playlist`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `type (str)` – Тип сгенерированного плейлиста.
- `ready (bool)` – Готовность плейлиста.
- `notify (bool)` – Уведомлён ли пользователь об обновлении содержания.
- `data (yandex_music.Playlist, optional)` – Объект класса `yandex_music.Playlist`, представляющий сгенерированный плейлист.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[GeneratedPlaylist]`

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.GeneratedPlaylist`.

Тип результата `yandex_music.GeneratedPlaylist`

classmethod `de_list(data: dict, client: Client) → List[GeneratedPlaylist]`

Десериализация списка объектов.

Параметры

- `data (list)` – Список словарей с полями и значениями десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.GeneratedPlaylist`.

Тип результата `list` из `yandex_music.GeneratedPlaylist`

2.18.4 `yandex_music.AlbumEvent`

```
class yandex_music.AlbumEvent(album: Optional[Album], tracks: List[Track], client:
                             Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[AlbumEvent]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.AlbumEvent`.

Тип результата `yandex_music.AlbumEvent`

classmethod `de_list(data: dict, client: Client) → List[AlbumEvent]`

Десериализация списка объектов.

Параметры

- `data` (list) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.AlbumEvent`.

Тип результата list из `yandex_music.AlbumEvent`

2.18.5 yandex_music.ArtistEvent

```
class yandex_music.ArtistEvent(artist: Optional[Artist], tracks: List[Track],
                              similar_to_artists_from_history: List[Artist], client:
                              Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[ArtistEvent]`
 Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.ArtistEvent`.

Тип результата `yandex_music.ArtistEvent`

classmethod `de_list(data: dict, client: Client) → List[ArtistEvent]`
 Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.ArtistEvent`.

Тип результата `list` из `yandex_music.ArtistEvent`

2.18.6 yandex_music.Event

```
class yandex_music.Event(id_: str, type_: str, type_for_from: Optional[str] = None,
                        title: Optional[str] = None, tracks: List[Track] = None, artists:
                        List[ArtistEvent] = None, albums: List[AlbumEvent] = None,
                        message=None, device=None, tracks_count: Optional[int] = None,
                        client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Event]`
 Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Event`.

Тип результата *yandex_music.Event*

classmethod `de_list(data: dict, client: Client) → List[Event]`

Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Event*.

Тип результата `list` из *yandex_music.Event*

2.18.7 *yandex_music.Day*

```
class yandex_music.Day(day: str, events: List[Event], tracks_to_play_with_ads:
    List[TrackWithAds], tracks_to_play: List[Track], client: Optional[Client]
    = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий .

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Типе *yandex_music.Client*

Параметры

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Day]`

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Day*.

Тип результата *yandex_music.Day*

classmethod `de_list(data: dict, client: Client) → List[Day]`

Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.Day`.

Тип результата `list` из `yandex_music.Day`

2.19 Пакет `yandex_music.genre`

2.19.1 `yandex_music.Images`

```
class yandex_music.Images(_208x208: Optional[str] = None, _300x300: Optional[str] = None,
                          client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий изображение жанра.

`_208x208`

Ссылка на изображение размером 208 на 208.

Тип `str`

`_300x300`

Ссылка на изображение размером 300 на 300.

Тип `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `_208x208` (`str`, optional) – Ссылка на изображение размером 208 на 208.
- `_300x300` (`str`, optional) – Ссылка на изображение размером 300 на 300.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Images]
```

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Images`.

Тип результата `yandex_music.Images`

```
download_208x208(filename: str) → None
```

Загрузка изображения 208x208.

Параметры `filename` (`str`) – Путь для сохранения файла с названием и расширением.

```
download_300x300(filename: str) → None
```

Загрузка изображения 300x300.

Параметры `filename (str)` – Путь для сохранения файла с названием и расширением.

2.19.2 `yandex_music.Genre`

```
class yandex_music.Genre(id_: str, weight: int, composer_top: bool, title: str, titles: Dict[str,
Optional[Title]], images: Optional[Images], show_in_menu: bool,
full_title: Optional[str] = None, url_part: Optional[str] = None, color:
Optional[str] = None, radio_icon: Optional[Icon] = None, sub_genres:
List[Genre] = None, hide_in_regions=None, client: Optional[Client] =
None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий жанр музыки.

`id`

Уникальный идентификатор жанра.

Type `str`

`weight`

Вес TODO (возможно, чем выше показатель, тем больше нравится пользователю).

Type `int`

`composer_top`

TODO.

Type `bool`

`title`

Заголовок жанра.

Type `str`

`titles`

Словарь заголовков на разных языках, где ключ - язык.

Type `dict`

`images`

Объект класса `yandex_music.Images`, представляющий изображение жанра.

Type `yandex_music.Images`

`show_in_menu`

Показывать в меню.

Type `bool`

`full_title`

Полный заголовок.

Type `str`

`url_part`

Часть ссылки на жанр для открытия в браузере.

Type `str`

`color`

Цвет фона изображения.

Типе `str`

`radio_icon`

Объект класса `yandex_music.Images`, представляющий иконку радио жанра.

Типе `yandex_music.Icon`

`sub_genres`

Список объектов класса `yandex_music.Genre`, представляющие поджанры текущего жанра музыки.

Типе `list` из `yandex_music.Genre`

`hide_in_regions`

В каких регионах скрывать жанр.

Типе `list`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `id (str)` – Уникальный идентификатор жанра.
- `weight (int)` – Вес TODO (возможно, чем выше показатель, тем больше нравится пользователю).
- `composer_top (bool)` – TODO.
- `title (str)` – Заголовок жанра.
- `titles (dict)` – Словарь заголовков на разных языках, где ключ - язык.
- `images (yandex_music.Images)` – Объект класса `yandex_music.Images`, представляющий изображение жанра.
- `show_in_menu (bool)` – Показывать в меню.
- `full_title (str, optional)` – Полный заголовок.
- `url_part (str, optional)` – Часть ссылки на жанр для открытия в браузере.
- `color (str, optional)` – Цвет фона изображения.
- `radio_icon (yandex_music.Icon, optional)` – Объект класса `yandex_music.Images`, представляющий иконку радио жанра.
- `sub_genres (list из yandex_music.Genre, optional)` – Список объектов класса `yandex_music.Genre`, представляющие поджанры текущего жанра музыки.
- `hide_in_regions (list, optional)` – В каких регионах скрывать жанр.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Genre]`

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.

- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Genre*.

Тип результата *yandex_music.Genre*

`classmethod de_list(data: dict, client: Client) → List[Genre]`

Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Genre*.

Тип результата `list` из *yandex_music.Genre*

2.19.3 *yandex_music.Title*

```
class yandex_music.Title(title: str, full_title: Optional[str] = None, client: Optional[Client] = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий заголовок жанра.

`title`

Заголовок.

Тип `str`

`full_title`

Полный заголовок.

Тип `str`

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `title` (`str`) – Заголовок.
- `full_title` (`str`, `optional`) – Полный заголовок.
- `client` (*yandex_music.Client*, `optional`) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_dict(data, client) → Dict[str, Optional[yandex_music.genre.title.Title]]`

Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.

- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Title*.

Тип результата `list` из *yandex_music.Title*

`classmethod de_json(data: dict, client: Client) → Optional[Title]`

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Title*.

Тип результата *yandex_music.Title*

2.20 Пакет *yandex_music.landing*

2.20.1 *yandex_music.BlockEntity*

```
class yandex_music.BlockEntity(id_: str, type_: str, data: Union[GeneratedPlaylist, Promotion,
    Album, Playlist, ChartItem, PlayContext, MixLink, None], client:
    Optional[Client] = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий .

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[BlockEntity]`

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.BlockEntity*.

Тип результата *yandex_music.BlockEntity*

`classmethod de_list(data: dict, client: Client) → List[BlockEntity]`

Десериализация списка объектов.

Параметры

- `data (list)` – Список словарей с полями и значениями десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.BlockEntity`.

Тип результата `list` из `yandex_music.BlockEntity`

2.20.2 yandex_music.PlayContextsData

```
class yandex_music.PlayContextsData(other_tracks: List[TrackShortOld], client: Optional[Client]
    = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[PlayContextsData]
```

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.PlayContextsData`.

Тип результата `yandex_music.PlayContextsData`

2.20.3 yandex_music.Chart

```
class yandex_music.Chart(position: int, progress: str, listeners: int, shift: int, track_id:
    Optional[TrackId] = None, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[Chart]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Chart*.

Тип результата *yandex_music.Chart*

`classmethod de_list(data: dict, client: Client) → List[Chart]`

Десериализация списка объектов.

Параметры

- `data` (list) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Chart*.

Тип результата list из *yandex_music.Chart*

2.20.4 yandex_music.TrackShortOld

```
class yandex_music.TrackShortOld(track_id: Optional[TrackId], timestamp: str, client: Optional[Client] = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий .

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[TrackShortOld]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.TrackShortOld`.

Тип результата `yandex_music.TrackShortOld`

```
classmethod de_list(data: dict, client: Client) → List[TrackShortOld]
```

Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.TrackShortOld`.

Тип результата `list` из `yandex_music.TrackShortOld`

2.20.5 `yandex_music.Landing`

```
class yandex_music.Landing(pumpkin: bool, content_id: Union[str, int], blocks: List[Block], client:
    Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Landing]
```

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Landing`.

Тип результата `yandex_music.Landing`

2.20.6 `yandex_music.ChartItem`

```
class yandex_music.ChartItem(track: Optional[Track], chart: Optional[Chart], client:
    Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[ChartItem]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.ChartItem`.

Тип результата `yandex_music.ChartItem`

classmethod `de_list(data: dict, client: Client) → List[ChartItem]`

Десериализация списка объектов.

Параметры

- `data` (list) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.ChartItem`.

Тип результата list из `yandex_music.ChartItem`

2.20.7 `yandex_music.PlayContext`

```
class yandex_music.PlayContext(client_: str, context: str, context_item: str, tracks: List[TrackShortOld], client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[PlayContext]`
 Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.PlayContext`.

Тип результата `yandex_music.PlayContext`

2.20.8 `yandex_music.MixLink`

```
class yandex_music.MixLink(title: str, url: str, url_scheme: str, text_color: str, background_color: str, background_image_uri: str, cover_white: str, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[MixLink]`
 Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.MixLink`.

Тип результата `yandex_music.MixLink`

classmethod `de_list(data: dict, client: Client) → List[MixLink]`
 Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.MixLink`.

Тип результата `list` из `yandex_music.MixLink`

`downloadBackgroundImage(filename: str, size: str = '200x200') → None`

Псевдоним для `download_background_image`

`download_background_image(filename: str, size: str = '200x200') → None`

Загрузка заднего фона.

Параметры

- `filename (str)` – Путь для сохранения файла с названием и расширением.
- `size (str, optional)` – Размер заднего фона.

2.20.9 `yandex_music.TrackId`

```
class yandex_music.TrackId(id_: int, album_id: Optional[int] = None, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[TrackId]
```

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.TrackId`.

Тип результата `yandex_music.TrackId`

2.20.10 `yandex_music.PersonalPlaylistsData`

```
class yandex_music.PersonalPlaylistsData(is_wizard_passed: bool, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[PersonalPlaylistsData]`
Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.PersonalPlaylistsData*.

Тип результата *yandex_music.PersonalPlaylistsData*

2.20.11 yandex_music.Promotion

```
class yandex_music.Promotion(promo_id: str, title: str, subtitle: str, heading: str, url: str,
                             url_scheme: str, text_color: str, gradient: str, image: str, client:
                             Optional[Client] = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий .

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Типе *yandex_music.Client*

Параметры

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Promotion]`
Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Promotion*.

Тип результата *yandex_music.Promotion*

classmethod `de_list(data: dict, client: Client) → List[Promotion]`
Десериализация списка объектов.

Параметры

- `data` (list) – Список словарей с полями и значениями десериализуемого объекта.

- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Promotion*.

Тип результата `list` из *yandex_music.Promotion*

2.20.12 *yandex_music.Block*

```
class yandex_music.Block(id_: str, type_: str, type_for_from: str, title: str, entities:
    List[BlockEntity], description: Optional[str] = None, data:
    Union[PersonalPlaylistsData, PlayContextsData, None] = None,
    client: Optional[Client] = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий .

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Block]
```

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Block*.

Тип результата *yandex_music.Block*

```
classmethod de_list(data: dict, client: Client) → List[Block]
```

Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Block*.

Тип результата `list` из *yandex_music.Block*

2.21 Пакет `yandex_music.likes`

2.21.1 `yandex_music.ArtistsLikes`

2.21.2 `yandex_music.AlbumsLikes`

2.21.3 `yandex_music.PlaylistsLikes`

2.22 Пакет `yandex_music.playlist`

2.22.1 `yandex_music.User`

```
class yandex_music.User(uid: int, login: str, name: str, sex: str, verified: bool, client: Optional[Client]
                       = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий пользователя.

`uid`

Идентификатор пользователя.

Type `int`

`login`

Логин пользователя.

Type `str`

`name`

Имя пользователя.

Type `str`

`sex`

Пол пользователя.

Type `str`

`verified`

Участвует ли пользователь в генерации плейлистов дня и т.д., и т.п.

Type `bool`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Type `yandex_music.Client`

Параметры

- `uid` (`int`) – Идентификатор пользователя.
- `login` (`str`) – Логин пользователя.
- `name` (`str`) – Имя пользователя.
- `sex` (`str`) – Пол пользователя.
- `verified` (`bool`) – Участвует ли пользователь в генерации плейлистов дня и т.д., и т.п.

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[User]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.User*.

Тип результата *yandex_music.User*

`downloadAvatar(filename: str, format_: str = 'normal') → None`

Псевдоним для `download_avatar`

`download_avatar(filename: str, format_: str = 'normal') → None`

Загрузка изображения пользователя.

Параметры

- `filename` (str) – Путь для сохранения файла с названием и расширением.
- `format` (str, optional) – Формат желаемого изображения (*normal, orig, small, big*).

2.22.2 yandex_music.MadeFor

class `yandex_music.MadeFor(user_info: Optional[User], case_forms: Optional[CaseForms], client: Optional[Client] = None, **kwargs)`

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий пользователя, для которого был сделан плейлист.

`user_info`

Объект класса *yandex_music.User*, представляющий пользователя, для которого был сделан плейлист.

Тип *yandex_music.User*

`case_forms`

Объект класса *yandex_music.CaseForms*, представляющий склонение имени пользователя, для которого был сделан плейлист.

Тип *yandex_music.CaseForms*

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `user_info` (*yandex_music.User*) – Объект класса *yandex_music.User*, представляющий пользователя, для которого был сделан плейлист.

- `case_forms` (*yandex_music.CaseForms*) – Объект класса *yandex_music.CaseForms*, представляющий склонение имени пользователя, для которого был сделан плейлист.
- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[MadeFor]`
 Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.MadeFor*.

Тип результата *yandex_music.MadeFor*

2.22.3 yandex_music.PlayCounter

```
class yandex_music.PlayCounter(value: int, description: str, updated: bool, client: Optional[Client]
                             = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий счётчик дней.

Примечание: Присутствует только у плейлиста дня. Счётчик считает количество дней подряд, на протяжении которых был прослушан плейлист.

`value`

Значение (количество дней).

Тип `int`

`description`

Описание счётчика.

Тип `str`

`updated`

Обновлён ли сегодня (в этих сутках).

Тип `bool`

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `value` (`int`) – Значение (количество дней).
- `description` (`str`) – Описание счётчика.
- `updated` (`bool`) – Обновлён ли сегодня (в этих сутках).

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[PlayCounter]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.PlayCounter*.

Тип результата *yandex_music.PlayCounter*

2.22.4 yandex_music.PlaylistAbsence

```
class yandex_music.PlaylistAbsence(kind: int, reason: str, client: Optional[Client] = None,
                                   **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий причину отсутствия плейлиста.

`kind`

Уникальный идентификатор плейлиста.

Тип `int`

`reason`

Причина отсутствия.

Тип `str`

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `kind` (int) – Уникальный идентификатор плейлиста.
- `reason` (str) – Причина отсутствия.
- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[PlaylistAbsence]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.PlaylistAbsence*.

Тип результата *yandex_music.PlaylistAbsence*

2.22.5 *yandex_music.Playlist*

```
class yandex_music.Playlist(owner: Optional[User], cover: Optional[Cover], made_for:
    Optional[MadeFor], play_counter: Optional[PlayCounter],
    playlist_absence: Optional[PlaylistAbsence], uid: Optional[int]
    = None, kind: Optional[int] = None, title: Optional[str] = None,
    track_count: Optional[int] = None, tags: Optional[list] = None,
    revision: Optional[int] = None, snapshot: Optional[int] = None,
    visibility: Optional[str] = None, collective: Optional[bool] =
    None, created: Optional[str] = None, modified: Optional[str] =
    None, available: Optional[bool] = None, is_banner: Optional[bool]
    = None, is_premiere: Optional[bool] = None, duration_ms:
    Optional[int] = None, og_image: Optional[str] = None, tracks:
    List[TrackShort] = None, prerolls: Optional[list] = None,
    likes_count: Optional[int] = None, generated_playlist_type:
    Optional[str] = None, animated_cover_uri: Optional[str]
    = None, ever_played: Optional[bool] = None, description:
    Optional[str] = None, description_formatted: Optional[str] =
    None, is_for_from=None, regions=None, client: Optional[Client]
    = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий .

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Типе *yandex_music.Client*

Параметры

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[Playlist]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Playlist*.

Тип результата *yandex_music.Playlist*

`classmethod de_list(data: dict, client: Client) → List[Playlist]`

Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.Playlist`.

Тип результата `list` из `yandex_music.Playlist`

`dislike(*args, **kwargs) → bool`

Сокращение для:

`client.users_likes_playlists_remove(playlist.uid, user.id *args, **kwargs)`

`downloadAnimatedCover(filename: str, size: str = '200x200') → None`

Псевдоним для `download_animated_cover`

`downloadOgImage(filename: str, size: str = '200x200') → None`

Псевдоним для `download_og_image`

`download_animated_cover(filename: str, size: str = '200x200') → None`

Загрузка анимированной обложки.

Параметры

- `filename` (`str`) – Путь для сохранения файла с названием и расширением (GIF).
- `size` (`str`, optional) – Размер анимированной обложки.

`download_og_image(filename: str, size: str = '200x200') → None`

Загрузка обложки.

Используйте это только когда нет `self.cover!`

Параметры

- `filename` (`str`) – Путь для сохранения файла с названием и расширением.
- `size` (`str`, optional) – Размер обложки.

`isMine`

Псевдоним для `is_mine`

`is_mine`

`like(*args, **kwargs) → bool`

Сокращение для:

`client.users_likes_playlists_add(playlist.uid, user.id *args, **kwargs)`

`playlistId`

Псевдоним для `playlist_id`

`playlist_id`

`rename(name: str) → None`

2.22.6 yandex_music.CaseForms

```
class yandex_music.CaseForms(nominative: str, genitive: str, dative: str, accusative: str,
                             instrumental: str, prepositional: str, client: Optional[Client] = None,
                             **kwargs)
```


Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий склонение имени.

`nominative`

Именительный.

Type `str`

`genitive`

Родительный.

Type `str`

`dative`

Дательный.

Type `str`

`accusative`

Винительный.

Type `str`

`instrumental`

Творительный.

Type `str`

`prepositional`

Предложный.

Type `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Type `yandex_music.Client`

Параметры

- `nominative (str)` – Именительный.
- `genitive (str)` – Родительный.
- `dative (str)` – Дательный.
- `accusative (str)` – Винительный.
- `instrumental (str)` – Творительный.
- `prepositional (str)` – Предложный.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[CaseForms]`

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.CaseForms*.

Тип результата *yandex_music.CaseForms*

2.22.7 *yandex_music.PlaylistId*

```
class yandex_music.PlaylistId(uid: int, kind: int, client: Optional[Client] = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий уникальный идентификатор плейлиста.

uid

Уникальный идентификатор пользователя владеющим плейлистом.

Тип *int*

kind

Уникальный идентификатор плейлиста.

Тип *int*

client

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- **uid** (*int*) – Уникальный идентификатор пользователя владеющим плейлистом.
- **kind** (*int*) – Уникальный идентификатор плейлиста.
- **client** (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- ****kwargs** – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[PlaylistId]
```

Десериализация объекта.

Параметры

- **data** (*dict*) – Поля и значения десериализуемого объекта.
- **client** (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.PlaylistId*.

Тип результата *yandex_music.PlaylistId*

```
classmethod de_list(data: dict, client: Client) → List[PlaylistId]
```

Десериализация списка объектов.

Параметры

- **data** (*list*) – Список словарей с полями и значениями десериализуемого объекта.
- **client** (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.PlaylistId*.

Тип результата `list` из `yandex_music.PlaylistId`

2.23 Пакет `yandex_music.rotor`

2.23.1 `yandex_music.RotorSettings`

```
class yandex_music.RotorSettings(language: str, diversity: str, mood: Optional[int] = None,
                                energy: Optional[int] = None, mood_energy: Optional[str] =
                                None, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий настройки станции.

Примечание: Поля `energy`, `mood` используются в старых настройках (`settings1`).

Значения `mood_energy`: `fun`, `active`, `calm`, `sad`, `all`.

Значения `diversity`: `favorite`, `popular`, `discover`, `default`.

Значения `language`: `not-russian`, `russian`, `any`.

`language`

Язык.

Тип `str`

`diversity`

Разнообразие (треки).

Тип `str`

`mood`

Настроение (старое).

Тип `int`

`energy`

Энергичное.

Тип `int`

`mood_energy`

Настроение.

Тип `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `language (str)` – Язык.
- `diversity (str)` – Разнообразие (треки).
- `mood (int, optional)` – Настроение (старое).
- `energy (int, optional)` – Энергичное.

- `mood_energy` (`str`, optional) – Настройка.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[RotorSettings]`
Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.RotorSettings`.

Тип результата `yandex_music.RotorSettings`

2.23.2 `yandex_music.DiscreteScale`

```
class yandex_music.DiscreteScale(type_: str, name: str, min_: Optional[Value], max_: Optional[Value], client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий дискретное значение.

Примечание: Известные значения поля `type`: `discrete-scale`.

`type`

Тип.

Тип `str`

`name`

Название.

Тип `str`

`min`

Объект класса `yandex_music.Value`, представляющий минимальное значение.

Тип `yandex_music.Value`

`max`

Объект класса `yandex_music.Value`, представляющий максимальное значение.

Тип `yandex_music.Value`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `type` (`str`) – Тип.
- `name` (`str`) – Название.

- `min` (*yandex_music.Value*) – Объект класса *yandex_music.Value*, представляющий минимальное значение.
- `max` (*yandex_music.Value*) – Объект класса *yandex_music.Value*, представляющий максимальное значение.
- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[DiscreteScale]`
 Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.DiscreteScale*.

Тип результата *yandex_music.DiscreteScale*

2.23.3 yandex_music.Enum

```
class yandex_music.Enum(type_: str, name: str, possible_values: List[Value], client: Optional[Client]
                       = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий перечисления.

`type`

Тип перечисления.

Тип `str`

`name`

Название перечисления.

Тип `str`

`possible_values`

Список из *yandex_music.Value*, представляющий доступные значения.

Тип `list` из *yandex_music.Value*

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `type` (str) – Тип перечисления.
- `name` (str) – Название перечисления.
- `possible_values` (list из *yandex_music.Value*) – Список из *yandex_music.Value*, представляющий доступные значения.

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Enum]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Enum*.

Тип результата *yandex_music.Enum*

2.23.4 yandex_music.Id

class `yandex_music.Id(type_: str, tag: str, client: Optional[Client] = None, **kwargs)`

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий уникальный идентификатор станции.

Примечание: Известные типы станций: *user*, *genre*.

`type`

Тип станции.

Тип `str`

`tag`

Тег станции.

Тип `str`

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `type` (str) – Тип станции.
- `tag` (str) – Тег станции.
- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Id]`

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.

- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Id*.

Тип результата *yandex_music.Id*

2.23.5 *yandex_music.Sequence*

```
class yandex_music.Sequence(type_: str, track: Optional[Track], liked: bool, client: Optional[Client]
                           = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий .

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Sequence]
```

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Sequence*.

Тип результата *yandex_music.Sequence*

```
classmethod de_list(data: dict, client: Client) → List[Sequence]
```

Десериализация списка объектов.

Параметры

- `data` (list) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Sequence*.

Тип результата list из *yandex_music.Sequence*

2.23.6 yandex_music.Station

```
class yandex_music.Station(id_: Optional[Id], name: str, icon: Optional[Icon], mts_icon:
    Optional[Icon], geocell_icon: Optional[Icon], id_for_from:
    str, restrictions: Optional[Restrictions], restrictions2:
    Optional[Restrictions], parent_id: Optional[Id] = None, client:
    Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий станцию.

Примечание: `id_for_from` обозначает предка станции, например, жанр, настроение или занятие. Неизвестно когда используется `id_for_from`, а когда `parent_id`.

`id`

Объект класса `yandex_music.Id`, представляющий уникальный идентификатор станции.

Типе `yandex_music.Id`

`name`

Название станции.

Типе `str`

`icon`

Объект класса `yandex_music.Icon`, представляющий иконку станции.

Типе `yandex_music.Icon`

`mts_icon`

Объект класса `yandex_music.Icon`, представляющий иконку TODO.

Типе `yandex_music.Icon`

`geocell_icon`

Объект класса `yandex_music.Icon`, представляющий иконку TODO.

Типе `yandex_music.Icon`

`id_for_from`

Категория (тип) станции.

Типе `str`

`restrictions`

Объект класса `yandex_music.Restrictions`, представляющий ограничения для настроек станции старого формата.

Типе `yandex_music.Restrictions`

`restrictions2`

Объект класса `yandex_music.Restrictions`, представляющий ограничения для настроек станции.

Типе `yandex_music.Restrictions`

`parent_id`

Объект класса `yandex_music.Id`, представляющий уникальный идентификатор станции, являющейся предком текущей.

Типе `yandex_music.Id`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `id` (`yandex_music.Id`) – Объект класса `yandex_music.Id`, представляющий уникальный идентификатор станции.
- `name` (`str`) – Название станции.
- `icon` (`yandex_music.Icon`) – Объект класса `yandex_music.Icon`, представляющий иконку станции.
- `mts_icon` (`yandex_music.Icon`) – Объект класса `yandex_music.Icon`, представляющий иконку TODO.
- `geocell_icon` (`yandex_music.Icon`) – Объект класса `yandex_music.Icon`, представляющий иконку TODO.
- `id_for_from` (`str`) – Категория (тип) станции.
- `restrictions` (`yandex_music.Restrictions`) – Объект класса `yandex_music.Restrictions`, представляющий ограничения для настроек станции старого формата.
- `restrictions2` (`yandex_music.Restrictions`) – Объект класса `yandex_music.Restrictions`, представляющий ограничения для настроек станции.
- `parent_id` (`yandex_music.Id`, optional) – Объект класса `yandex_music.Id`, представляющий уникальный идентификатор станции, являющейся предком текущей.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Station]`

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Station`.

Тип результата `yandex_music.Station`

2.23.7 yandex_music.AdParams

```
class yandex_music.AdParams(partner_id: Union[str, int], category_id: Union[str, int], page_ref: str, target_ref: str, other_params: str, ad_volume: int, genre_id: Optional[str] = None, genre_name: Optional[str] = None, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий параметры рекламного объявления.

Примечание: Известные дополнительные параметры(*other_params*): *user:{ID}*.

`partner_id`

Уникальный идентификатор заказчика рекламы.

Type `str | int`

`category_id`

Уникальный идентификатор категории рекламы.

Type `str | int`

`page_ref`

Ссылка на ссылающуюся страницу.

Type `str`

`target_ref`

Ссылка на целевую страницу.

Type `str`

`other_params`

Другие параметры.

Type `str`

`ad_volume`

Громкость воспроизводимой рекламы.

Type `int`

`genre_id`

Уникальный идентификатор жанра.

Type `str`

`genre_name`

Название жанра.

Type `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Type `yandex_music.Client`

Параметры

- `partner_id` (`str | int`) – Уникальный идентификатор заказчика рекламы.
- `category_id` (`str | int`) – Уникальный идентификатор категории рекламы.
- `page_ref` (`str`) – Ссылка на ссылающуюся страницу.
- `target_ref` (`str`) – Ссылка на целевую страницу.
- `other_params` (`str`) – Другие параметры.
- `ad_volume` (`int`) – Громкость воспроизводимой рекламы.
- `genre_id` (`str`, optional) – Уникальный идентификатор жанра.

- `genre_name` (`str`, optional) – Название жанра.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[AdParams]`
 Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.AdParams`.

Тип результата `yandex_music.AdParams`

2.23.8 `yandex_music.Dashboard`

```
class yandex_music.Dashboard(dashboard_id: str, stations: List[StationResult], pumpkin: bool,
                             client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий рекомендованные станции пользователя.

`dashboard_id`

Уникальный идентификатор панели.

Тип `str`

`stations`

Список объектов класса `yandex_music.StationResult`, представляющих станции со всеми возможными настройками и параметрами.

Тип `list` из `yandex_music.StationResult`

`pumpkin`

Хэллоуин.

Тип `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `dashboard_id` (`str`) – Уникальный идентификатор панели.
- `stations` (`list` из `yandex_music.StationResult`) – Список объектов класса `yandex_music.StationResult`, представляющих станции со всеми возможными настройками и параметрами.
- `pumpkin` (`str`) – Хэллоуин.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

- ****kwargs** – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Dashboard]`
Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Dashboard`.

Тип результата `yandex_music.Dashboard`

2.23.9 `yandex_music.Value`

class `yandex_music.Value(value: str, name: str, client: Optional[Client] = None, **kwargs)`

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий значение(переменную).

`value`

Значение.

Тип `str`

`name`

Название.

Тип `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `value` (`str`) – Значение.
- `name` (`str`) – Название.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- ****kwargs** – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Value]`
Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Value`.

Тип результата `yandex_music.Value`

classmethod `de_list(data: dict, client: Client) → List[Value]`
 Десериализация списка объектов.

Параметры

- `data (list)` – Список словарей с полями и значениями десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.Value`.

Тип результата `list` из `yandex_music.Value`

2.23.10 yandex_music.Restrictions

```
class yandex_music.Restrictions(language: Optional[Enum], diversity: Optional[Enum],
                               mood: Optional[DiscreteScale] = None, energy:
                               Optional[DiscreteScale] = None, mood_energy: Optional[Enum]
                               = None, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий ограничения для настроек станции.

`language`

Объект класса `yandex_music.Enum`, представляющий перечисление значений для языков.

Типе `yandex_music.Enum`

`diversity`

Объект класса `yandex_music.Enum`, представляющий перечисление значений для разнообразия (треков).

Типе `yandex_music.Enum`

`mood`

Объект класса `yandex_music.DiscreteScale`, представляющий ограничения для значения настроения.

Типе `yandex_music.DiscreteScale`

`energy`

Объект класса `yandex_music.DiscreteScale`, представляющий ограничения для значения энергичности.

Типе `yandex_music.DiscreteScale`

`mood_energy`

Объект класса `yandex_music.Enum`, представляющий перечисление значений для настроения.

Типе `yandex_music.Enum`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `language` (*yandex_music.Enum*) – Объект класса *yandex_music.Enum*, представляющий перечисление значений для языков.
- `diversity` (*yandex_music.Enum*) – Объект класса *yandex_music.Enum*, представляющий перечисление значений для разнообразия (треков).
- `mood` (*yandex_music.DiscreteScale*, optional) – Объект класса *yandex_music.DiscreteScale*, представляющий ограничения для значения настроения.
- `energy` (*yandex_music.DiscreteScale*, optional) – Объект класса *yandex_music.DiscreteScale*, представляющий ограничения для значения энергичности.
- `mood_energy` (*yandex_music.Enum*, optional) – Объект класса *yandex_music.Enum*, представляющий перечисление значений для настроения.
- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[Restrictions]`
 Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Restrictions*.

Тип результата *yandex_music.Restrictions*

2.23.11 yandex_music.StationResult

```
class yandex_music.StationResult(station: Optional[Station], settings: Optional[RotorSettings],
                                settings2: Optional[RotorSettings], ad_params:
                                Optional[AdParams], explanation: Optional[str] = None,
                                prerolls: Optional[list] = None, client: Optional[Client] =
                                None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий .

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[StationResult]`
 Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.StationResult`.

Тип результата `yandex_music.StationResult`

```
classmethod de_list(data: dict, client: Client) → List[StationResult]
```

Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.StationResult`.

Тип результата `list` из `yandex_music.StationResult`

2.23.12 `yandex_music.StationTracksResult`

```
class yandex_music.StationTracksResult(id_: Optional[Id], sequence: List[Sequence], batch_id: str,
pumpkin: bool, client: Optional[Client] = None,
**kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data, client) → Optional[yandex_music.rotor.station_tracks_result.StationTracksResult]
```

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.StationTracksResult`.

Тип результата `yandex_music.StationTracksResult`

2.24 Пакет `yandex_music.search`

2.24.1 `yandex_music.Suggestions`

```
class yandex_music.Suggestions(best: Optional[Best], suggestions: List[str], client:
    Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий подсказки при поиске.

best

Объект класса `yandex_music.Best`, представляющий лучший результат.

Тип `yandex_music.Best`

suggestions

Список подсказок-дополнений для поискового запроса.

Тип list из str

client

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- **best** (`yandex_music.Best`) – Объект класса `yandex_music.Best`, представляющий лучший результат.
- **suggestions** (list из str) – Список подсказок-дополнений для поискового запроса.
- **client** (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- ****kwargs** – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Suggestions]
```

Десериализация объекта.

Параметры

- **data** (dict) – Поля и значения десериализуемого объекта.
- **client** (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Suggestions`.

Тип результата `yandex_music.Suggestions`

2.24.2 `yandex_music.Search`

```
class yandex_music.Search(search_request_id: str, text: str, best: Optional[Best], albums:
    Optional[SearchResult], artists: Optional[SearchResult], playlists:
    Optional[SearchResult], tracks: Optional[SearchResult], videos:
    Optional[SearchResult], misspell_corrected: Optional[bool] = None,
    nocorrect: Optional[bool] = None, client: Optional[Client] = None,
    **kwargs)
```


Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий результаты поиска.

`search_request_id`

ID запроса.

Type `str`

`text`

Текст запроса.

Type `str`

`best`

Объект класса `yandex_music.Best` представляющий лучший результат.

Type `yandex_music.Best`

`albums`

Объект класса `yandex_music.SearchResult` представляющий найденные альбомы.

Type `yandex_music.SearchResult`

`artists`

Объект класса `yandex_music.SearchResult` представляющий найденных исполнителей.

Type `yandex_music.SearchResult`

`playlists`

Объект класса `yandex_music.SearchResult` представляющий найденные плейлисты.

Type `yandex_music.SearchResult`

`tracks`

Объект класса `yandex_music.SearchResult` представляющий найденные треки.

Type `yandex_music.SearchResult`

`videos`

Объект класса `yandex_music.SearchResult` представляющий найденные видео.

Type `yandex_music.SearchResult`

`misspell_corrected`

Был ли исправлен запрос.

Type `bool`

`nocorrect`

Было ли отключено исправление результата.

Type `bool`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Type `yandex_music.Client`

Параметры

- `search_request_id` (`str`) – ID запроса.
- `text` (`str`) – Текст запроса.
- `best` (`yandex_music.Best`) – Объект класса `yandex_music.Best` представляющий лучший результат.

- `albums` (*yandex_music.SearchResult*) – Объект класса *yandex_music.SearchResult* представляющий найденные альбомы.
- `artists` (*yandex_music.SearchResult*) – Объект класса *yandex_music.SearchResult* представляющий найденных исполнителей.
- `playlists` (*yandex_music.SearchResult*) – Объект класса *yandex_music.SearchResult* представляющий найденные плейлисты.
- `tracks` (*yandex_music.SearchResult*) – Объект класса *yandex_music.SearchResult* представляющий найденные треки.
- `videos` (*yandex_music.SearchResult*) – Объект класса *yandex_music.SearchResult* представляющий найденные видео.
- `misspell_corrected` (bool, optional) – Был ли исправлен запрос.
- `nocorrect` (bool, optional) – Было ли отключено исправление результата.
- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Search]`
 Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.Search*.

Тип результата *yandex_music.Search*

2.24.3 yandex_music.SearchResult

```
class yandex_music.SearchResult(type_: str, total: int, per_page: int, order: int,
                               results: List[Union[yandex_music.track.track.Track,
                                                    yandex_music.artist.artist.Artist,
                                                    yandex_music.album.album.Album,
                                                    yandex_music.playlist.playlist.Playlist,
                                                    yandex_music.video.Video]], client: Optional[Client] = None,
                               **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий результаты поиска.

Примечание: Значения поля *type*: *track*, *artist*, *playlist*, *album*, *video*.

`type`

Тип результата.

Type `str`

`total`

Количество результатов.

Тип `int`

per_page
Максимальное количество результатов на странице.

Тип `int`

order
Позиция блока.

Тип `int`

results
Результаты поиска.

Тип `list` из `yandex_music.Track` | `yandex_music.Artist` | `yandex_music.Album` | `yandex_music.Playlist` | `yandex_music.Video`

client
Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `type (str)` – Тип результата.
- `total (int)` – Количество результатов.
- `per_page (int)` – Максимальное количество результатов на странице.
- `order (int)` – Позиция блока.
- `results (list` из `yandex_music.Track` | `yandex_music.Artist` | `yandex_music.Album` | `yandex_music.Playlist` | `yandex_music.Video`) – Результаты поиска.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client, type_: str = None) → Optional[SearchResult]`
Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `type (str, optional)` – Тип результата.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.SearchResult`.

Тип результата `yandex_music.SearchResult`

2.24.4 `yandex_music.Best`

```
class yandex_music.Best(type_: str, result: Union[yandex_music.track.track.Track,
yandex_music.artist.artist.Artist, yandex_music.album.album.Album,
yandex_music.playlist.playlist.Playlist, yandex_music.video.Video,
None], text: Optional[str] = None, client: Optional[Client] = None,
**kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий лучший результат поиска.

`type_`

Тип лучшего результата.

Типе `str`

`result`

Лучший результат.

Типе `yandex_music.Track | yandex_music.Artist | yandex_music.Album | yandex_music.Playlist | yandex_music.Video`

`text`

TODO.

Типе `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `type (str)` – Тип лучшего результата.
- `result (yandex_music.Track | yandex_music.Artist | yandex_music.Album | yandex_music.Playlist | yandex_music.Video)` – Лучший результат.
- `text (str, optional)` – TODO.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Best]`

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Best`.

Тип результата `yandex_music.Best`

2.25 Пакет `yandex_music.shot`

2.25.1 `yandex_music.Shot`

```
class yandex_music.Shot(order: int, played: bool, shot_data: ShotData, shot_id: str, status: str,
                       client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий шот от Алисы.

Примечание: Известные значения поля *status*: *ready*.

`order`

Порядковый номер при воспроизведении.

Type `int`

`played`

Был ли проигран шот.

Type `bool`

`shot_data`

Объект класса `yandex_music.ShotData` представляющий основную информацию о шоте.

Type `yandex_music.ShotData`

`shot_id`

Уникальный идентификатор шота.

Type `str`

`status`

Статус шота.

Type `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Type `yandex_music.Client`

Параметры

- `order` (`int`) – Порядковый номер при воспроизведении.
- `played` (`bool`) – Был ли проигран шот.
- `shot_data` (`yandex_music.ShotData`) – Объект класса `yandex_music.ShotData` представляющий основную информацию о шоте.
- `shot_id` (`str`) – Уникальный идентификатор шота.
- `status` (`str`) – Статус шота.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[Shot]`

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Shot`.

Тип результата *yandex_music.Shot*

```
classmethod de_list(data: dict, client: Client) → List[Shot]
```

Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.
- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Shot*.

Тип результата `list` из *yandex_music.Shot*

2.25.2 *yandex_music.ShotData*

```
class yandex_music.ShotData(cover_uri: str, mds_url: str, shot_text: str, shot_type: ShotType,
                           client: Optional[Client] = None, **kwargs)
```

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий основную информацию о шоте.

`cover_uri`

Ссылка на обложку шота (иконка Алисы).

Тип `str`

`mds_url`

Ссылка на аудиоверсию шота в озвучке от Алисы.

Тип `str`

`shot_text`

Текстовая версия шота.

Тип `str`

`shot_type`

Объект класса *yandex_music.ShotType* представляющий тип шота.

Тип *yandex_music.ShotType*

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `cover_uri` (`str`) – Ссылка на обложку шота (иконка Алисы).
- `mds_url` (`str`) – Ссылка на аудиоверсию шота в озвучке от Алисы.
- `shot_text` (`str`) – Текстовая версия шота.
- `shot_type` (*yandex_music.ShotType*) – Объект класса *yandex_music.ShotType* представляющий тип шота.
- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

- ****kwargs** – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[ShotData]`
 Десериализация объекта.

Параметры

- **data** (dict) – Поля и значения десериализуемого объекта.
- **client** (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.ShotData*.

Тип результата *yandex_music.ShotData*

`downloadCover(filename: str, size: str = '200x200') → None`
 Псевдоним для *download_cover*

`downloadMds(filename: str) → None`
 Псевдоним для *download_mds*

`download_cover(filename: str, size: str = '200x200') → None`
 Загрузка обложки.

Параметры

- **filename** (str) – Путь для сохранения файла с названием и расширением.
- **size** (str, optional) – Размер обложки.

`download_mds(filename: str) → None`
 Загрузка аудиоверсии шота.

Параметры filename (str) – Путь для сохранения файла с названием и расширением.

2.25.3 yandex_music.ShotEvent

`class yandex_music.ShotEvent(event_id: str, shots: List[Shot], client: Optional[Client] = None, **kwargs)`

Базовые классы: *yandex_music.base.YandexMusicObject*

Класс, представляющий событие-шот перед началом следующего трека.

event_id
 Уникальный идентификатор события.

Тип str

shots
 Список объектов класса *yandex_music.Shot* представляющие шоты от Алисы.

Тип list из *yandex_music.Shot*

client
 Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- **event_id** (str) – Уникальный идентификатор события.

- `shots` (`list` из `yandex_music.Shot`) – Список объектов класса `yandex_music.Shot` представляющие шоты от Алисы.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[ShotEvent]`
Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.ShotEvent`.

Тип результата `yandex_music.ShotEvent`

2.25.4 `yandex_music.ShotType`

`class yandex_music.ShotType(id_: str, title: str, client: Optional[Client] = None, **kwargs)`

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий тип шота от Алисы.

`id`

Уникальный идентификатор типа.

Тип `str`

`title`

Заголовок шота.

Тип `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `id` (`str`) – Уникальный идентификатор типа.
- `title` (`str`) – Заголовок шота.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[ShotType]`
Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.

- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Объект класса *yandex_music.ShotType*.

Тип результата *yandex_music.ShotType*

2.26 Пакет `yandex_music.supplement`

2.26.1 `yandex_music.Supplement`

```
class yandex_music.Supplement(id_: int, lyrics: Optional[Lyrics], videos: List[VideoSupplement],
                             radio_is_available: bool, client: Optional[Client] = None,
                             **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий дополнительную информацию о треке.

`id`

Уникальный идентификатор дополнительной информации.

Тип `int`

`lyrics`

Объект класса *yandex_music.Lyrics* представляющий текст песни.

Тип *yandex_music.Lyrics*

`videos`

Объект класса *yandex_music.VideoSupplement* представляющий видео.

Тип *yandex_music.VideoSupplement*

`radio_is_available`

Доступно ли радио.

Тип `bool`

`client`

Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Тип *yandex_music.Client*

Параметры

- `id` (`int`) – Уникальный идентификатор дополнительной информации.
- `lyrics` (*yandex_music.Lyrics*) – Объект класса *yandex_music.Lyrics* представляющий текст песни.
- `videos` (*yandex_music.VideoSupplement*) – Объект класса *yandex_music.VideoSupplement* представляющий видео.
- `radio_is_available` (`bool`) – Доступно ли радио.
- `client` (*yandex_music.Client*, optional) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Supplement]`
Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Supplement`.

Тип результата `yandex_music.Supplement`

2.26.2 `yandex_music.Lyrics`

```
class yandex_music.Lyrics(id_: int, lyrics: str, full_lyrics: str, has_rights: bool, text_language: str,
                          show_translation: bool, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий текст трека.

`id`

Уникальный идентификатор текста трека.

Тип `int`

`lyrics`

Первые строки текст песни.

Тип `str`

`has_rights`

Есть ли права.

Тип `bool`

`full_lyrics`

Текст песни.

Тип `str`

`text_language`

Язык текста.

Тип `str`

`show_translation`

Показывать ли перевод.

Тип `bool`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `id` (`int`) – Уникальный идентификатор текста трека.
- `lyrics` (`str`) – Первые строки текст песни.
- `has_rights` (`bool`) – Есть ли права.

- `full_lyrics (str)` – Текст песни.
- `text_language (str)` – Язык песни.
- `show_translation (bool)` – Показывать ли перевод.
- `client (yandex_music.Client, optional)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

classmethod `de_json(data: dict, client: Client) → Optional[Lyrics]`
 Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Lyrics`.

Тип результата `yandex_music.Lyrics`

2.26.3 `yandex_music.VideoSupplement`

```
class yandex_music.VideoSupplement(cover: str, title: str, provider: str, provider_video_id: str,
                                   url: Optional[str] = None, embed_url: Optional[str] = None,
                                   embed: Optional[str] = None, client: Optional[Client] =
                                   None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий видеоклипы.

`cover`

URL на обложку видео.

Type `str`

`title`

Название видео.

Type `str`

`provider`

Сервис поставляющий видео.

Type `str`

`provider_video_id`

Уникальный идентификатор видео на сервисе.

Type `str`

`url`

URL на видео.

Type `str`

`embed_url`

URL на видео, находящегося на серверах Яндекса.

Type `str`

`embed`

HTML тег для встраивания видео.

Типе `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `cover (str)` – URL на обложку видео.
- `title (str)` – Название видео.
- `provider (str)` – Сервис поставляющий видео.
- `provider_video_id (str)` – Уникальный идентификатор видео на сервисе.
- `url (str)` – URL на видео.
- `embed_url (str)` – URL на видео, находящегося на серверах Яндекса.
- `embed (str)` – HTML тег для встраивания видео.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[VideoSupplement]`

Десериализация объекта.

Параметры

- `data (dict)` – Поля и значения десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.VideoSupplement`.

Тип результата `yandex_music.VideoSupplement`

`classmethod de_list(data: dict, client: Client) → List[VideoSupplement]`

Десериализация списка объектов.

Параметры

- `data (list)` – Список словарей с полями и значениями десериализуемого объекта.
- `client (yandex_music.Client)` – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Список объектов класса `yandex_music.VideoSupplement`.

Тип результата `list` из `yandex_music.VideoSupplement`

2.27 Пакет `yandex_music.track`

2.27.1 `yandex_music.Normalization`

```
class yandex_music.Normalization(gain: float, peak: int, client: Optional[Client] = None,
                               **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Normalization]
```

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Normalization`.

Тип результата `yandex_music.Normalization`

2.27.2 `yandex_music.Major`

```
class yandex_music.Major(id_: int, name: str, client: Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Major]
```

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Major`.

Тип результата `yandex_music.Major`

2.27.3 `yandex_music.Track`

```
class yandex_music.Track(id_: Union[str, int], title: str, available: bool, artists: List[Artist],
                        albums: List[Album], available_for_premium_users: Optional[bool] =
                        None, lyrics_available: Optional[bool] = None, real_id: Union[str,
                        int, None] = None, og_image: Optional[str] = None, type_:
                        Optional[str] = None, cover_uri: Optional[str] = None, major:
                        Optional[Major] = None, duration_ms: Optional[int] = None,
                        storage_dir: Optional[str] = None, file_size: Optional[int] =
                        None, normalization: Optional[Normalization] = None, error=None,
                        regions=None, available_as_rbt=None, content_warning=None,
                        explicit=None, preview_duration_ms: Optional[int] = None,
                        available_full_without_permission: Optional[bool] = None, client:
                        Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

`classmethod de_json(data: dict, client: Client) → Optional[Track]`

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.Track`.

Тип результата `yandex_music.Track`

`classmethod de_list(data: dict, client: Client) → List[Track]`

Десериализация списка объектов.

Параметры

- `data` (`list`) – Список словарей с полями и значениями десериализуемого объекта.

- `client` (*yandex_music.Client*) – Объект класса *yandex_music.Client*, представляющий клиент Yandex Music.

Результат Список объектов класса *yandex_music.Track*.

Тип результата `list` из *yandex_music.Track*

`dislike(*args, **kwargs) → bool`

Сокращение для:

`client.users_likes_tracks_remove(track.id, user.id *args, **kwargs)`

`download(filename: str, codec: str = 'mp3', bitrate_in_kbps: int = 192) → None`

`downloadCover(filename: str, size: str = '200x200') → None`

Псевдоним для *download_cover*

`downloadOgImage(filename: str, size: str = '200x200') → None`

Псевдоним для *download_og_image*

`download_cover(filename: str, size: str = '200x200') → None`

Загрузка обложки.

Параметры

- `filename` (`str`) – Путь для сохранения файла с названием и расширением.
- `size` (`str`, optional) – Размер обложки.

`download_og_image(filename: str, size: str = '200x200') → None`

Загрузка обложки.

Предпочтительнее использовать *self.download_cover()*.

Параметры

- `filename` (`str`) – Путь для сохранения файла с названием и расширением.
- `size` (`str`, optional) – Размер обложки.

`getDownloadInfo(get_direct_links=False) → List[DownloadInfo]`

Псевдоним для *get_download_info*

`getSupplement(*args, **kwargs) → Optional[Supplement]`

Псевдоним для *get_supplement*

`get_download_info(get_direct_links=False) → List[DownloadInfo]`

`get_supplement(*args, **kwargs) → Optional[Supplement]`

Сокращение для:

`client.track_supplement(track.id, *args, **kwargs)`

`like(*args, **kwargs) → bool`

Сокращение для:

`client.users_likes_tracks_add(track.id, user.id, *args, **kwargs)`

`trackId`

Псевдоним для *track_id*

`track_id`

2.28 Пакет `yandex_music.utils`

2.28.1 `yandex_music.utils.request.Request`

```
class yandex_music.utils.request.Request(client=None, headers=None, proxy_url=None)
```

Базовые классы: `object`

Вспомогательный класс для `yandex_music` представляющий методы для выполнения POST и GET запросов, скачивания файлов.

Параметры

- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `headers` (dict, optional) – Заголовки передаваемые с каждым запросом.
- `proxy_url` (str, optional) – Прокси.

2.28.2 `yandex_music.utils.response.Response`

```
class yandex_music.utils.response.Response(data, invocation_info=None, result=None,
                                          error=None, error_description=None, client:
                                          Optional[Client] = None, **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий .

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Типе `yandex_music.Client`

Параметры

- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client) → Optional[Response]
```

Десериализация объекта.

Параметры

- `data` (dict) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.utils.response.Response`.

Тип результата `yandex_music.utils.response.Response`

`error`

`result`

2.28.3 yandex_music.utils.captcha_response.CaptchaResponse

```
class yandex_music.utils.captcha_response.CaptchaResponse(x_captcha_url, x_captcha_key,
                                                         error_description, error, client:
                                                         Optional[Client] = None,
                                                         **kwargs)
```

Базовые классы: `yandex_music.base.YandexMusicObject`

Класс, представляющий ответ сервера с запросом на ввод капчи.

`x_captcha_url`

Ссылка на изображение с капчей.

Тип `str`

`x_captcha_key`

Уникальный ключ капчи.

Тип `str`

`error_description`

Описание ошибки.

Тип `str`

`error`

Код ошибки.

Тип `str`

`client`

Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Тип `yandex_music.Client`

Параметры

- `x_captcha_url` (`str`) – Ссылка на изображение с капчей.
- `x_captcha_key` (`str`) – Уникальный ключ капчи.
- `error_description` (`str`) – Описание ошибки.
- `error` (`str`) – Код ошибки.
- `client` (`yandex_music.Client`, optional) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.
- `**kwargs` – Произвольные ключевые аргументы полученные от API.

```
classmethod de_json(data: dict, client: Client)
```

Десериализация объекта.

Параметры

- `data` (`dict`) – Поля и значения десериализуемого объекта.
- `client` (`yandex_music.Client`) – Объект класса `yandex_music.Client`, представляющий клиент Yandex Music.

Результат Объект класса `yandex_music.utils.captcha_response.CaptchaResponse`.

Тип результата `yandex_music.utils.captcha_response.CaptchaResponse`

`download(filename=None)`

Загрузка изображения с капчей.

Параметры `filename` (`str`, optional) – Путь и(или) название файла вместе с расширением. По умолчанию ключ капчи и расширение `.gif`.

2.28.4 `yandex_music.utils.difference.Difference`

`class yandex_music.utils.difference.Difference`

Базовые классы: `object`

`add_delete(from_, to)`

`add_insert(at, tracks: dict)`

`to_json()`

3.1 Версия 0.0.16

29.12.2019

Переломные изменения

- Поле `account` переименовано в `me` и теперь содержит объект `Status`, вместо `Account` (#162).
- Убрано использование зарезервированных имён в аргументах конструкторов (теперь они с `_` на конце). Имена с нижними подчёркиваниями есть как при сериализации так и при десериализации (#168).

Крупные изменения

- **Добавлены аннотации типов во всей библиотеке!**

Незначительные изменения и/или исправления

- Добавлен аргумент `fetch_account_status` для опциональности получения информации об аккаунте при инициализации клиента (#162).
- Добавлены тесты с передачей пустого словаря в `de_json` и `de_list` (#174).
- Использование `ujson` при наличии, обновлены зависимости (#161).
- Добавлен в зависимости для разработки `importlib_metadata` для поддержки старых версий (в новой версии `pytest` его больше не используют, в угоду `importlib.metadata` #pytest-5537) (#161).
- Добавлен в зависимости для разработки `atomicwrites`, который используется `pytest` теперь только на Windows - #pytest-6148 (#161).
- Исправлен баг с передачей `timeout` аргумента в аргумент `params` в следующих методах: `artists`, `albums`, `playlists_list` (#120).
- Исправлена инициализация клиента при помощи логина и пароля с использованием прокси (#159).
- Исправлен баг в загрузке обложки альбома.

3.2 Версия 0.0.15

01.12.2019

Переломные изменения

- У классов `Artist`, `Track` и `Playlist` изменился перечень полей для генерации хеша.

Крупные изменения

- Добавлена возможность выполнять запросы через прокси-сервер для использования библиотеки `ProxyAgent`.
 - Добавлен пример использования в README.
- Добавлена обработка капчи при авторизации с возможностью использования callback-функции для обработки ответа.
 - Новые исключения:
 - * `Captcha`:
 - `CaptchaRequired`.
 - `CaptchaWrong`.
 - Новые классы:
 - * `CaptchaResponse`.
 - Новые примеры в README:
 - * Пример обработки с использованием callback-функции.
 - * Пример полностью своей обработки капчи.
 - Добавлена документация для класса `Search` (#83).
 - Добавлена возможность получения всех альбомов исполнителя (#141):
 - Новые классы:
 - * `ArtistAlbums`.
 - Новые методы:
 - * `artists_direct_albums` у `Client`.
 - * `get_albums` у `Artist`.
 - Добавлена обработка несуществующего плейлиста (#147):
 - Новые классы:
 - * `PlaylistAbsence`.

Незначительные изменения и/или исправления

- Исправлен баг с загрузкой файлов (#149).
- Исправлен баг некорректной десериализации плейлиста при отсутствии прав на него (#147).
- Исправлен баг неправильной десериализации треков и артистов у собственных загруженных файлов (#154).

3.3 Версия 0.0.14

10.11.2019

Переломные изменения

- Практически у всех классов был обновлён список полей участвующих при сравнении объектов.
- Если в атрибутах для сравнения объектов присутствуют списки, то они будут преобразованы к `frozenset`.
- Убрано конвертирование даты из строки в объект. Теперь все даты представляны строками в ISO формате.
- Классы `AlbumSearchResult`, `ArtistSearchResult`, `PlaylistSearchResult`, `TrackSearchResult`, `VideoSearchResult` были объединены в один - `SearchResult`.

Крупные изменения

- Добавлен метод получения треков исполнителя (#123).
- Добавлены классы-обёртки над пагинацией (`Pager`) и списка треков артиста (`ArtistsTracks`).
- Добавлено 554 unit-теста для всех классов-обёрток над объектами API.
- Добавлен `codecov` и `workflows` для GitHub Actions.

Незначительные изменения и/или исправления

- Поле `cover_uri` класса `Album` теперь опциональное.
- Поле `region` у класса `Account` теперь не обязательное.
- Исправлен баг в `.to_dict()` методе, связанный с десериализацией объектов списков и словарей.
- Исправлен баг в `.to_dict()` методе, связанный с не рекурсивной десериализацией.
- Исправлена десериализация `similar_artists` в `BriefInfo`.
- Исправлен баг с десериализацией `artist` в классе `ArtistEvent`.
- Исправлен баг десериализации списка альбомов и артистов у класса `Track` (#122).
- Исправлена загрузка обложки у трека.
- Исправлены сравнения объектов.

СИМВОЛЫ

- название_эксперимента (атрибут *yandex_music.Experiments*), 35
- _208x208 (атрибут *yandex_music.Images*), 82
- _300x300 (атрибут *yandex_music.Images*), 82
- ## А
- account (атрибут *yandex_music.Status*), 45
- Account (класс в *yandex_music*), 47
- account_experiments() (метод *yandex_music.Client*), 8
- account_settings() (метод *yandex_music.Client*), 8
- account_settings_set() (метод *yandex_music.Client*), 8
- account_status (атрибут *yandex_music.PromoCodeStatus*), 39
- account_status() (метод *yandex_music.Client*), 9
- accountExperiments() (метод *yandex_music.Client*), 8
- accountStatus() (метод *yandex_music.Client*), 8
- accusative (атрибут *yandex_music.CaseForms*), 101
- ad_volume (атрибут *yandex_music.AdParams*), 110
- add_delete() (метод *yandex_music.utils.difference.Difference*), 134
- add_insert() (метод *yandex_music.utils.difference.Difference*), 134
- add_new_track_on_playlist_top (атрибут *yandex_music.UserSettings*), 58
- AdParams (класс в *yandex_music*), 109
- ads_disabled (атрибут *yandex_music.UserSettings*), 59
- after_track() (метод *yandex_music.Client*), 9
- afterTrack() (метод *yandex_music.Client*), 9
- Album (класс в *yandex_music*), 60
- album_id (атрибут *yandex_music.TrackShort*), 41
- AlbumEvent (класс в *yandex_music*), 78
- albums (атрибут *yandex_music.ArtistAlbums*), 75
- albums (атрибут *yandex_music.Search*), 117
- albums() (метод *yandex_music.Client*), 10
- albums_with_tracks() (метод *yandex_music.Client*), 10
- albumsWithTracks() (метод *yandex_music.Client*), 10
- alerts (атрибут *yandex_music.PermissionAlerts*), 38
- also_albums (атрибут *yandex_music.Counts*), 66
- also_tracks (атрибут *yandex_music.Counts*), 67
- amount (атрибут *yandex_music.Price*), 57
- Artist (класс в *yandex_music*), 69
- ArtistAlbums (класс в *yandex_music*), 75
- ArtistEvent (класс в *yandex_music*), 79
- artists (атрибут *yandex_music.Album*), 61
- artists (атрибут *yandex_music.Search*), 117
- artists() (метод *yandex_music.Client*), 11
- artists_brief_info() (метод *yandex_music.Client*), 11
- artists_direct_albums() (метод *yandex_music.Client*), 11
- artists_tracks() (метод *yandex_music.Client*), 11
- artistsBriefInfo() (метод *yandex_music.Client*), 11
- artistsDirectAlbums() (метод *yandex_music.Client*), 11
- artistsTracks() (метод *yandex_music.Client*), 11
- ArtistTracks (класс в *yandex_music*), 74
- auto_play_radio (атрибут *yandex_music.UserSettings*), 59
- auto_renewable (атрибут *yandex_music.Subscription*), 52
- AutoRenewable (класс в *yandex_music*), 53
- available (атрибут *yandex_music.Album*), 61
- available (атрибут *yandex_music.Product*), 55
- available_for_mobile (атрибут *yandex_music*)

- yandex_music.Album*), 62
 - available_for_premium_users (*ampublic yandex_music.Album*), 61
 - available_partially (*ampublic yandex_music.Album*), 62
- B**
- background_color (*ampublic yandex_music.Icon*), 36
 - base_url (*ampublic yandex_music.Client*), 7
 - best (*ampublic yandex_music.Search*), 117
 - best (*ampublic yandex_music.Suggestions*), 116
 - Best (класс в *yandex_music*), 119
 - bests (*ampublic yandex_music.Album*), 62
 - birthday (*ampublic yandex_music.Account*), 47
 - bitrate_in_kbps (*ampublic yandex_music.DownloadInfo*), 33
 - Block (класс в *yandex_music*), 94
 - BlockEntity (класс в *yandex_music*), 86
 - BriefInfo (класс в *yandex_music*), 73
 - button_additional_text (*ampublic yandex_music.Product*), 55
 - button_text (*ampublic yandex_music.Product*), 55
- C**
- cache_limit (*ampublic yandex_music.Status*), 45
 - can_start_trial (*ampublic yandex_music.Subscription*), 52
 - CaptchaResponse (класс в *yandex_music.utils.captcha_response*), 133
 - case_forms (*ampublic yandex_music.MadeFor*), 96
 - CaseForms (класс в *yandex_music*), 100
 - category_id (*ampublic yandex_music.AdParams*), 110
 - Chart (класс в *yandex_music*), 87
 - ChartItem (класс в *yandex_music*), 89
 - client (*ampublic yandex_music.Account*), 48
 - client (*ampublic yandex_music.AdParams*), 110
 - client (*ampublic yandex_music.Album*), 62
 - client (*ampublic yandex_music.AlbumEvent*), 78
 - client (*ampublic yandex_music.Artist*), 69
 - client (*ampublic yandex_music.ArtistAlbums*), 75
 - client (*ampublic yandex_music.ArtistEvent*), 79
 - client (*ampublic yandex_music.ArtistTracks*), 74
 - client (*ampublic yandex_music.AutoRenewable*), 53
 - client (*ampublic yandex_music.Best*), 120
 - client (*ampublic yandex_music.Block*), 94
 - client (*ampublic yandex_music.BlockEntity*), 86
 - client (*ampublic yandex_music.BriefInfo*), 73
 - client (*ampublic yandex_music.CaseForms*), 101
 - client (*ampublic yandex_music.Chart*), 87
 - client (*ampublic yandex_music.ChartItem*), 89
 - client (*ampublic yandex_music.Counts*), 67
 - client (*ampublic yandex_music.Cover*), 32
 - client (*ampublic yandex_music.Dashboard*), 111
 - client (*ampublic yandex_music.Day*), 81
 - client (*ampublic yandex_music.Description*), 73
 - client (*ampublic yandex_music.DiscreteScale*), 104
 - client (*ampublic yandex_music.DownloadInfo*), 34
 - client (*ampublic yandex_music.Enum*), 105
 - client (*ampublic yandex_music.Event*), 80
 - client (*ampublic yandex_music.Experiments*), 35
 - client (*ampublic yandex_music.Feed*), 76
 - client (*ampublic yandex_music.GeneratedPlaylist*), 78
 - client (*ampublic yandex_music.Genre*), 84
 - client (*ampublic yandex_music.Icon*), 36
 - client (*ampublic yandex_music.Id*), 106
 - client (*ampublic yandex_music.Images*), 82
 - client (*ampublic yandex_music.InvocationInfo*), 37
 - client (*ampublic yandex_music.Label*), 65
 - client (*ampublic yandex_music.Landing*), 89
 - client (*ampublic yandex_music.Link*), 68
 - client (*ampublic yandex_music.Lyrics*), 126
 - client (*ampublic yandex_music.MadeFor*), 96
 - client (*ampublic yandex_music.Major*), 129
 - client (*ampublic yandex_music.MixLink*), 91
 - client (*ampublic yandex_music.Normalization*), 129
 - client (*ampublic yandex_music.Pager*), 38
 - client (*ampublic yandex_music.PassportPhone*), 49
 - client (*ampublic yandex_music.PermissionAlerts*), 38
 - client (*ampublic yandex_music.Permissions*), 51
 - client (*ampublic yandex_music.PersonalPlaylistsData*), 92
 - client (*ampublic yandex_music.PlayContext*), 90
 - client (*ampublic yandex_music.PlayContextsData*), 87
 - client (*ampublic yandex_music.PlayCounter*), 97
 - client (*ampublic yandex_music.Playlist*), 99
 - client (*ampublic yandex_music.PlaylistAbsence*), 98
 - client (*ampublic yandex_music.PlaylistId*), 102
 - client (*ampublic yandex_music.Plus*), 50
 - client (*ampublic yandex_music.Price*), 57
 - client (*ampublic yandex_music.Product*), 55
 - client (*ampublic yandex_music.PromoCodeStatus*), 39
 - client (*ampublic yandex_music.Promotion*), 93
 - client (*ampublic yandex_music.Ratings*), 70
 - client (*ampublic yandex_music.Restrictions*), 113

- client (*ампубум* *yandex_music.RotorSettings*), 103
 client (*ампубум* *yandex_music.Search*), 117
 client (*ампубум* *yandex_music.SearchResult*), 119
 client (*ампубум* *yandex_music.Sequence*), 107
 client (*ампубум* *yandex_music.Settings*), 40
 client (*ампубум* *yandex_music.Shot*), 121
 client (*ампубум* *yandex_music.ShotData*), 122
 client (*ампубум* *yandex_music.ShotEvent*), 123
 client (*ампубум* *yandex_music.ShotType*), 124
 client (*ампубум* *yandex_music.Station*), 108
 client (*ампубум* *yandex_music.StationResult*), 114
 client (*ампубум* *yandex_music.StationTracksResult*), 115
 client (*ампубум* *yandex_music.Status*), 46
 client (*ампубум* *yandex_music.Subscription*), 52
 client (*ампубум* *yandex_music.Suggestions*), 116
 client (*ампубум* *yandex_music.Supplement*), 125
 client (*ампубум* *yandex_music.Title*), 85
 client (*ампубум* *yandex_music.Track*), 130
 client (*ампубум* *yandex_music.TrackId*), 92
 client (*ампубум* *yandex_music.TrackPosition*), 65
 client (*ампубум* *yandex_music.TrackShort*), 41
 client (*ампубум* *yandex_music.TrackShortOld*), 88
 client (*ампубум* *yandex_music.TracksList*), 43
 client (*ампубум* *yandex_music.TrackWithAds*), 76
 client (*ампубум* *yandex_music.User*), 95
 client (*ампубум* *yandex_music.UserSettings*), 59
 client (*ампубум* *yandex_music.utils.captcha_response.CaptchaResponse*), 133
 client (*ампубум* *yandex_music.utils.response.Response*), 132
 client (*ампубум* *yandex_music.Value*), 112
 client (*ампубум* *yandex_music.Video*), 44
 client (*ампубум* *yandex_music.VideoSupplement*), 128
 client (*ампубум* *yandex_music.Vinyl*), 71
 Client (класс в *yandex_music*), 7
 codec (*ампубум* *yandex_music.DownloadInfo*), 33
 color (*ампубум* *yandex_music.Genre*), 83
 common_period_duration (*ампубум* *yandex_music.Product*), 54
 composer_top (*ампубум* *yandex_music.Genre*), 83
 consume_promo_code() (*метод* *yandex_music.Client*), 12
 consumePromoCode() (*метод* *yandex_music.Client*), 12
 content_warning (*ампубум* *yandex_music.Album*), 61
 Counts (класс в *yandex_music*), 66
 cover (*ампубум* *yandex_music.VideoSupplement*), 127
 Cover (класс в *yandex_music*), 31
 cover_uri (*ампубум* *yandex_music.Album*), 61
 cover_uri (*ампубум* *yandex_music.ShotData*), 122
 currency (*ампубум* *yandex_music.Price*), 57
 custom (*ампубум* *yandex_music.Cover*), 32

D
 Dashboard (класс в *yandex_music*), 111
 dashboard_id (*ампубум* *yandex_music.Dashboard*), 111
 data (*ампубум* *yandex_music.GeneratedPlaylist*), 77
 dative (*ампубум* *yandex_music.CaseForms*), 101
 day (*ампубум* *yandex_music.Ratings*), 70
 Day (класс в *yandex_music*), 81
 de_dict() (*метод* класса *yandex_music.Title*), 85
 de_json() (*метод* класса *yandex_music.Account*), 48
 de_json() (*метод* класса *yandex_music.AdParams*), 111
 de_json() (*метод* класса *yandex_music.Album*), 63
 de_json() (*метод* класса *yandex_music.AlbumEvent*), 79
 de_json() (*метод* класса *yandex_music.Artist*), 69
 de_json() (*метод* класса *yandex_music.ArtistAlbums*), 75
 de_json() (*метод* класса *yandex_music.ArtistEvent*), 79
 de_json() (*метод* класса *yandex_music.ArtistTracks*), 74
 de_json() (*метод* класса *yandex_music.AutoRenewable*), 54
 de_json() (*метод* класса *yandex_music.Best*), 120
 de_json() (*метод* класса *yandex_music.Block*), 94
 de_json() (*метод* класса *yandex_music.BlockEntity*), 86
 de_json() (*метод* класса *yandex_music.BriefInfo*), 74
 de_json() (*метод* класса *yandex_music.CaseForms*), 101
 de_json() (*метод* класса *yandex_music.Chart*), 88
 de_json() (*метод* класса *yandex_music.ChartItem*), 90
 de_json() (*метод* класса *yandex_music.Counts*), 67
 de_json() (*метод* класса *yandex_music.Cover*), 32
 de_json() (*метод* класса *yandex_music.Dashboard*), 112
 de_json() (*метод* класса *yandex_music.Day*), 81
 de_json() (*метод* класса *yandex_music.Description*), 73
 de_json() (*метод* класса *yandex_music.DiscreteScale*), 105

- de_json() (метод класса *yandex_music.DownloadInfo*), 34
- de_json() (метод класса *yandex_music.Enum*), 106
- de_json() (метод класса *yandex_music.Event*), 80
- de_json() (метод класса *yandex_music.Experiments*), 35
- de_json() (метод класса *yandex_music.Feed*), 76
- de_json() (метод класса *yandex_music.GeneratedPlaylist*), 78
- de_json() (метод класса *yandex_music.Genre*), 84
- de_json() (метод класса *yandex_music.Icon*), 36
- de_json() (метод класса *yandex_music.Id*), 106
- de_json() (метод класса *yandex_music.Images*), 82
- de_json() (метод класса *yandex_music.InvocationInfo*), 37
- de_json() (метод класса *yandex_music.Label*), 66
- de_json() (метод класса *yandex_music.Landing*), 89
- de_json() (метод класса *yandex_music.Link*), 68
- de_json() (метод класса *yandex_music.Lyrics*), 127
- de_json() (метод класса *yandex_music.MadeFor*), 97
- de_json() (метод класса *yandex_music.Major*), 129
- de_json() (метод класса *yandex_music.MixLink*), 91
- de_json() (метод класса *yandex_music.Normalization*), 129
- de_json() (метод класса *yandex_music.Pager*), 38
- de_json() (метод класса *yandex_music.PassportPhone*), 49
- de_json() (метод класса *yandex_music.PermissionAlerts*), 39
- de_json() (метод класса *yandex_music.Permissions*), 51
- de_json() (метод класса *yandex_music.PersonalPlaylistsData*), 93
- de_json() (метод класса *yandex_music.PlayContext*), 90
- de_json() (метод класса *yandex_music.PlayContextsData*), 87
- de_json() (метод класса *yandex_music.PlayCounter*), 98
- de_json() (метод класса *yandex_music.Playlist*), 99
- de_json() (метод класса *yandex_music.PlaylistAbsence*), 98
- de_json() (метод класса *yandex_music.PlaylistId*), 102
- de_json() (метод класса *yandex_music.Plus*), 50
- de_json() (метод класса *yandex_music.Price*), 57
- de_json() (метод класса *yandex_music.Product*), 56
- de_json() (метод класса *yandex_music.PromoCodeStatus*), 40
- de_json() (метод класса *yandex_music.Promotion*), 93
- de_json() (метод класса *yandex_music.Ratings*), 71
- de_json() (метод класса *yandex_music.Restrictions*), 114
- de_json() (метод класса *yandex_music.RotorSettings*), 104
- de_json() (метод класса *yandex_music.Search*), 118
- de_json() (метод класса *yandex_music.SearchResult*), 119
- de_json() (метод класса *yandex_music.Sequence*), 107
- de_json() (метод класса *yandex_music.Settings*), 41
- de_json() (метод класса *yandex_music.Shot*), 121
- de_json() (метод класса *yandex_music.ShotData*), 123
- de_json() (метод класса *yandex_music.ShotEvent*), 124
- de_json() (метод класса *yandex_music.ShotType*), 124
- de_json() (метод класса *yandex_music.Station*), 109
- de_json() (метод класса *yandex_music.StationResult*), 114
- de_json() (метод класса *yandex_music.StationTracksResult*), 115
- de_json() (метод класса *yandex_music.Status*), 46
- de_json() (метод класса *yandex_music.Subscription*), 52
- de_json() (метод класса *yandex_music.Suggestions*), 116
- de_json() (метод класса *yandex_music.Supplement*), 125
- de_json() (метод класса *yandex_music.Title*), 86
- de_json() (метод класса *yandex_music.Track*), 130
- de_json() (метод класса *yandex_music.TrackId*), 92
- de_json() (метод класса *yandex_music.TrackPosition*), 65
- de_json() (метод класса *yandex_music.TrackShort*), 42
- de_json() (метод класса *yandex_music.TrackShortOld*), 88
- de_json() (метод класса *yandex_music.TracksList*), 43

`de_json()` (метод класса `yandex_music.TrackWithAds`), 77
`de_json()` (метод класса `yandex_music.User`), 96
`de_json()` (метод класса `yandex_music.UserSettings`), 60
`de_json()` (метод класса `yandex_music.utils.captcha_response.CaptchaResponse`), 133
`de_json()` (метод класса `yandex_music.utils.response.Response`), 132
`de_json()` (метод класса `yandex_music.Value`), 112
`de_json()` (метод класса `yandex_music.Video`), 44
`de_json()` (метод класса `yandex_music.VideoSupplement`), 128
`de_json()` (метод класса `yandex_music.Vinyl`), 72
`de_json()` (метод класса `yandex_music.YandexMusicObject`), 31
`de_list()` (метод класса `yandex_music.Album`), 64
`de_list()` (метод класса `yandex_music.AlbumEvent`), 79
`de_list()` (метод класса `yandex_music.Artist`), 69
`de_list()` (метод класса `yandex_music.ArtistEvent`), 80
`de_list()` (метод класса `yandex_music.AutoRenewable`), 54
`de_list()` (метод класса `yandex_music.Block`), 94
`de_list()` (метод класса `yandex_music.BlockEntity`), 86
`de_list()` (метод класса `yandex_music.Chart`), 88
`de_list()` (метод класса `yandex_music.ChartItem`), 90
`de_list()` (метод класса `yandex_music.Cover`), 33
`de_list()` (метод класса `yandex_music.Day`), 81
`de_list()` (метод класса `yandex_music.DownloadInfo`), 34
`de_list()` (метод класса `yandex_music.Event`), 81
`de_list()` (метод класса `yandex_music.GeneratedPlaylist`), 78
`de_list()` (метод класса `yandex_music.Genre`), 85
`de_list()` (метод класса `yandex_music.Label`), 66
`de_list()` (метод класса `yandex_music.Link`), 68
`de_list()` (метод класса `yandex_music.MixLink`), 91
`de_list()` (метод класса `yandex_music.PassportPhone`), 49
`de_list()` (метод класса `yandex_music.Playlist`), 99
`de_list()` (метод класса `yandex_music.PlaylistId`), 102
`de_list()` (метод класса `yandex_music.Product`), 56
`de_list()` (метод класса `yandex_music.Promotion`), 93
`de_list()` (метод класса `yandex_music.Sequence`), 107
`de_list()` (метод класса `yandex_music.Shot`), 122
`de_list()` (метод класса `yandex_music.StationResult`), 115
`de_list()` (метод класса `yandex_music.Track`), 130
`de_list()` (метод класса `yandex_music.TrackShort`), 42
`de_list()` (метод класса `yandex_music.TrackShortOld`), 89
`de_list()` (метод класса `yandex_music.TrackWithAds`), 77
`de_list()` (метод класса `yandex_music.Value`), 112
`de_list()` (метод класса `yandex_music.Video`), 44
`de_list()` (метод класса `yandex_music.VideoSupplement`), 128
`de_list()` (метод класса `yandex_music.Vinyl`), 72
`debug` (атрибут `yandex_music.Product`), 55
`default` (атрибут `yandex_music.Permissions`), 51
`default_email` (атрибут `yandex_music.Status`), 45
`description` (атрибут `yandex_music.PlayCounter`), 97
`description` (атрибут `yandex_music.Product`), 55
`Description` (класс в `yandex_music`), 72
`Difference` (класс в `yandex_music.utils.difference`), 134
`dir` (атрибут `yandex_music.Cover`), 32
`direct_albums` (атрибут `yandex_music.Counts`), 66
`direct_link` (атрибут `yandex_music.DownloadInfo`), 34
`DiscreteScale` (класс в `yandex_music`), 104
`disk_enabled` (атрибут `yandex_music.UserSettings`), 59
`dislike()` (метод `yandex_music.Album`), 64
`dislike()` (метод `yandex_music.Artist`), 69
`dislike()` (метод `yandex_music.Playlist`), 100
`dislike()` (метод `yandex_music.Track`), 131
`display_name` (атрибут `yandex_music.Account`), 47
`diversity` (атрибут `yandex_music.Restrictions`), 113
`diversity` (атрибут `yandex_music.RotorSettings`), 103
`download()` (метод `yandex_music.Cover`), 33
`download()` (метод `yandex_music.DownloadInfo`), 35
`download()` (метод `yandex_music.Icon`), 36
`download()` (метод `yandex_music.Track`), 131

- download() (метод *yandex_music.utils.captcha_response.DownloadImage*), 133
- download_208x208() (метод *yandex_music.Images*), 82
- download_300x300() (метод *yandex_music.Images*), 82
- download_animated_cover() (метод *yandex_music.Playlist*), 100
- download_avatar() (метод *yandex_music.Account*), 49
- download_avatar() (метод *yandex_music.User*), 96
- download_background_image() (метод *yandex_music.MixLink*), 92
- download_cover() (метод *yandex_music.Album*), 64
- download_cover() (метод *yandex_music.ShotData*), 123
- download_cover() (метод *yandex_music.Track*), 131
- download_info_url (ампубум *yandex_music.DownloadInfo*), 34
- download_mds() (метод *yandex_music.ShotData*), 123
- download_og_image() (метод *yandex_music.Album*), 64
- download_og_image() (метод *yandex_music.Playlist*), 100
- download_og_image() (метод *yandex_music.Track*), 131
- download_op_image() (метод *yandex_music.Artist*), 70
- downloadAnimatedCover() (метод *yandex_music.Playlist*), 100
- downloadAvatar() (метод *yandex_music.Account*), 49
- downloadAvatar() (метод *yandex_music.User*), 96
- downloadBackgroundImage() (метод *yandex_music.MixLink*), 91
- downloadCover() (метод *yandex_music.Album*), 64
- downloadCover() (метод *yandex_music.ShotData*), 123
- downloadCover() (метод *yandex_music.Track*), 131
- DownloadInfo (класс в *yandex_music*), 33
- downloadMds() (метод *yandex_music.ShotData*), 123
- downloadOgImage() (метод *yandex_music.Album*), 64
- downloadOgImage() (метод *yandex_music.Playlist*), 100
- downloadOgImage() (метод *yandex_music.Track*), 131
- downloadThumbnailImage() (метод *yandex_music.Artist*), 70
- duration (ампубум *yandex_music.Product*), 54
- ## E
- embed (ампубум *yandex_music.VideoSupplement*), 127
- embed_url (ампубум *yandex_music.VideoSupplement*), 127
- end (ампубум *yandex_music.Subscription*), 52
- energy (ампубум *yandex_music.Restrictions*), 113
- energy (ампубум *yandex_music.RotorSettings*), 103
- Enum (класс в *yandex_music*), 105
- error (ампубум *yandex_music.Album*), 61
- error (ампубум *yandex_music.Cover*), 32
- error (ампубум *yandex_music.utils.captcha_response.CaptchaResponse*), 133
- error (ампубум *yandex_music.utils.response.Response*), 132
- error_description (ампубум *yandex_music.utils.captcha_response.CaptchaResponse*), 133
- Event (класс в *yandex_music*), 80
- event_id (ампубум *yandex_music.ShotEvent*), 123
- exec_duration_millis (ампубум *yandex_music.InvocationInfo*), 37
- Experiments (класс в *yandex_music*), 35
- expires (ампубум *yandex_music.AutoRenewable*), 53
- ## F
- facebook_scrobbling_enabled (ампубум *yandex_music.UserSettings*), 58
- feature (ампубум *yandex_music.Product*), 55
- features (ампубум *yandex_music.Product*), 55
- Feed (класс в *yandex_music*), 75
- feed() (метод *yandex_music.Client*), 12
- feed_wizard_is_passed() (метод *yandex_music.Client*), 12
- feedWizardIsPassed() (метод *yandex_music.Client*), 12
- finished (ампубум *yandex_music.AutoRenewable*), 53
- first_name (ампубум *yandex_music.Account*), 47
- from_credentials() (метод класса *yandex_music.Client*), 13
- from_token() (метод класса *yandex_music.Client*), 13
- fromCredentials() (метод класса *yandex_music.Client*), 13
- fromToken() (метод класса *yandex_music.Client*), 13
- full_lyrics (ампубум *yandex_music.Lyrics*), 126

- full_name (*ampubym yandex_music.Account*), 47
 full_title (*ampubym yandex_music.Genre*), 83
 full_title (*ampubym yandex_music.Title*), 85
- ## G
- gain (*ampubym yandex_music.DownloadInfo*), 33
 generate_token_by_username_and_password() (*метод yandex_music.Client*), 14
 GeneratedPlaylist (*класс в yandex_music*), 77
 generateTokenByUsernameAndPassword() (*метод yandex_music.Client*), 13
 genitive (*ampubym yandex_music.CaseForms*), 101
 genre (*ampubym yandex_music.Album*), 61
 Genre (*класс в yandex_music*), 83
 genre_id (*ampubym yandex_music.AdParams*), 110
 genre_name (*ampubym yandex_music.AdParams*), 110
 genres() (*метод yandex_music.Client*), 14
 geocell_icon (*ampubym yandex_music.Station*), 108
 get_albums() (*метод yandex_music.Artist*), 70
 get_direct_link() (*метод yandex_music.DownloadInfo*), 35
 get_download_info() (*метод yandex_music.Track*), 131
 get_supplement() (*метод yandex_music.Track*), 131
 get_tracks() (*метод yandex_music.Artist*), 70
 getAlbums() (*метод yandex_music.Artist*), 70
 getDirectLink() (*метод yandex_music.DownloadInfo*), 35
 getDownloadInfo() (*метод yandex_music.Track*), 131
 getSupplement() (*метод yandex_music.Track*), 131
 getTracks() (*метод yandex_music.Artist*), 70
- ## H
- has_info_for_app_metrics (*ampubym yandex_music.Account*), 48
 has_plus (*ampubym yandex_music.Plus*), 50
 has_rights (*ampubym yandex_music.Lyrics*), 126
 hide_in_regions (*ampubym yandex_music.Genre*), 84
 hosted_user (*ampubym yandex_music.Account*), 47
 hostname (*ampubym yandex_music.InvocationInfo*), 37
 href (*ampubym yandex_music.Link*), 67
- ## I
- icon (*ampubym yandex_music.Station*), 108
 Icon (*класс в yandex_music*), 36
 id (*ampubym yandex_music.Album*), 61
 id (*ampubym yandex_music.Genre*), 83
 id (*ampubym yandex_music.Label*), 65
 id (*ampubym yandex_music.Lyrics*), 126
 id (*ampubym yandex_music.ShotType*), 124
 id (*ampubym yandex_music.Station*), 108
 id (*ampubym yandex_music.Supplement*), 125
 id (*ampubym yandex_music.TrackShort*), 41
 Id (*класс в yandex_music*), 106
 id_for_from (*ampubym yandex_music.Station*), 108
 image_url (*ampubym yandex_music.Icon*), 36
 images (*ampubym yandex_music.Genre*), 83
 Images (*класс в yandex_music*), 82
 in_app_products (*ampubym yandex_music.Settings*), 40
 instrumental (*ampubym yandex_music.CaseForms*), 101
 InvocationInfo (*класс в yandex_music*), 37
 is_mine (*ampubym yandex_music.Playlist*), 100
 is_tutorial_completed (*ampubym yandex_music.Plus*), 50
 isMine (*ampubym yandex_music.Playlist*), 100
 items_uri (*ampubym yandex_music.Cover*), 32
- ## K
- kind (*ampubym yandex_music.PlaylistAbsence*), 98
 kind (*ampubym yandex_music.PlaylistId*), 102
- ## L
- Label (*класс в yandex_music*), 65
 labels (*ampubym yandex_music.Album*), 61
 Landing (*класс в yandex_music*), 89
 landing() (*метод yandex_music.Client*), 14
 language (*ampubym yandex_music.Restrictions*), 113
 language (*ampubym yandex_music.RotorSettings*), 103
 last_fm_scrobbling_enabled (*ampubym yandex_music.UserSettings*), 58
 like() (*метод yandex_music.Album*), 64
 like() (*метод yandex_music.Artist*), 70
 like() (*метод yandex_music.Playlist*), 100
 like() (*метод yandex_music.Track*), 131
 Link (*класс в yandex_music*), 67
 logger (*ampubym yandex_music.Client*), 7
 login (*ampubym yandex_music.Account*), 47
 login (*ampubym yandex_music.User*), 95
 lyrics (*ampubym yandex_music.Lyrics*), 126
 lyrics (*ampubym yandex_music.Supplement*), 125
 Lyrics (*класс в yandex_music*), 126
- ## M
- MadeFor (*класс в yandex_music*), 96
 Major (*класс в yandex_music*), 129

- max (*ampubym yandex_music.DiscreteScale*), 104
mcdonalds (*ampubym yandex_music.Subscription*), 52
mds_url (*ampubym yandex_music.ShotData*), 122
me (*ampubym yandex_music.Client*), 7
media (*ampubym yandex_music.Vinyl*), 71
min (*ampubym yandex_music.DiscreteScale*), 104
misspell_corrected (*ampubym yandex_music.Search*), 117
MixLink (класс в *yandex_music*), 91
modified (*ampubym yandex_music.UserSettings*), 58
month (*ampubym yandex_music.Ratings*), 70
mood (*ampubym yandex_music.Restrictions*), 113
mood (*ampubym yandex_music.RotorSettings*), 103
mood_energy (*ampubym yandex_music.Restrictions*), 113
mood_energy (*ampubym yandex_music.RotorSettings*), 103
mts_icon (*ampubym yandex_music.Station*), 108
- ## N
- name (*ampubym yandex_music.DiscreteScale*), 104
name (*ampubym yandex_music.Enum*), 105
name (*ampubym yandex_music.Label*), 65
name (*ampubym yandex_music.Station*), 108
name (*ampubym yandex_music.User*), 95
name (*ampubym yandex_music.Value*), 112
native_products (*ampubym yandex_music.Settings*), 40
nocorrect (*ampubym yandex_music.Search*), 117
nominative (*ampubym yandex_music.CaseForms*), 101
Normalization (класс в *yandex_music*), 129
notify (*ampubym yandex_music.GeneratedPlaylist*), 77
now (*ampubym yandex_music.Account*), 47
- ## O
- oauth_url (*ampubym yandex_music.Client*), 7
og_image (*ampubym yandex_music.Album*), 62
order (*ampubym yandex_music.SearchResult*), 119
order (*ampubym yandex_music.Shot*), 121
order_id (*ampubym yandex_music.AutoRenewable*), 53
other_params (*ampubym yandex_music.AdParams*), 110
- ## P
- page (*ampubym yandex_music.Pager*), 38
page_ref (*ampubym yandex_music.AdParams*), 110
pager (*ampubym yandex_music.ArtistAlbums*), 75
pager (*ampubym yandex_music.ArtistTracks*), 74
Pager (класс в *yandex_music*), 37
parent_id (*ampubym yandex_music.Station*), 108
partner_id (*ampubym yandex_music.AdParams*), 110
passport_phones (*ampubym yandex_music.Account*), 48
PassportPhone (класс в *yandex_music*), 49
payment_method_types (*ampubym yandex_music.Product*), 55
per_page (*ampubym yandex_music.Pager*), 38
per_page (*ampubym yandex_music.SearchResult*), 119
permission_alerts() (*метод yandex_music.Client*), 15
PermissionAlerts (класс в *yandex_music*), 38
permissionAlerts() (*метод yandex_music.Client*), 15
permissions (*ampubym yandex_music.Status*), 45
Permissions (класс в *yandex_music*), 51
PersonalPlaylistsData (класс в *yandex_music*), 92
phone (*ampubym yandex_music.PassportPhone*), 49
picture (*ampubym yandex_music.Vinyl*), 71
play_audio() (*метод yandex_music.Client*), 15
playAudio() (*метод yandex_music.Client*), 15
PlayContext (класс в *yandex_music*), 90
PlayContextsData (класс в *yandex_music*), 87
PlayCounter (класс в *yandex_music*), 97
played (*ampubym yandex_music.Shot*), 121
Playlist (класс в *yandex_music*), 99
playlist_id (*ampubym yandex_music.Playlist*), 100
PlaylistAbsence (класс в *yandex_music*), 98
playlistId (*ampubym yandex_music.Playlist*), 100
PlaylistId (класс в *yandex_music*), 102
playlists (*ampubym yandex_music.Search*), 117
playlists_list() (*метод yandex_music.Client*), 16
playlistsList() (*метод yandex_music.Client*), 16
plus (*ampubym yandex_music.Status*), 45
Plus (класс в *yandex_music*), 50
possible_values (*ampubym yandex_music.Enum*), 105
prefix (*ampubym yandex_music.Cover*), 32
premium_region (*ampubym yandex_music.Status*), 46
prepositional (*ampubym yandex_music.CaseForms*), 101
preview (*ampubym yandex_music.DownloadInfo*), 33
price (*ampubym yandex_music.Product*), 55
price (*ampubym yandex_music.Vinyl*), 71
Price (класс в *yandex_music*), 57
product (*ampubym yandex_music.AutoRenewable*), 53

- Product (класс в *yandex_music*), 54
 product_id (ампубум *yandex_music.AutoRenewable*), 53
 product_id (ампубум *yandex_music.Product*), 54
 promo_codes_enabled (ампубум *yandex_music.Settings*), 40
 PromoCodeStatus (класс в *yandex_music*), 39
 promos_disabled (ампубум *yandex_music.UserSettings*), 59
 Promotion (класс в *yandex_music*), 93
 provider (ампубум *yandex_music.VideoSupplement*), 127
 provider_video_id (ампубум *yandex_music.VideoSupplement*), 127
 pumpkin (ампубум *yandex_music.Dashboard*), 111
- ## R
- radio_icon (ампубум *yandex_music.Genre*), 84
 radio_is_available (ампубум *yandex_music.Supplement*), 125
 Ratings (класс в *yandex_music*), 70
 rbt_disabled (ампубум *yandex_music.UserSettings*), 59
 ready (ампубум *yandex_music.GeneratedPlaylist*), 77
 reason (ампубум *yandex_music.PlaylistAbsence*), 98
 recent (ампубум *yandex_music.Album*), 62
 region (ампубум *yandex_music.Account*), 47
 registered_at (ампубум *yandex_music.Account*), 48
 release_date (ампубум *yandex_music.Album*), 62
 rename() (метод *yandex_music.Playlist*), 100
 req_id (ампубум *yandex_music.InvocationInfo*), 37
 request (ампубум *yandex_music.Client*), 16
 Request (класс в *yandex_music.utils.request*), 132
 Response (класс в *yandex_music.utils.response*), 132
 restrictions (ампубум *yandex_music.Station*), 108
 Restrictions (класс в *yandex_music*), 113
 restrictions2 (ампубум *yandex_music.Station*), 108
 result (ампубум *yandex_music.Best*), 120
 result (ампубум *yandex_music.utils.response.Response*), 132
 results (ампубум *yandex_music.SearchResult*), 119
 revision (ампубум *yandex_music.TracksList*), 43
 rotor_account_status() (метод *yandex_music.Client*), 17
 rotor_station_feedback() (метод *yandex_music.Client*), 17
 rotor_station_feedback_radio_started() (метод *yandex_music.Client*), 18
 rotor_station_feedback_skip() (метод *yandex_music.Client*), 18
 rotor_station_feedback_track_finished() (метод *yandex_music.Client*), 19
 rotor_station_feedback_track_started() (метод *yandex_music.Client*), 19
 rotor_station_info() (метод *yandex_music.Client*), 19
 rotor_station_settings2() (метод *yandex_music.Client*), 19
 rotor_station_tracks() (метод *yandex_music.Client*), 20
 rotor_stations_dashboard() (метод *yandex_music.Client*), 21
 rotor_stations_list() (метод *yandex_music.Client*), 21
 rotorAccountStatus() (метод *yandex_music.Client*), 16
 RotorSettings (класс в *yandex_music*), 103
 rotorStationFeedback() (метод *yandex_music.Client*), 16
 rotorStationFeedbackRadioStarted() (метод *yandex_music.Client*), 16
 rotorStationFeedbackSkip() (метод *yandex_music.Client*), 16
 rotorStationFeedbackTrackFinished() (метод *yandex_music.Client*), 16
 rotorStationFeedbackTrackStarted() (метод *yandex_music.Client*), 17
 rotorStationInfo() (метод *yandex_music.Client*), 17
 rotorStationsDashboard() (метод *yandex_music.Client*), 17
 rotorStationSettings2() (метод *yandex_music.Client*), 17
 rotorStationsList() (метод *yandex_music.Client*), 17
 rotorStationTracks() (метод *yandex_music.Client*), 17
- ## S
- Search (класс в *yandex_music*), 116
 search() (метод *yandex_music.Client*), 21
 search_request_id (ампубум *yandex_music.Search*), 117
 search_suggest() (метод *yandex_music.Client*), 22
 SearchResult (класс в *yandex_music*), 118
 searchSuggest() (метод *yandex_music.Client*), 22
 second_name (ампубум *yandex_music.Account*), 47
 Sequence (класс в *yandex_music*), 107

- service_available (*ampubym yandex_music.Account*), 47
 - Settings (*класс в yandex_music*), 40
 - settings() (*метод yandex_music.Client*), 22
 - sex (*ampubym yandex_music.User*), 95
 - Shot (*класс в yandex_music*), 120
 - shot_data (*ampubym yandex_music.Shot*), 121
 - shot_id (*ampubym yandex_music.Shot*), 121
 - shot_text (*ampubym yandex_music.ShotData*), 122
 - shot_type (*ampubym yandex_music.ShotData*), 122
 - ShotData (*класс в yandex_music*), 122
 - ShotEvent (*класс в yandex_music*), 123
 - shots (*ampubym yandex_music.ShotEvent*), 123
 - ShotType (*класс в yandex_music*), 124
 - show_disk_tracks_in_library (*ampubym yandex_music.UserSettings*), 59
 - show_in_menu (*ampubym yandex_music.Genre*), 83
 - show_translation (*ampubym yandex_music.Lyrics*), 126
 - shuffle_enabled (*ampubym yandex_music.UserSettings*), 58
 - skips_per_hour (*ampubym yandex_music.Status*), 45
 - social_network (*ampubym yandex_music.Link*), 68
 - Station (*класс в yandex_music*), 108
 - station_exists (*ampubym yandex_music.Status*), 45
 - StationResult (*класс в yandex_music*), 114
 - stations (*ampubym yandex_music.Dashboard*), 111
 - StationTracksResult (*класс в yandex_music*), 115
 - status (*ampubym yandex_music.PromoCodeStatus*), 39
 - status (*ampubym yandex_music.Shot*), 121
 - Status (*класс в yandex_music*), 45
 - status_desc (*ampubym yandex_music.PromoCodeStatus*), 39
 - sub_genres (*ampubym yandex_music.Genre*), 84
 - subeditor (*ampubym yandex_music.Status*), 45
 - subeditor_level (*ampubym yandex_music.Status*), 45
 - subscription (*ampubym yandex_music.Status*), 45
 - Subscription (*класс в yandex_music*), 51
 - suggestions (*ampubym yandex_music.Suggestions*), 116
 - Suggestions (*класс в yandex_music*), 116
 - Supplement (*класс в yandex_music*), 125
- T**
- tag (*ampubym yandex_music.Id*), 106
 - target_ref (*ampubym yandex_music.AdParams*), 110
 - text (*ampubym yandex_music.Best*), 120
 - text (*ampubym yandex_music.Description*), 72
 - text (*ampubym yandex_music.Search*), 117
 - text_language (*ampubym yandex_music.Lyrics*), 126
 - theme (*ampubym yandex_music.UserSettings*), 59
 - timestamp (*ampubym yandex_music.TrackShort*), 41
 - title (*ampubym yandex_music.Album*), 61
 - title (*ampubym yandex_music.Genre*), 83
 - title (*ampubym yandex_music.Link*), 67
 - title (*ampubym yandex_music.ShotType*), 124
 - title (*ampubym yandex_music.Title*), 85
 - title (*ampubym yandex_music.VideoSupplement*), 127
 - title (*ampubym yandex_music.Vinyl*), 71
 - Title (*класс в yandex_music*), 85
 - titles (*ampubym yandex_music.Genre*), 83
 - to_dict() (*метод yandex_music.YandexMusicObject*), 31
 - to_json() (*метод yandex_music.utils.difference.Difference*), 134
 - to_json() (*метод yandex_music.YandexMusicObject*), 31
 - token (*ampubym yandex_music.Client*), 7
 - total (*ampubym yandex_music.Pager*), 38
 - total (*ampubym yandex_music.SearchResult*), 118
 - track (*ampubym yandex_music.TrackShort*), 42
 - track (*ampubym yandex_music.TrackWithAds*), 76
 - Track (*класс в yandex_music*), 130
 - track_count (*ampubym yandex_music.Album*), 61
 - track_id (*ampubym yandex_music.Track*), 131
 - track_id (*ampubym yandex_music.TrackShort*), 42
 - track_position (*ampubym yandex_music.Album*), 62
 - track_supplement() (*метод yandex_music.Client*), 23
 - trackId (*ampubym yandex_music.Track*), 131
 - trackId (*ampubym yandex_music.TrackShort*), 42
 - TrackId (*класс в yandex_music*), 92
 - TrackPosition (*класс в yandex_music*), 65
 - tracks (*ampubym yandex_music.ArtistTracks*), 74
 - tracks (*ampubym yandex_music.Counts*), 66
 - tracks (*ampubym yandex_music.Search*), 117
 - tracks (*ampubym yandex_music.TracksList*), 43
 - tracks() (*метод yandex_music.Client*), 23
 - tracks_download_info() (*метод yandex_music.Client*), 23
 - tracks_ids (*ampubym yandex_music.TracksList*), 43
 - tracks_similar() (*метод yandex_music.Client*), 23
 - tracksDownloadInfo() (*метод yandex_music.Client*), 23
 - TrackShort (*класс в yandex_music*), 41
 - TrackShortOld (*класс в yandex_music*), 88
 - tracksIds (*ampubym yandex_music.TracksList*), 43

- TracksList (класс в *yandex_music*), 43
- tracksSimilar() (метод *yandex_music.Client*), 23
- trackSupplement() (метод *yandex_music.Client*), 22
- TrackWithAds (класс в *yandex_music*), 76
- trial_available (ампубум *yandex_music.Product*), 55
- trial_duration (ампубум *yandex_music.Product*), 55
- type (ампубум *yandex_music.Album*), 62
- type (ампубум *yandex_music.Cover*), 31
- type (ампубум *yandex_music.DiscreteScale*), 104
- type (ампубум *yandex_music.Enum*), 105
- type (ампубум *yandex_music.GeneratedPlaylist*), 77
- type (ампубум *yandex_music.Id*), 106
- type (ампубум *yandex_music.Product*), 54
- type (ампубум *yandex_music.SearchResult*), 118
- type (ампубум *yandex_music.TrackWithAds*), 76
- type_ (ампубум *yandex_music.Best*), 120
- type_ (ампубум *yandex_music.Link*), 67
- ## U
- uid (ампубум *yandex_music.Account*), 47
- uid (ампубум *yandex_music.PlaylistId*), 102
- uid (ампубум *yandex_music.TracksList*), 43
- uid (ампубум *yandex_music.User*), 95
- uid (ампубум *yandex_music.UserSettings*), 58
- until (ампубум *yandex_music.Permissions*), 51
- updated (ампубум *yandex_music.PlayCounter*), 97
- uri (ампубум *yandex_music.Cover*), 32
- url (ампубум *yandex_music.Description*), 73
- url (ампубум *yandex_music.VideoSupplement*), 127
- url (ампубум *yandex_music.Vinyl*), 71
- url_part (ампубум *yandex_music.Genre*), 83
- User (класс в *yandex_music*), 95
- user_info (ампубум *yandex_music.MadeFor*), 96
- user_music_visibility (ампубум *yandex_music.UserSettings*), 58
- user_social_visibility (ампубум *yandex_music.UserSettings*), 59
- users_dislikes_tracks() (метод *yandex_music.Client*), 26
- users_dislikes_tracks_add() (метод *yandex_music.Client*), 26
- users_dislikes_tracks_remove() (метод *yandex_music.Client*), 26
- users_likes_albums() (метод *yandex_music.Client*), 26
- users_likes_albums_add() (метод *yandex_music.Client*), 26
- users_likes_albums_remove() (метод *yandex_music.Client*), 26
- users_likes_artists() (метод *yandex_music.Client*), 26
- users_likes_artists_add() (метод *yandex_music.Client*), 26
- users_likes_artists_remove() (метод *yandex_music.Client*), 26
- users_likes_playlists() (метод *yandex_music.Client*), 26
- users_likes_playlists_add() (метод *yandex_music.Client*), 26
- users_likes_playlists_remove() (метод *yandex_music.Client*), 26
- users_likes_tracks() (метод *yandex_music.Client*), 26
- users_likes_tracks_add() (метод *yandex_music.Client*), 26
- users_likes_tracks_remove() (метод *yandex_music.Client*), 26
- users_playlists() (метод *yandex_music.Client*), 26
- users_playlists_change() (метод *yandex_music.Client*), 27
- users_playlists_create() (метод *yandex_music.Client*), 27
- users_playlists_delete() (метод *yandex_music.Client*), 28
- users_playlists_delete_track() (метод *yandex_music.Client*), 28
- users_playlists_insert_track() (метод *yandex_music.Client*), 29
- users_playlists_list() (метод *yandex_music.Client*), 29
- users_playlists_name() (метод *yandex_music.Client*), 29
- users_playlists_visibility() (метод *yandex_music.Client*), 30
- users_settings() (метод *yandex_music.Client*), 30
- usersDislikesTracks() (метод *yandex_music.Client*), 24
- usersDislikesTracksAdd() (метод *yandex_music.Client*), 24
- usersDislikesTracksRemove() (метод *yandex_music.Client*), 24
- UserSettings (класс в *yandex_music*), 58
- usersLikesAlbums() (метод *yandex_music.Client*), 24
- usersLikesAlbumsAdd() (метод *yandex_music.Client*), 24
- usersLikesAlbumsRemove() (метод *yandex_music.Client*), 24
- usersLikesArtists() (метод *yandex_music.Client*), 24
- usersLikesArtistsAdd() (метод *yandex_music.Client*), 24

yandex_music.Client), 24
 usersLikesArtistsRemove() (*method* *yandex_music.Client*), 24
 usersLikesPlaylists() (*method* *yandex_music.Client*), 24
 usersLikesPlaylistsAdd() (*method* *yandex_music.Client*), 24
 usersLikesPlaylistsRemove() (*method* *yandex_music.Client*), 25
 usersLikesTracks() (*method* *yandex_music.Client*), 25
 usersLikesTracksAdd() (*method* *yandex_music.Client*), 25
 usersLikesTracksRemove() (*method* *yandex_music.Client*), 25
 usersPlaylists() (*method* *yandex_music.Client*), 25
 usersPlaylistsChange() (*method* *yandex_music.Client*), 25
 usersPlaylistsCreate() (*method* *yandex_music.Client*), 25
 usersPlaylistsDelete() (*method* *yandex_music.Client*), 25
 usersPlaylistsDeleteTrack() (*method* *yandex_music.Client*), 25
 usersPlaylistsInsertTrack() (*method* *yandex_music.Client*), 25
 usersPlaylistsList() (*method* *yandex_music.Client*), 25
 usersPlaylistsName() (*method* *yandex_music.Client*), 25
 usersPlaylistsVisibility() (*method* *yandex_music.Client*), 26

V

value (*ампубум* *yandex_music.PlayCounter*), 97
 value (*ампубум* *yandex_music.Value*), 112
 Value (*класс* в *yandex_music*), 112
 values (*ампубум* *yandex_music.Permissions*), 51
 vendor (*ампубум* *yandex_music.AutoRenewable*), 53
 vendor_help_url (*ампубум* *yandex_music.AutoRenewable*), 53
 vendor_trial_available (*ампубум* *yandex_music.Product*), 55
 verified (*ампубум* *yandex_music.User*), 95
 version (*ампубум* *yandex_music.Album*), 61
 version (*ампубум* *yandex_music.Cover*), 32
 very_important (*ампубум* *yandex_music.Album*), 62
 Video (*класс* в *yandex_music*), 44
 videos (*ампубум* *yandex_music.Search*), 117
 videos (*ампубум* *yandex_music.Supplement*), 125
 VideoSupplement (*класс* в *yandex_music*), 127

Vinyl (*класс* в *yandex_music*), 71
 volume_percents (*ампубум* *yandex_music.UserSettings*), 58
 volumes (*ампубум* *yandex_music.Album*), 62

W

web_payment_month_product_price (*ампубум* *yandex_music.Settings*), 40
 web_payment_url (*ампубум* *yandex_music.Settings*), 40
 week (*ампубум* *yandex_music.Ratings*), 70
 weight (*ампубум* *yandex_music.Genre*), 83
 with_tracks() (*method* *yandex_music.Album*), 64
 withTracks() (*method* *yandex_music.Album*), 64

X

x_captcha_key (*ампубум* *yandex_music.utils.captcha_response.CaptchaResponse*), 133
 x_captcha_url (*ампубум* *yandex_music.utils.captcha_response.CaptchaResponse*), 133

Y

YandexMusicObject (*класс* в *yandex_music*), 31
 year (*ампубум* *yandex_music.Album*), 62
 year (*ампубум* *yandex_music.Vinyl*), 71