Geocodio Python Library: Project Plan

This document outlines the phases, tasks, and timeline for developing the official Geocodio Python Library, designed to be fully feature-comparable to the official PHP client.

📅 Timeline Overview

Phase	Duration
1. Research & Design	1-2 weeks
2. Core Implementation (Sync)	2-3 weeks
3. Async Implementation	2-4 weeks
4. Improvements & Robustness	1-2 weeks
5. Testing & Documentation	2 weeks
6. Automation & Publishing	1 week
Total Estimated Time	**9-12 weeks**

🛠️ Tasks by Phase

Phase 1: Research & Design

- Review existing Geocodio libraries and official API.
- Create detailed API design ensuring feature parity with the PHP client.
- **Deliverable:** API specification document.

Phase 2: Core Implementation (Sync)

- Implement synchronous methods: forward/reverse geocoding, batch processing, address parsing, data appending.
- **Deliverable:** Working synchronous Python library.

Phase 3: Async Implementation

- Implement asynchronous support using Python's asyncio and aiohttp libraries, enabling concurrent requests for improved performance.

Example:

```
from geocodio import AsyncGeocodioClient

async def geocode_addresses(addresses):
    client = AsyncGeocodioClient("YOUR_API_KEY")
    results = await client.batch_geocode(addresses)
    return results
```

- **Deliverable:** Functional async capabilities alongside synchronous methods.

Phase 4: Improvements & Robustness

- Implement precise rate limit handling with exponential backoff.

Example:
If a request exceeds the API limit, the library automatically retries after incremental delays (e.g., 1 second, 2 seconds, 4 seconds).

- Enhance error handling, logging clarity, and overall stability.
- **Deliverable:** Robust, reliable library with clear error reporting.

Phase 5: Testing & Documentation

- Write comprehensive unit and integration tests.
- Create detailed user documentation and examples, provided in the format requested by Geocodio to seamlessly integrate with current library documentation.
- **Deliverable:** Fully tested and documented library.

Phase 6: Automation & Publishing

- Configure automated testing and publishing workflows matching the official PHP client's CI/CD practices.
- **Deliverable:** Official Python library published to PyPI.

🎯 Key Benefits

- **Async Support:** Enables users to handle multiple geocoding requests concurrently, significantly boosting throughput and application responsiveness.
- **Reliability:** Clear and automated management of API rate limits and error conditions, reducing developer overhead and manual intervention.
- **Ease of Integration:** Documentation aligned directly with Geocodio's existing library documentation for consistency and ease of use.
- **Simplified Ongoing Maintenance:** Automated testing and publishing workflows streamline updates and ensure ongoing library health with minimal effort.

📦 Deliverables

- Complete Python library feature-equivalent to the official PHP library.
- Robust synchronous and asynchronous client implementations.
- Comprehensive rate-limit handling, error management, and clear logging.
- Fully integrated user documentation matching Geocodio's documentation standards.
- Automated CI/CD setup for ongoing ease of maintenance and deployment.

📌 Next Steps

- Confirm and approve the API design (Phase 1).
- Begin core feature development (Phase 2).

Please let us know if you have any questions or require further details.

📅 Project Timeline Summary

If the project begins on **Monday, April 15, 2025**, the estimated completion window is:

- **Earliest end date:** June 16, 2025 (9 weeks)
- **Latest end date:** July 21, 2025 (12 weeks)

This assumes a consistent weekly commitment of approximately 30 hours.