
mikud

Release 0.0.3

David Lev

Jun 12, 2022

CONTENTS

1	Installation	3
2	Features	5
3	Usage	7
4	Requirements	9
5	Setup and Usage	11
6	Disclaimer	13
7	Reference	15
	Index	19



mikud is a Python3 library to search israel's zip codes (mikud) by their addresses.

For a **complete documentation** of available functions, see the [Reference](#).

>> *This tool based on israelpost website - [Click here](#).*

INSTALLATION

- **Install using pip3:**

```
pip3 install -U mikud
```

- **Install from source:**

```
git clone https://github.com/david-lev/mikud.git  
cd mikud && python3 setup.py install
```


FEATURES

- Search zip code by address
- Search address by zip code
- Get cities
- Get streets

USAGE

```
from mikud import Mikud

mikud = Mikud()
mikud_res = mikud.search_mikud(city_name="",
                                street_name=" ",
                                house_number="20")

print(mikud_res.zip)
#> 9546432

addr = mikud.search_address(zip_code=9546432)
print(addr.city_name, addr.street_name, addr.house_number)
#> 20
```


REQUIREMENTS

- Python 3.6 or higher - <https://www.python.org>

SETUP AND USAGE

See the [Documentation](#) for detailed instructions

DISCLAIMER

This application is intended for educational purposes only. Any use in professional manner or to harm anyone or any organization doesn't relate to me and can be considered as illegal.

REFERENCE

Mikud.**search_mikud**(city_name: *str* = "", city_id: *Union[str, int]* = "", street_name: *str* = "", street_id: *Union[str, int]* = "", house_number: *Union[str, int]* = "", pob: *Union[str, int]* = "", entry: *str* = "") → *Address*

Get zip code (mikud) of address.

- You can specify the name of the city and street (or their id from `search_cities()` and `search_streets()`) and the house number.
- Alternatively, you can specify just the `city_name` (or `city_id`) and the pob (Post office box) to get the zip code.

Parameters

- **city_name** (*str*) – City name. Default - empty. Optional if pob provided.
- **city_id** (*Union[int, str]*) – City ID. Default - empty. Optional if city_name provided.
- **street_name** (*str*) – Street name. Default - empty. Optional if pob provided.
- **street_id** (*Union[int, str]*) – Street ID. Default - empty. Optional if street_name or pob provided.
- **house_number** (*Union[int, str]*) – House number. Default - empty. Optional if pob provided.
- **pob** (*Union[int, str]*) – Post office box. Default - empty. Optional if city_name, street_name and house_number provided.
- **entry** (*str*) – Entry. for example - , . Default - empty. Optional if pob provided.

Returns

Object *Address()*

Return type

Address()

Mikud.**search_address**(zip_code: *Union[int, str]*) → *Address*

Get address of zip code (mikud).

Parameters

- **zip_code** (*Union[int, str]*) – Zip code of 7 numbers.

Raises

- **Exception** – if zip code length is not 7 numbers.

Returns

Object *Address()*.

Return type

Address()

Mikud.get_cities(*city_name: str*) → *List[City]*

Search cities by name to use in *search_mikud()* and in *search_streets()*.

Parameters

city_name (*str*) – The full name or part of it of the city.

Returns

list with cities *City()*.

Return type

List[City()]

Mikud.get_streets(*city_name: str, street_name: str, city_id: Union[int, str] = ""*) → *List[Street]*

Search street by name to use in *search_mikud()*.

Parameters

- **city_name** (*str*) – The full name or part of it of the city name. Optional if *city_id* provided.
- **city_id** (*Union[int, str]*) – City id from *search_cities()*. Optional if *city_name* provided.
- **street_name** (*str*) – The full name or part of it of the street name.

Returns

List of *Street()*.

Return type

List[Street()]

```
class mikud.Address(zip: Optional[int] = None, city: Optional[str] = None, cityname: Optional[str] = None,  
cityid: Optional[int] = None, cityID: Optional[int] = None, street: Optional[str] = None,  
streetid: Optional[int] = None, streetID: Optional[int] = None, housenumber:  
Optional[int] = None, pob: Optional[int] = None, messageResult: Optional[str] = None)
```

This class represent a Address object.

- The following parameters are the attributes of this object.
- Ignore the names of the parameters in the class structure ^, they are there to deal with the json parsing.

Parameters

- **zip** (*Union[int, None]*) – Zip code of 7 numbers.
- **city_name** (*Union[str, None]*) – The name of the city.
- **city_id** (*Union[int, None]*) – The id of the city.
- **street_name** (*Union[str, None]*) – The name of the street.
- **street_id** (*Union[int, None]*) – The id of the street.
- **house_number** (*Union[int, None]*) – The number of the house.
- **pob** (*Union[int, None]*) – The number of the post office box.
- **results** (*Union[str, None]*) – The message from the api.

```
class mikud.City(id: Optional[int] = None, n: Optional[str] = None, zip: Optional[int] = None)
```

This class represent a city object.

- The following parameters are the attributes of this object.
- Ignore the names of the parameters in the class structure ^, they are there to deal with the json parsing.

Parameters

- **id** (*Union[int, None]*) – The id of the city.
- **name** (*Union[str, None]*) – The name of the city.
- **zip** (*Union[int, None]*) – The zip code of the city.

```
class mikud.Street(cityID: Optional[int] = None)
```

This class represent a street object.

- The following parameters are the attributes of this object.
- Ignore the names of the parameters in the class structure ^, they are there to deal with the json parsing.

Parameters

- **id** (*Union[int, None]*) – The id of the street.
- **city_id** (*Union[int, None]*) – The id of the city.
- **name** (*Union[str, None]*) – The name of the city.
- **zip** (*Union[int, None]*) – The zip code of the address.

INDEX

A

`Address` (*class in mikud*), [16](#)

C

`City` (*class in mikud*), [17](#)

G

`get_cities()` (*mikud.Mikud method*), [16](#)

`get_streets()` (*mikud.Mikud method*), [16](#)

S

`search_address()` (*mikud.Mikud method*), [15](#)

`search_mikud()` (*mikud.Mikud method*), [15](#)

`Street` (*class in mikud*), [17](#)