API Aggregator - Documentation

1. Overview

The API Aggregator is a .NET 9 Web API responsible for consolidating data from multiple external APIs, such as weather, news, and GitHub repositories. It provides a unified endpoint to fetch, filter, and sort aggregated information in a scalable, parallelized, and cache-optimized manner. The API also supports real-time performance statistics, and structured error handling.

2. Codebase Structure

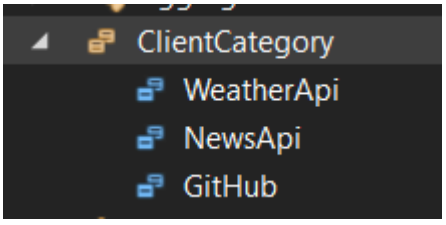
- **AposAPI_Aggregator**: The Web API project containing controllers and startup configuration.
    - Controllers:
      - AggregationController: Exposes endpoints to fetch aggregated data.
      - StatisticsController: Provides runtime performance metrics for each external API.
- **Application**: Business logic layer responsible for orchestrating aggregation.
- **Domain**: Contains core DTOs, enums, and interfaces shared across the app.
- **Clients**: Contains API client implementations (Weather, News, GitHub).
- **Tests**: NUnit-based test project for unit and integration testing.

3. Actions for Adding a New API Client

Each API client implements the IApiClient interface and is registered in the DI container. Clients are categorized and dynamically invoked based on the request parameters.

To add a new external API client:

1. Add the Api Name-Category on ClientCategory enum DTO



2. Implement the IApiClient interface in a new class inside Client library.

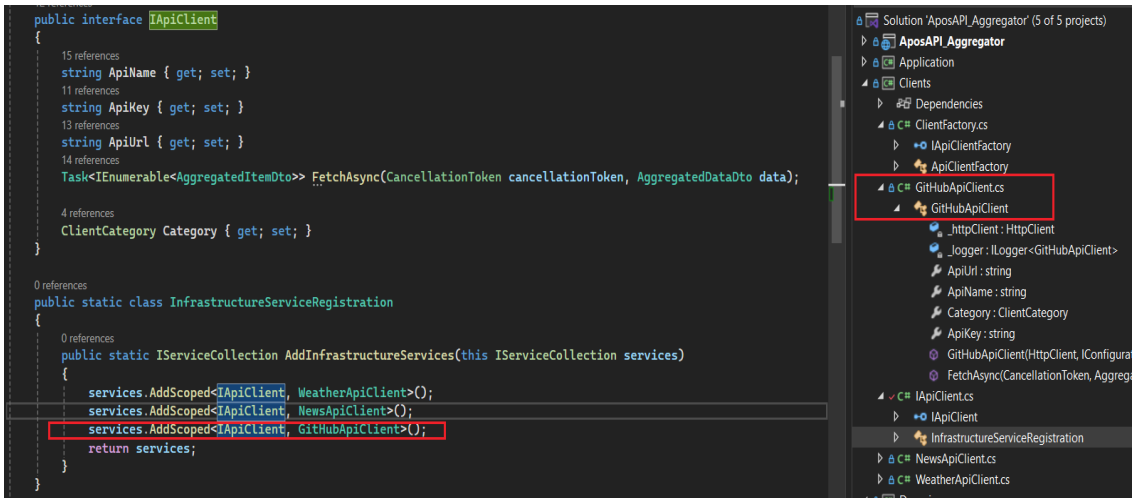


Illustration 1: Implement IApiCLient and Add to Service

3. Register the client in AddInfrastructureServices with AddHttpClient and scoped binding.

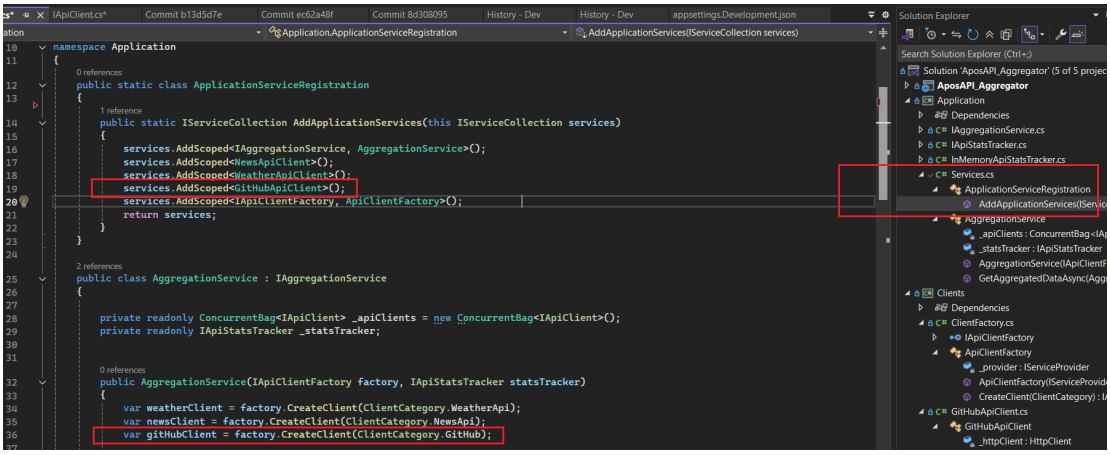


Illustration 2: Register client to APplication

4. Use HttpClient to fetch and deserialize data from the external API

   Convert the API response into a list of AggregatedItemDto.
   Inject IApiStatsTracker and ILogger to log response time and errors.
   Assign a Category to the client (e.g., 'weather', 'code') to allow dynamic filtering.

5. Optionally, add an integration test using NUnit to validate behavior and reliability.