Fetching Mobile Number Details Using Python

Using Phonenumbers for Retrieving Mobile Number Details

Himanshu Sharma 1 day ago · 3 min read★

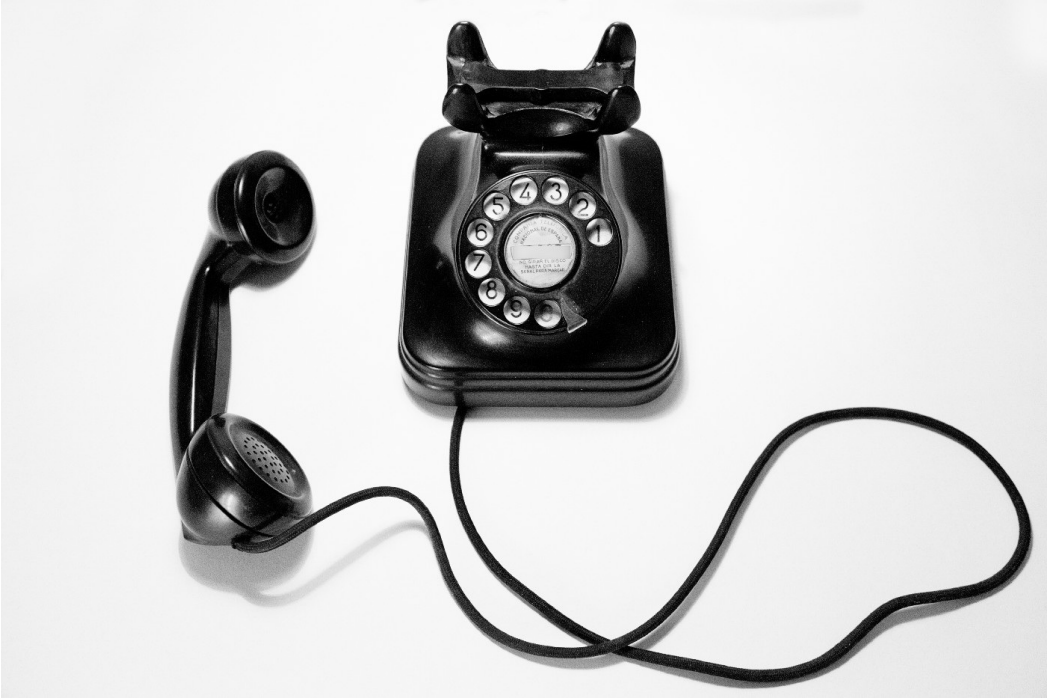

Photo by Quino Al on Unsplash

Python is a very rich language due to a large number of libraries that it offers. It contains a variety of functionalities that can be helpful in solving real-life business problems. Today we will be talking about one such library.

Phonenumbers is an open-source Python library that is used for accessing information for phone numbers. It also helps in validating a phone number, parsing a phone number, etc.

In this article, we will explore Phonenumbers and their functionalities.

Let's get started…

Installing required libraries

We will start by installing Phonenumbers using pip installation. The command given below will install Phonenumbers using pip.

```
pip install phonenumbers
```

Importing required libraries

In this step, we will import all the libraries that are required for accessing the information of a phone number.

```
import phonenumbers
from phonenumbers import carrier, geocoder, timezone
```

Accessing Information about a Phone Number

Now we will access the information for any phone number that we want, we need to pass the country code followed by the phone number and it will display the details of that number.

```
mobileNo=input("Mobile no. with country code:")
mobileNo=phonenumbers.parse(mobileNo)
print(timezone.time_zones_for_number(mobileNo))
print(carrier.name_for_number(mobileNo,"en"))
print(geocoder.description_for_number(mobileNo,"en"))
print("Valid Mobile Number:",phonenumbers.is_valid_number(mobileNo))
print("Checking possibility of
Number:",phonenumbers.is_possible_number(mobileNo))
```


Information(Source: By Author)

Here you can see how the information related to the number that we have provided is displayed. Similarly, you can check it for different mobile numbers.

Next, let us see how we can parse a phone number from a text and how to format it. The code given below will perform these operations.

```
text = "Call me at 1800-202-2022 if it's before 9:30, or on 703-
4800500 after 10am."
for match in phonenumbers.PhoneNumberMatcher(text, "US"):
    print(match)
x = phonenumbers.parse("+442083661177", None)
phonenumbers.format_number(x,
phonenumbers.PhoneNumberFormat.NATIONAL)
```


Parsing(Source: By Author)

Try this with different texts and numbers to parse the data and format the phone number. You can easily implement this in any real-life project or problem. Let me know what you think about this library in the response section.

This article is in collaboration with Piyush Ingale.

Before You Go

Thanks *for reading! If you want to get in touch with me, feel free to reach me at hmix13@gmail.com or my* ***LinkedIn Profile****. You can view my* ***Github*** *profile for different data science projects and packages tutorials. Also, feel free to explore* ***my profile*** *and read different articles I have written related to Data Science.*

Data Science | Data Visualization | Machine Learning | Artificial Intelligence | Python