**AI Documentation Entry #1**  
**Prompt:**  
I'm trying to build a function in Python to get coordinates (latitude and longitude) from a place name using the GeoNames API. I'm not sure how to make the request and extract the result.

**Response:**  
Sure! You're on the right track. To query GeoNames for a place, you'll use the requests library to send a GET request to their searchJSON endpoint. Here's how to think about it:

1. Create a URL like:  
   http://api.geonames.org/searchJSON?q=PLACE\_NAME&username=YOUR\_USERNAME&maxRows=1
2. Use requests.get(url, params=your\_parameters) to send the request.
3. The response will be a JSON object — you can extract the coordinates by accessing:  
   response.json()['geonames'][0]['lat'] and 'lng'.

Here’s a small snippet for how you might structure it:

python

Copy code

params = {"q": place, "username": username, "maxRows": 1}

response = requests.get("http://api.geonames.org/searchJSON?", params=params)

data = response.json()

lat = data['geonames'][0]['lat']

Let me know if you get stuck with error handling or adding time delays.

**AI Documentation Entry #2**  
**Prompt:**  
I’m getting an IndexError when trying to extract the first result from the JSON. What should I do to prevent that?

**Response:**  
Good observation! That usually means that no result was returned. You should wrap your extraction code in a try-except block to handle cases where geonames might be an empty list.

Here’s a pattern you can follow:

python

Copy code

try:

result = data['geonames'][0]

lat = result['lat']

lng = result['lng']

except (IndexError, KeyError):

lat = "NA"

lng = "NA"

This ensures your program won't crash and you can still save "NA" if the location isn't found.

**AI Documentation Entry #3**  
**Prompt:**  
I want to read a TSV file and extract place names from the column titled "place". How can I find the correct column index?

**Response:**  
You're on the right path! First, read the first line of the TSV file (the header) and split it by tab characters (\t). Then use the .index() function to find the column number for "place".

Example:

python

Copy code

header = lines[0].strip().split('\t')

place\_index = header.index("place")

Now you can loop through the rest of the lines and grab the place name using columns[place\_index].

**AI Documentation Entry #4**  
**Prompt:**  
Should I add a delay when querying the API in a loop to avoid being blocked?

**Response:**  
Yes, definitely. GeoNames (and most APIs) have rate limits. It’s good practice to add a time.sleep() call between requests. A one-second delay (time.sleep(1)) is usually safe for small-scale use.

**AI Documentation Entry #5**  
**Prompt:**  
How should I write the output with place names and their coordinates to a TSV file?

**Response:**  
After collecting the data into a list of dictionaries, open the file in write mode and write a header line first. Then, loop through your list and write each row with tab-separated values.

Here’s a general idea:

python

Copy code

with open("output.tsv", 'w', encoding="utf-8") as file:

file.write("Place\tLatitude\tLongitude\n")

for row in coordinates\_data:

file.write(f"{row['Place']}\t{row['Latitude']}\t{row['Longitude']}\n")

Make sure to check your list for any unexpected values before writing.