pyQiwi Documentation

Выпуск 2.1.6

Levent Duivel

апр. 27, 2022

Контент:

Глава 1

pyQiwi

Python Qiwi API Wrapper

- Лицензия: MIT
- Документация: https://pyqiwi.readthedocs.io.

1.1 Возможности

- Оплата любых услуг
- Переводы на любой Qiwi Кошелек
- Статистика по платежам
- История о сделанных платежах в любой промежуток времени
- Прохождение упрощенной идентификации
- Определение провайдера мобильного телефона
- Получение текущего курса валют

1.2 Установка

```
$ pip install qiwipy
```

1.3 Использование

```
import pyqiwi
wallet = pyqiwi.Wallet(token='', number='79001234567')
```

1.4 Быстрый туториал

1.4.1 Получить текущий баланс

```
print(wallet.balance())
```

1.4.2 Отправка платежа

```
payment = wallet.send(pid=99, recipient='79001234567', amount=1.11, comment='Привет!')
example = 'Payment is {0}\nRecipient: {1}\nPayment Sum: {2}'.format(
          payment.transaction['state']['code'], payment.fields['account'], payment.sum)
print(example)
```

1.4.3 Получить комиссию для платежа

```
commission = wallet.commission(pid=99, recipient='79001234567', amount=1.11)
print(commission.qw_commission.amount)
```

Для более подробных инструкций, посетите документацию.

Глава 2

Установка

Поддерживаемые версии Python: *3.4* и выше

2.1 Стабильный релиз

Для установки pyQiwi, запустите эту команду в вашем терминале:

```
$ pip install qiwipy
```

Это предпочтительный метод установки, так как он всегда будет устанавливать самый последний стабильный релиз. Если у вас нет установленного pip, этот Справочник по установке Python может помочь в процессе.

2.2 Из исходных файлов

Исходные файлы для pyQiwi могут быть загружены с GitHub репозитория.

Вы можете либо клонировать публичный репозиторий:

```
$ git clone git://github.com/mostm/pyqiwi
```

Или загрузить tarball:

```
$ curl  -OL https://github.com/mostm/pyqiwi/tarball/master
```

Как только вы получите копию исходных файлов, вы можете установить их с помощью:

```
$ python setup.py install
```

Глава 3

Использование

Для того чтобы использовать pyQiwi в проекте:

```
import pyqiwi
```

Глава 4

Участие в развитии

Вклады в развитие библиотеки приветствуются, и они высоко ценятся!

Вы можете внести свой вклад во несколькими вариантами:

4.1 Типы вкладов

4.1.1 Сообщайте об ошибках

Сообщайте об ошибках на https://github.com/mostm/pyqiwi/issues.

Если вы сообщаете об баге, пожалуйста добавьте:

- Ваша операционная система, и версия библиотеки.
- Любые детали об вашей установке могли бы помочь в устранении неисправности.
- Подробные шаги для воспроизведения ошибки.

4.1.2 Исправляйте ошибки

Посмотрите через GitHub Issues об ошибках. Все что помечено «bug» или «help wanted» открыто для тех, кто хочет его реализовать.

4.1.3 Реализуйте новые фичи

Просмотрите через GitHub Issues об фичах. Что-нибудь с тегами «enhancement» и «help wanted» открыто для тех, кто хочет её реализовать.

4.1.4 Напишите документацию

pyQiwi всегда может использовать больше документации, будь то как часть официальных документациях, в докстрингах, или даже в блог постах, статях и тому подобном.

4.1.5 Отправить Отзыв

Лучший способ отправить отзыв-отправить issue на https://github.com/mostm/pyqiwi/issues.

Если вы предлагаете фичу:

- Подробно объясните, как это будет работать.
- Держите область как можно более узкой, чтобы упростить ее реализацию.
- Помните, что это добровольный проект, и что вклады приветствуются :)

4.2 Начните!

Готовы внести свой вклад? Вот как настроить *pyQiwi* для разработки.

1. Сделайте форк репозитория *pyqiwi* на GitHub.
2. Клонируйте свой форк локально:

```
$ git clone git@github.com:ваше_имя_здесь/pyqiwi.git
```

3. Установите вашу локальную копию в virtualenv. Предпологая то что вы уже установили virtualenvwrapper, это как вы настраиваете свой форк для локальной разработки:

```
$ mkvirtualenv pyqiwi
$ cd pyqiwi/
$ python setup.py develop
```

4. Создайте ветку для локальной разработки:

```
$ git checkout -b имя-вашего-фикса-или-фичи
```

   Теперь вы можете внести свое изменение.

5. Как только вы закончили делать изменения, проверьте то что ваши изменения проходят flake8 и тесты, включая тестируя несколько других Python версий с tox:

```
$ flake8 pyqiwi tests
$ python setup.py test or py.test
$ tox
```

   Для того чтобы получить flake8 и tox, просто pip install их в ваш virtualenv.

6. Сделайте коммит ваших изменений и отправьте вашу ветку на GitHub:

```
$ git add .
$ git commit -m "Подробное описание изменений."
$ git push origin имя-вашего-фикса-или-фичи
```

7. Отправьте пулл реквест используя сайт GitHub.

4.3 Рекомендации по пулл реквестам

Перед тем как отправить пулл реквест, проверьте то что он отвечает этим требованиям:

1. Пулл реквест должен добавлять тесты.
2. Если пулл реквест добавляет функциональность, документация должна быть обновлена. Добавьте вашу новую функциональность в функцию с докстрингом, и добавьте вашу фичу в список в README.rst.
3. Пулл реквест должен работать с Python 3.4, 3.5 и 3.6. Проверьте https://travis-ci.org/mostm/pyqiwi/pull_requests и будьте уверены в том что все тесты прошли успешно на всех поддерживаемых Python версиях.

4.4 Подсказка

Для того чтобы запустить тесты:

```
$ py.test tests.test_pyqiwi
```

4.5 Развертывание

Напоминание разработчикам о том, как развернуть. Убедитесь, что все изменения закоммитчены (включая запись в HISTORY.rst). Затем запустите:

```
$ bumpversion patch # возможные: major / minor / patch
$ git push
$ git push --tags
```

Travis CI затем отправит все это на PyPI, если тесты прошли успешно.

Глава 5

Титры

5.1 Ведущий разработчик

- Levent Duivel <mostm@endcape.ru>

5.2 Вкладчики

Пока никого. Почему бы не быть первым?

Глава 6

История изменений

6.1 2.1 (6.05.2018)

- *Wallet.balance* теперь имеет базовое значение *currency* 643 (Российский рубль)
- Новый метод для идентификации кошельков: *Wallet.identification*
- Весь блок методов к истории платежей был обновлен до API v2
- Для получения квитанции по платежу был добавлен метод *Wallet.cheque*
- **Если у вас по какой-то причине нет «балансов» на аккаунте, вы можете их создать при помощи V**
  Запрос доступных счетов, доступных для создания реализован в *Wallet.offered_ accounts*
- Создание ссылки для автозаполненных платежных форм достуно в *pyqiwi.generate_ form_ link*
- Вы хотите определить ID провайдера для пополнения мобильного телефона? Используйте *pyqiwi.detect_ mobile*

6.2 2.0.8 (29.04.2018)

- У нас появились тесты!
- Небольшие исправления

6.3 2.0.7 (14.04.2018)

- Небольшие исправления

6.4 2.0.6 (9.03.2018)

- Небольшие исправления

6.5 2.0.5 (6.11.2017)

- Логгер был перенесен из *pyqiwi.logger* в *pyqiwi.apihelper.logger*
- Появился метод в *Wallet transaction* для получения определенной транзакции по ID и её типу.
- Вместе с этим и появился для него собственный тип: *pyqiwi.types.Transaction*
- Небольшие исправления в документации
- Небольшие исправления

6.6 2.0.4 (6.11.2017)

- Небольшие исправления

6.7 2.0.3 (6.11.2017)

- Небольшие исправления

6.8 2.0.2 (29.10.2017)

- Небольшие исправления

6.9 2.0.1 (29.10.2017)

- У нас появилась документация на ReadTheDocs!
- Документация в коде была широко дополнена
- Были внесены изменения в вид документации в коде

6.10 2.0 (28.10.2017)

Первое большое изменение библиотеки!

- Интересны логи? У нас появился логгер *pyqiwi* в *pyqiwi.logger*
- Хотите посмотреть счета в Qiwi? *Wallet.accounts*
- Теперь выдается не *dict*-обьекты, а что-либо из *pyqiwi/types.py*
- Все вызовы к API были перенесены в *pyqiwi/apihelper.py*
- Если у вас возникла ошибка в запросе к API, вы получите исключение из *pyqiwi/exceptions.py*

6.11 1.2.1 (24.10.2017)

- Релиз на PyPI.

6.12 1.2 (24.10.2017)

- Переименование класса *Person* в *Wallet*
- Методы класса *Payment* теперь в *Wallet*
- Класс *Payment* удален
- Нет необходимости в config'е, теперь нужно передать токен в *Wallet()* [Возможность мульти-аккаунта]
- Вместе с этим, нужно передавать токен в *get_ commission* (но эта же функция находится в *Wallet* с подготовленным токеном)
- Методы *Wallet.history()* и *Wallet.stat()* требуют *datetime.datetime*, вместо *str*
- Любые обращения к f-строкам, были заменены на метод *str.format*

6.13 1.1 (5.09.2017)

- Небольшие улучшения

6.14 1.0 (5.09.2017)

- Первый релиз!

Глава 7

API

7.1 pyqiwi

Python Qiwi API Wrapper 2.1 by mostm

See pyQiwi Documentation: pyqiwi.readthedocs.io

`class pyqiwi.Wallet(`*token*, *number=None*, *contract_info=True*, *auth_info=True*, *user_info=True*`)`

Visa QIWI Кошелек

Параметры

- `token` (*str*) – Ключ Qiwi API пользователя.
- `number` (*Optional[str]*) – Номер для указанного кошелька. По умолчанию - `None`. Если не указан, статистика и история работать не будет.
- `contract_info` (*Optional[bool]*) – Логический признак выгрузки данных о кошельке пользователя. По умолчанию - `True`.
- `auth_info` (*Optional[bool]*) – Логический признак выгрузки настроек авторизации пользователя. По умолчанию - `True`.
- `user_info` (*Optional[bool]*) – Логический признак выгрузки прочих пользовательских данных. По умолчанию - `True`.

`accounts`

Все доступные счета на кошельке. Использовать можно только рублевый Visa QIWI Wallet.

Type iterable of *Account*

`profile`

Профиль пользователя.

Type *Profile*

`offered_accounts`

Доступные счета для создания

Type iterable of *Account*

balance(*currency=643*)
Баланс Visa QIWI Кошелька

Параметры currency (*int*) – ID валюты в number-3 ISO-4217. Например, 643 для российского рубля.

Результат Баланс кошелька.

Тип результата float

Raises ValueError – Во всех добавленных вариантах оплаты с указанного Qiwi-кошелька нет информации об балансе и его сумме. Скорее всего это временная ошибка Qiwi API, и вам стоит попробовать позже. Так же, эта ошибка может быть вызвана только-что зарегистрированным Qiwi-кошельком,

либо довольно старым Qiwi-кошельком, которому необходимо изменение пароля.

cheque(*txn_id*, *txn_type*, *file_format='PDF'*, *email=None*)
Получение чека по транзакции, на E-Mail или файл.

Параметры

- txn_id (*int*) – ID транзакции
- txn_type (*str*) – Тип указанной транзакции
- file_format (*str*) – Формат файла(игнорируется при использовании email)
- email (*str*) – E-Mail, куда отправить чек, если это необходимо.

Результат ??? | Прямой возврат ответа от Qiwi API

Тип результата binary

commission(*pid*, *recipient*, *amount*)
Расчет комиссии для платежа

Параметры

- pid (*str*) – Идентификатор провайдера.
- recipient (*str*) – Номер телефона (с международным префиксом) или номер карты/счета получателя. В зависимости от провайдера.
- amount (*float/int*) – Сумма платежа. Положительное число, округленное до 2 знаков после десятичной точки. При большем числе знаков значение будет округлено до копеек в меньшую сторону.

Результат Комиссия для платежа

Тип результата *OnlineCommission*

create_account(*account_alias*)
Создание счета-баланса в Visa QIWI Wallet

Параметры account_alias (*str*) – Псевдоним нового счета. Один из доступных в Wallet.offered_accounts.

Результат Был ли успешно создан счет?

Тип результата bool

cross_rates
Курсы валют QIWI Кошелька

Результат Состоит из: *Rate* - Курса.

Тип результата list

history(*rows=20*, *operation=None*, *start_date=None*, *end_date=None*, *sources=None*, *next_txn_date=None*, *next_txn_id=None*)

История платежей

Предупреждение:

Максимальная интенсивность запросов истории платежей - не более 100 запросов в минуту для одного и того же номера кошелька.

При превышении доступ к API блокируется на 5 минут.

Параметры

- **rows** (*Optional[int]*) – Число платежей в ответе, для разбивки отчета на части. От 1 до 50, по умолчанию 20.
- **operation** (*Optional[str]*) – Тип операций в отчете, для отбора. Варианты: ALL, IN, OUT, QIWI_CARD. По умолчанию - ALL.
- **start_date** (*Optional[datetime.datetime]*) – Начальная дата поиска платежей.
- **end_date** (*Optional[datetime.datetime]*) – Конечная дата поиска платежей.
- **sources** (*Optional[list]*) – Источники платежа, для отбора. Варианты: QW_RUB, QW_USD, QW_EUR, CARD, MK. По умолчанию - все указанные.
- **next_txn_date** (*Optional[datetime.datetime]*) – Дата транзакции для отсчета от предыдущего списка (равна параметру nextTxnDate в предыдущем списке).
- **next_txn_id** (*Optional[int]*) – Номер предшествующей транзакции для отсчета от предыдущего списка (равен параметру nextTxnId в предыдущем списке).

Примечание: Если вы хотите использовать start_date или end_date, вы должны указать оба параметра. Такое же использование и у next_txn_date и next_txn_id. Максимальный допустимый интервал между start_date и end_date - 90 календарных дней.

Результат Состоит из: transactions[list[*Transaction*]] - Транзакции. next_txn_date[datetime.datetime] - Дата транзакции(для использования в следующем использовании). next_txn_id[int] - Номер транзакции.

Тип результата dict

identification(*birth_date*, *first_name*, *middle_name*, *last_name*, *passport*, *inn=None*, *snils=None*, *oms=None*)

Идентификация пользователя

Данный запрос позволяет отправить данные для упрощенной идентификации своего QIWI кошелька.

Предупреждение: Данный метод не тестируется, соответственно я не могу гарантировать того что он будет работать как должен. Вы делаете это на свой страх и риск.

Параметры

- `birth_date` (*str*) – Дата рождения пользователя (в формате "ГГГГ-ММ-ДД")
- `first_name` (*str*) – Имя пользователя
- `middle_name` (*str*) – Отчество пользователя
- `last_name` (*str*) – Фамилия пользователя
- `passport` (*str*) – Серия и номер паспорта пользователя (только цифры)
- `inn` (*str*) – ИНН пользователя
- `snils` (*str*) – Номер СНИЛС пользователя
- `oms` (*str*) – Номер полиса ОМС пользователя

Результат Текущая идентификация пользователя. Параметр внутри отвечающий за подтверждение успешной идентификации: Identity.check

Тип результата *Identity*

`mobile`(*account*, *amount*)

Оплата мобильной связи.

Параметры

- `account` (*str*) – Номер мобильного телефона (с кодом страны, 7/8, без +)
- `amount` (*float*) – Сумма платежа

Результат Платеж

Тип результата *Payment*

Raises `ValueError` – В случае, если не удалось определить провайдера.

`qiwi_transfer`(*account*, *amount*, *comment=None*)

Перевод на Qiwi Кошелек

Параметры

- `account` (*str*) – Номер Qiwi Кошелька
- `amount` (*float*) – Сумма перевода
- `comment` (*str*) – Комментарий

Результат Платеж

Тип результата *Payment*

`send`(*pid*, *recipient*, *amount*, *comment=None*, *fields=None*)

Отправить платеж

Параметры

- `pid` (*str*) – Идентификатор провайдера.
- `recipient` (*str*) – Номер телефона (с международным префиксом) или номер карты/счета получателя. В зависимости от провайдера.

- `amount` (*float/int*) – Сумма платежа. Положительное число, округленное до 2 знаков после десятичной точки. При большем числе знаков значение будет округлено до копеек в меньшую сторону.
- `comment` (*Optional[str]*) – Комментарий к платежу.
- `fields` (*dict*) – Ручное добавление dict'а в платежи. Требуется для специфичных платежей. Например, перевод на счет в банке.

Результат Платеж

Тип результата *Payment*

`stat`(*start_date=None*, *end_date=None*, *operation=None*, *sources=None*)
Статистика платежей

Примечание: Изначально берется статистика с начала месяца

Параметры

- `operation` (*Optional[str]*) – Тип операций в отчете, для отбора. Варианты: ALL, IN, OUT, QIWI_CARD. По умолчанию - ALL.
- `start_date` (*Optional[datetime.datetime]*) – Начальная дата поиска платежей.
- `end_date` (*Optional[datetime.datetime]*) – Конечная дата поиска платежей.
- `sources` (*Optional[list]*) – Источники платежа, для отбора. Варианты: QW_RUB, QW_USD, QW_EUR, CARD, MK. По умолчанию - все указанные.

Результат Статистика

Тип результата *Statistics*

`transaction`(*txn_id*, *txn_type*)
Получение транзакции из Qiwi API

Параметры

- `txn_id` (*str*) – ID транзакции.
- `txn_type` (*str*) – Тип транзакции (IN/OUT/QIWI_CARD).

Результат Транзакция

Тип результата *Transaction*

`pyqiwi.detect_mobile`(*phone*)
Определение провайдера мобильного телефона

Параметры `phone` (*str*) – Номер телефона

Результат ID провайдера

Тип результата str

`pyqiwi.generate_form_link`(*pid*, *account*, *amount*, *comment*, *blocked=None*, *account_type=None*)
Создание автозаполненной платежной формы

Параметры

- `pid` (*str*) – ID провайдера
- `account` (*str*) – Счет получателя
- `amount` (*float*) – Сумма платежа
- `comment` (*str*) – Комментарий
- `blocked` (*list[str]*) – Список из значений «заблокированных» (не изменяемых на веб-странице) полей внутри ссылки. Варианты: sum, account, comment
- `account_type` (*int or str*) – Отвечает за вариант перевода при pid=99999 (вариация перевода на Qiwi Кошелек) Варианты: 0 (перевод по номеру телефона, phone), 1 (перевод по «никнейму», nickname),

  str (сами впишите вариант по соответствию с Qiwi API)

Примечание: Комментарий применяется только при переводе на Qiwi Кошелек по номеру (pid==99) Сумма платежа не может быть более 99999 из-за ограничений на один платеж. Тип счета для перевода на Qiwi Кошелек (pid=99999) с возможностью ввода «nickname» выбирается в account_type

---

Результат Ссылка

Тип результата str

Raises `ValueError` – amount>99999 или список blocked неверен

`pyqiwi.get_commission(`*token*, *pid*`)`

Получение стандартной комиссии

Параметры

- `token` (*str*) – Ключ Qiwi API
- `pid` (*str*) – Идентификатор провайдера.

Результат Комиссия для платежа

Тип результата *Commission*

7.2 Types

`class pyqiwi.types.Account(`*alias*, *fs_alias*, *title*, *has_balance*, *currency*, *_type*, *balance*, *obj*`)`

Счет из Visa QIWI Кошелька

`alias`

Псевдоним пользовательского баланса

Type str

`fs_alias`

Псевдоним банковского баланса

Type str

`title`

Название соответствующего счета кошелька

Type str

`has_balance`
Логический признак реального баланса в системе QIWI Кошелек (не привязанная карта, не счет мобильного телефона и т.д.)

Type str

`currency`
Код валюты баланса (number-3 ISO-4217).

Type int

`type`
Сведения о счете

Type *AccountType*

`balance`
Псевдоним пользовательского баланса

Type Optional[float]

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'a или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'a или строки

`class pyqiwi.types.AccountType(_id, title, obj)`
Сведения о счете из Visa QIWI Кошелька

`id`
Кодовое название счета

Type str

`title`
Название счета

Type str

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'a или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'a или строки

`class pyqiwi.types.AuthInfo(bound_email, ip, last_login_date, mobile_pin_info, pass_info, person_id, pin_info, registration_date, obj)`
Профиль пользователя Visa QIWI Кошелька

`bound_email`
E-mail, привязанный к кошельку. Если отсутствует, то `None`

Type str/None

`ip`
IP-адрес последней пользовательской сессии

Type str

`last_login_date`
Дата/время последней сессии в QIWI Кошельке

Type str

`mobile_pin_info`
Данные о PIN-коде мобильного приложения QIWI Кошелька

Type *MobilePinInfo*

`pass_info`
Данные о пароле к сайту qiwi.com

Type *PassInfo*

`person_id`
Номер кошелька пользователя

Type int

`pin_info`
Данные о PIN-коде к приложению QIWI Кошелька на QIWI терминалах

Type *PinInfo*

`registration_date`
Дата/время регистрации QIWI Кошелька пользователя (через сайт либо мобильное приложение, либо другим способом)

Type datetime.datetime

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'a или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'a или строки

`class pyqiwi.types.Commission(ranges, obj)`
Стандартная комиссия

`ranges`
Массив объектов с граничными условиями комиссий

Type list[*CommissionRange*]

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'a или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'a или строки

`class pyqiwi.types.CommissionRange(bound, fixed, rate, _min, _max, obj)`
Условия комиссии

`bound`
Сумма платежа, начиная с которой применяется условие

Type Optional[float/int]

`rate`
Комиссия (абсолютный множитель)

Type Optional[float/int]

`min`
Минимальная сумма комиссии

Type Optional[float/int]

`max`
Максимальная сумма комиссии

Type Optional[float/int]

`fixed`
Фиксированная сумма комиссии

Type Optional[float/int]

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.ContractInfo(blocked, contract_id, creation_date, features, identification_info, obj)`
Информация о кошельке пользователя

`blocked`
Логический признак блокировки кошелька

Type bool

`contract_id`
Номер кошелька пользователя

Type int

`creation_date`
Дата/время создания QIWI Кошелька пользователя (через сайт либо мобильное приложение, либо при первом пополнении, либо другим способом)

Type datetime.datetime

`features`
Служебная информация

Type ???

`identification_info`
Данные об идентификации пользователя

Type list[*IdentificationInfo*]

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.IdentificationInfo(bank_alias, identification_level, obj)`
Данные об идентификации пользователя

`bank_alias`
Акроним системы, в которой пользователь получил идентификацию: QIWI - QIWI Кошелек.

Type str

`identification_level`
Текущий уровень идентификации кошелька Возможные значения: ANONYMOUS - без идентификации SIMPLE, VERIFIED - упрощенная идентификация FULL - полная идентификация

Type str

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.Identity(_id, _type, birth_date, first_name, middle_name, last_name, passport, inn, snils, oms, base_inn, obj)`
Идентификация

`id`
Номер кошелька пользователя

Type int

`type`
Текущий уровень идентификации кошелька: SIMPLE - без идентификации. VERIFIED - упрощенная идентификация (данные для идентификации успешно прошли проверку). FULL – если кошелек уже ранее получал полную идентификацию по данным ФИО, номеру паспорта и дате рождения.

Type str

`birth_date`
Дата рождения пользователя

Type str

`first_name`
Имя пользователя

Type str

`middle_name`
Отчество пользователя

Type str

`last_name`
Фамилия пользователя

Type str

`passport`
Серия и номер паспорта пользователя

Type str

`inn`
ИНН пользователя

Type str

`snils`
Номер СНИЛС пользователя

Type str

`oms`
Номер полиса ОМС пользователя

Type str

`check`
Идентификация кошелька выполнена? (Используются варианты предлагаемые документацией Qiwi API)

Type bool

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.JsonDeserializable`
Субклассы этого класса гарантированно могут быть созданы из json-подобного dict'а или форматированной json строки Все субклассы этого класса должны перезаписывать de_json

`raw`
Содержит в себе исходные данные от Qiwi API

Type ???

`static check_json(json_type)`
Проверяет, json_type или dict или str. Если это dict, возвращает его в исходном виде Иначе, возвращает dict созданный из json.loads(json_type)

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`static decode_date(date_string: str)`
Декодирует дату из строки вида отправляемого Qiwi API ISO-8601

Результат

Тип результата datetime.datetime данной строки

`class pyqiwi.types.MobilePinInfo(mobile_pin_used, last_mobile_pin_change, next_mobile_pin_change, obj)`
Данные о PIN-коде мобильного приложения QIWI Кошелька

`mobile_pin_used`
Логический признак использования PIN-кода (фактически означает, что мобильное приложение используется)

Type bool

`last_mobile_pin_change`
Дата/время последнего изменения PIN-кода мобильного приложения QIWI Кошелька

Type datetime.datetime

`next_mobile_pin_change`
Дата/время следующего (планового) изменения PIN-кода мобильного приложения QIWI Кошелька

Type datetime.datetime

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.OnlineCommission(provider_id, withdraw_sum, enrollment_sum, qw_commission, funding_source_commission, withdraw_to_enrollment_rate, obj)`
Подсчитанная комиссия

`provider_id`
Идентификатор провайдера

Type int

`withdraw_sum`
Общая сумма платежа

Type *TransactionSum*

`enrollment_sum`
Сумма платежа с учетом комиссии

Type *TransactionSum*

`qw_commission`
Комиссия Qiwi

Type *TransactionSum*

`funding_source_commission`
Комиссия платежной системы(если Qiwi, то всегда 0)

Type *TransactionSum*

`withdraw_to_enrollment_rate`
???

Type float/int

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.PassInfo(last_pass_change, next_pass_change, password_used, obj)`
Данные о пароле к сайту qiwi.com

`last_pass_change`
Дата/время последнего изменения пароля сайта qiwi.com

Type str

`next_pass_change`
Дата/время следующего (планового) изменения пароля сайта qiwi.com

Type str

`password_used`
Логический признак использования пароля (фактически означает, что пользователь заходит на сайт)

Type bool

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.Payment(_id, terms, fields, _sum, transaction, source, comment, obj)`
Данные о принятом платеже

`id`
Клиентский ID транзакции (В этой библиотеке, он считается 1000*Unix timestamp)

Type str

`terms`
Идентификатор провайдера

Type str

`fields`
Реквизиты платежа

Type *PaymentFields*

`sum`
Данные о сумме платежа

Type *TransactionSum*

`source`
???

Type str

`comment`
Комментарий к платежу

Type Optional[str]

`transaction`
Данные о транзакции в процессинге

Type *Payment.Transaction*

`class Transaction(_id, state, obj)`
Данные о транзакции в процессинге

`id`
ID транзакции
Type str

`state`
Статус транзакции(в момент написания, только Accepted)

Type str

classmethod de_json(*json_type*)
Возвращает инстанс этого класса из созданного json dict'a или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'a или строки

classmethod de_json(*json_type*)
Возвращает инстанс этого класса из созданного json dict'a или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'a или строки

class pyqiwi.types.PaymentFields
Реквизиты платежа

Данный класс представляет из себя хаотичную структуру(но всегда присутствует «account») Судя по документации Qiwi API, создается из исходного поля fields для платежа

Примечание: Если вы хотите посмотреть исходный вид выданный Qiwi API, используйте str(PaymentFields)

---

account
Получатель платежа

Type str

classmethod de_json(*json_type*)
Возвращает инстанс этого класса из созданного json dict'a или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'a или строки

class pyqiwi.types.PinInfo(*pin_used*, *obj*)
Данные о PIN-коде к приложению QIWI Кошелька на QIWI терминалах

pin_used
Логический признак использования PIN-кода (фактически означает, что пользователь заходил в приложение)

Type bool

classmethod de_json(*json_type*)
Возвращает инстанс этого класса из созданного json dict'a или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'a или строки

class pyqiwi.types.Profile(*auth_info*, *contract_info*, *user_info*, *obj*)
Профиль пользователя Visa QIWI Кошелька

auth_info
Настройки авторизации пользователя

Type Optional[*AuthInfo*]

`contract_info`
Информация о кошельке пользователя

Type Optional[*ContractInfo*]

`user_info`
Прочие пользовательские данные

Type Optional[*UserInfo*]

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.Rate(_from, to, rate, obj)`
Курс валюты

`_from`
Код входящей валюты (number-3 ISO-4217)

Type int

`to`
Код исходящей валюты (number-3 ISO-4217)

Type int

`rate`
Значение

Type float

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.Statistics(incoming_total, outgoing_total, obj)`
Статистика платежей

`incoming_total`
Данные о входящих платежах (пополнениях), отдельно по каждой валюте

Type list[*TransactionSum*]

`outgoing_total`
Данные об исходящих платежах, отдельно по каждой валюте

Type list[*TransactionSum*]

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

class pyqiwi.types.Transaction(*txn_id*, *person_id*, *date*, *error_code*, *error*, *status*, *_type*, *status_text*, *trm_txn_id*, *account*, *_sum*, *commission*, *total*, *provider*, *source*, *comment*, *currency_rate*, *features*, *view*, *obj*)

Транзакция

`txn_id`
ID транзакции в процессинге

Type int

`person_id`
Номер кошелька

Type int

`date`
Дата/время платежа, время московское

Type datetime.datetime

`error_code`
Код ошибки платежа

Type int/float

`error`
Описание ошибки

Type str

`status`
Статус платежа. Возможные значения: WAITING - платеж проводится, SUCCESS - успешный платеж, ERROR - ошибка платежа.

Type str

`type`
Тип платежа. Возможные значения: IN - пополнение, OUT - платеж, QIWI_CARD - платеж с карт QIWI (QVC, QVP).

Type str

`status_text`
Текстовое описание статуса платежа

Type str

`trm_txn_id`
Клиентский ID транзакции

Type str

`account`
Номер счета получателя

Type str

`sum`
Данные о сумме платежа

Type *TransactionSum*

`commission`
Данные о комиссии платежа

Type *TransactionSum*

`total`
Данные об общей сумме платежа

Type *TransactionSum*

`provider`
Данные о провайдере

Type *TransactionProvider*

`comment`
Комментарий к платежу

Type str

`currency_rate`
Курс конвертации (если применяется в транзакции)

Type float/int

`source`
???

Type ???

`features`
???

Type ???

`view`
???

Type ???

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.TransactionProvider(_id, short_name, long_name, logo_url, description, keys, site_url, obj)`
Данные о провайдере

`id`
ID провайдера в процессинге

Type int

`short_name`
Краткое наименование провайдера

Type str

`long_name`
Развернутое наименование провайдера

Type str

`logo_url`
Ссылка на логотип провайдера

Type str

`description`
Описание провайдера (HTML)

Type str

`keys`
Список ключевых слов

Type str

`site_url`
Сайт провайдера

Type str

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.TransactionSum(amount, currency, obj)`
Данные о платеже

`amount`
Сумма

Type float/int

`currency`
Валюта

Type str

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

`class pyqiwi.types.UserInfo(default_pay_currency, default_pay_source, email, first_txn_id, language, operator, phone_hash, promo_enabled, obj)`
Прочие пользовательские данные

`default_pay_currency`
Код валюты баланса кошелька по умолчанию (number-3 ISO-4217)

Type int

`default_pay_source`
Служебная информация

Type ???

`email`
E-mail пользователя

Type str

`first_txn_id`
Номер первой транзакции пользователя после регистрации

Type int

`language`
Служебная информация

Type ???

`operator`
Название мобильного оператора номера пользователя

Type str

`phone_hash`
Служебная информация

Type ???

`promo_enabled`
Служебная информация

Type ???

`classmethod de_json(json_type)`
Возвращает инстанс этого класса из созданного json dict'а или строки Эта функция должна быть перезаписана для каждого субкласса

Результат

Тип результата Инстанс этого класса из созданного json dict'а или строки

7.3 Exceptions

`exception pyqiwi.exceptions.APIError(msg, method_name, response=None)`
Ошибка в Qiwi API

`msg`
Сообщение ошибки

Type str

`method_name`
Название метода, вызванного при возникновении ошибки

Type str

`request`
Чистый ответ от сервера, полученный от requests

Type requests.Response

`response`
Текст выданный Qiwi API, без какой либо обработки

Type str

`method`
Метод вызванный на сервере Qiwi

Type str

`params`
Параметры вызванного метода

Type dict

Глава 8

Индекс

- genindex
- modindex
- search

Содержание модулей Python

p

Символы

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W