

Telegram Qiwi Bot

Выполнил:
Лехов Кирилл Александрович

Введение

Приложение создано для рядовых пользователей Telegram и Qiwi

С помощью данного продукта можно совершать банковские операции, просматривать статус платежей, искать ближайшие пункты пополнения

Введение (используемые API)

QIWI API: <https://developer.qiwi.com/ru/qiwi-wallet-personal/>

QIWI API карт терминалов: <https://developer.qiwi.com/ru/qiwi-map/>

API для вычисления города по IP: <http://freegeoip.net/>

Static API Яндекс карт:

<https://tech.yandex.ru/maps/staticapi/?from=mapsapi>

Геокодер Яндекс карт:

<https://tech.yandex.ru/maps/geocoder/?from=mapsapi>

Python-Telegram-Bot:

<https://github.com/python-telegram-bot/python-telegram-bot>

В ходе разработки был создан проект на GitHub для работы с QIWI API:

https://github.com/Kirill-Lekhov/QIWI_API.git

Описание реализации

Структура:

Приложение разбито на 3 файла

1) Bot.py — основная логика бота

2) QIWI_API.py — работа с API

3) Languages.json — тест кнопок и диалогов

Описание реализации (особенности)

Приложение использует 6 видов API
Геокодер и Static API от компании «Яндекс»
работают в связке с QIWI-Terminal-Map API и
FreeGEOIP для отображения пользователю
карты терминалов или их адресов.
Так же у пользователя есть возможность
отправить своё местоположение.

Описание реализации (модули)

- 1) urllib — отправка GET запросов
- 2) json — помощь в отправке POST запросов и работа с локализацией
- 3) time — помощь в отправке POST запросов (UNIX TIME)
- 4) requests — отправка POST запросов
- 5) os — очистка рабочей папки после отправки изображения чека
- 6) telegram (Python-Telegram-Bot) — взаимодействие с серверами Telegram

Описание реализации (классы ошибок)

QiviError(Exception):

SyntaxError(QiviError) "Query execution failed"

TokenError(QiviError) "Wrong TOKEN"

class NoRightsError(QiviError): "No right"

TransactionNotFound(QiviError) "Transaction not found or missing payments with specified characteristics"

WalletError(QiviError) "Wallet not found"

HistoryError(QiviError) "Too many requests, the service is temporarily unavailable"

MapError(QiviError) "Map processing errors"

NotFoundAddress(MapError) "Could not find address"

CheckError(QiviError) "Could not get check"

WrongEmail(CheckError) "Wrong Email address"

WrongNumber(QiviError) "Wrong phone number"

TransactionError(QiviError) "Failed to carry out the transaction"

Описание реализации

class UserQiwi

UserQiwi.url = <https://edge.qiwi.com/>

self.token(str) = токен пользователя

self.headers(dict) = headerы для выполнения запросов

self.urls(dict) = формы заполнения ссылок для запросов

self.currency(dict) = «конвертер валюты»

self.identification(dict) = «конвертер идентификаторов»

self.user_date(dict) = информация о пользователе

Описание реализации class UserQiwi

Название	Переменные	Описание	Тип возвращаемых данных
<code>__init__</code>	<code>self, token</code>	объявляет нового пользователя	None
<code>change_token</code>	<code>self, new_token</code>	меняет токен пользователя	None
<code>get_user_token</code>	<code>self</code>	возвращает пользовательски токен	str
<code>get_balance</code>	<code>self</code>	возвращает баланс счетов пользователя	str
<code>update_info</code>	<code>self</code>	обновляет <code>self.user_data</code>	None
<code>get_info</code>	<code>self</code>	возвращает информацию о пользователе	str

Описание реализации class UserQiwі

Название	Переменные	Описание	Тип возвращаемых данных
get_last_transactions	self, rows (по умолчанию = 10)	Возвращает последние транзакции пользователя	str
get_info_about_transaction	self, transaction_id	Возвращает информацию по транзакции	str
get_map_terminates	self, address (по умолчанию None)	Возвращает ссылку на изображение карты и адреса терминалов	dict
get_image_check	self, transaction_id, file_name (по умолчанию «check.jpg»)	Создает изображение чека указанной транзакции (не работает с IN типом транзакций)	None
send_check_email	self, transaction_id, email (по умолчанию None)	Отправляет изображение чека на почту. Если почта не указана используется почта пользователя	None

Описание реализации class UserQiwi

Название	Переменные	Описание	Тип возвращаемых данных
transaction_telephone	self, amount, number (по умолчанию None)	Переводит средства со счета владельца на указанный номер. Если такого нет, используется номер пользователя.	None
transaction_qiwi	self, account_id, amount	Переводит средства со счета владельца на указанный счет.	None

Описание реализации (функции QIWI_API)

`run_the_query(headers, url)` — Выполняет GET запрос (`urllib + json`) | `dict/bool`

`found_address(ip)` — Выполняет GET запрос (`requests`). Возвращает город определенный по ip адресу | `str/bool`

`write_file(headers, url, file_name)` — Выполняет GET запрос (`urllib`). Записывает изображение чека в указанный файл. | `bool`

`found_id(number)` — Выполняет POST запрос (`requests`). Возвращает id мобильного оператора. | `str/bool`

Описание реализации Bot.py

Название	Переменные	Описание
start	bot, update	Начало работы с ботом
check_token	bot, update, user_data	Проверка токена
balance	bot, update, user_data	Отправляет баланс
transactions	bot, update	Начало диалога по поводу транзакций.
check_status	bot, update	Просьба ввести id транзакции
answer_about_transaction	bot, update, user_data	Отправляет информацию о транзакции
last	bot, update, user_data	Отпрвляет информацию о последних транзакицах

Описание реализации Bot.py

Название	Переменные	Описание
terminals	bot, update, user_data	Начало работы с транзакциями
take_command_found_address	bot, update, user_data	Выбор типа ответа
take_address	bot, update	Просьба ввести адрес
take_location	bot, update, user_data	Обработка координат геолокации пользователя
answer_about_terminates	bot, update	Отправка ответа
options	bot, update	Начало диалога по поводу настроек
get_info	bot, update, user_data	Отправляет информацию о пользователе

Описание реализации Bot.py

Название	Переменные	Описание
take_new_token	bot, update	Смена пользователя
update_user	bot, update, user_data	Обновление информации аккаунта
version	bot, update	Отправка версии программы
check	bot, update, user_data	Начало диалога про чеки
dialog_email	bot, update, user_data	Начало диалога про почту
enter_email	bot, update	Просьба ввести email
get_email	bot, update, user_data	Получение email. Просьба ввести id транзакции

Описание реализации Bot.py

Название	Переменные	Описание
enter_transaction_id	bot, update	Просьба ввести id транзакции.
get_transaction_id	bot, update, user_data	Вывод на экран/отправка на почту копии чека
pay	bot, update, user_data	Начало диалога про оплату
enter_user_id	bot, update, user_data	Просьба ввести id клиента
get_user_id	bot, update, user_data	Получение id клиента. Просьба ввести сумму перевода.
mobile_phone	bot, update, user_data	Начало диалога про выбор номера мобильного телефона
enter_mobile	bot, update	Просьба ввести номер мобильного телефона

Описание реализации Bot.py

Название	Переменные	Описание
get_mobile	bot, update, user_data	Получение номера мобильного телефона. Просьба ввести сумму оплаты.
enter_amount	bot, update	Просьба ввести сумму оплаты.
get_amount	bot, update, user_data	Оплата мобильного/перевод клиенту Qiwi
wrong_answer	bot, update	Отправка указания на использование клавиатуры
back	bot, update	Возврат к начальному меню.
stop	bot, update	Завершение работы с ботом

Заключение

В ходе работы над проектом я научился работать с API одной из крупнейших банковских систем и освоил его возможности приблизительно на 60%.

В целом я доволен проделанной работой.

В дальнейшем можно оптимизировать работу бота.

Добавить больше видов оплаты.

Доделать локализацию приложения.

Кастомизировать взаимодействие с ботом.

Улучшить обработку карты терминалов