

---

# **Py Zip Code API Documentation**

***Release 1.0.0***

**Mohamed Ben Makhlouf**

**Jan 10, 2023**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Options</b>	<b>5</b>
<b>3</b>	<b>Example</b>	<b>7</b>
<b>4</b>	<b>Output</b>	<b>9</b>
<b>5</b>	<b>Contributing</b>	<b>11</b>
<b>6</b>	<b>Issues</b>	<b>13</b>



Py ZipCodeApi will make it easier for you to use the different options in [ZipCodeAPI](#).



# CHAPTER 1

---

## Installation

---

Use pip to install from PyPI:

```
pip install pyzipcodeapi
```

Register for free api\_key [here](#).





## CHAPTER 2

---

### Options

---

As mentioned in the original website, the following options are supported in this package :

- `distance`
- `radius`
- `match-close`
- `info`
- `multi-info`
- `city-zips`
- `radius-sql`



## CHAPTER 3

### Example

```
# set different inputs
f = 'json'
u = 'km'
ou = 'degrees'
obj = ZipCodeApi(API_KEY)
# https://www.zipcodeapi.com/rest/<api_key>/distance.<format>/<zip_code1>/<zip_code2>/
↪</units>
print(obj.get('distance', f).filter(zip_code1='94106', zip_code2='94132', units=u))
print('-----')
# https://www.zipcodeapi.com/rest/<api_key>/radius.<format>/<zip_code>/<distance>/
↪<units>
print(obj.get('radius', f).filter(zip_code='94120', distance='94132', units=u))
print('-----')
# https://www.zipcodeapi.com/rest/<api_key>/match-close.<format>/<zip_codes>/
↪<distance>/<units>
print(obj.get('match-close', f).filter(zip_codes='941asd32', distance='5', units=u))
print('-----')
# https://www.zipcodeapi.com/rest/<api_key>/info.<format>/<zip_code>/<units>
print(obj.get('info', f).filter(zip_code='94132', units=ou))
print('-----')
# https://www.zipcodeapi.com/rest/<api_key>/multi-info.<format>/<zip_code>/<units>
print(obj.get('multi-info', f).filter(zip_code='94132', units=ou))
print('-----')
# https://www.zipcodeapi.com/rest/<api_key>/city-zips.<format>/<city>/<state>
print(obj.get('city-zips', f).filter(city='San Francisco', state='CA'))
print('-----')
# https://www.zipcodeapi.com/rest/<api_key>/radius-sql.<format>/<lat>/<long>/<lat_
↪long_units>/<distance>/<units>/
# <lat_field_name>/<long_field_name>/<precision>
print(obj.get('radius-sql', f).filter(
    lat='37.722223',
    long='-122.484048',
    lat_long_units=ou,
    distance='5',
```

(continues on next page)

(continued from previous page)

```
    units=u,  
    lat_field_name='lat',  
    long_field_name='long',  
    precision='4'  
))
```

## CHAPTER 4

---

### Output

---

for each request you make, you can choose between the different outputs :

- json
- csv (the output is an instance from CSV Reader Objects [DictReader](#))
- xml

**Warning:** Depend on the option you will choose, Please refer to the [original website](#) to see the context of each output.



## CHAPTER 5

---

### Contributing

---

To contribute to PyZipCodeAPI [create a fork](#) on GitHub. Clone your fork, make some changes, and submit a pull request.





## CHAPTER 6

---

### Issues

---

Use the GitHub [issue tracker](#) for PyZipCodeAPI to submit bugs, issues, and feature requests.