API Aggregator – Documentation

## **1. Overview**

Το **API Aggregator** συνδέεται με τρεις εξωτερικές πηγές (HackerNews, GitHub, Open-Meteo) και επιστρέφει ενοποιημένα αποτελέσματα μέσω ενός ενιαίου endpoint.

* Στόχος: Ενοποίηση, caching, resilience.
* Output: JSON με unified schema (Title, Url, PublishedAt, Source, κ.ά.).

## **2. Architecture**

* **AggregateService** → core service που ενώνει τα αποτελέσματα.
* **IApiConnector** → abstraction για connectors.
  + HnApiConnectorAdapter, GhApiConnectorAdapter, MeteoApiConnectorAdapter.
* **Resilience** → Policies με Polly (timeout, retries, circuit breaker).
* **Caching** → In-memory caching με TTL per source.
* **Stats** → ApiStatsSnapshot για monitoring.
* **(Optional)** → JWT Bearer authentication για secured access.

## **3. Setup & Run**

### **Prerequisites**

* .NET 8/9 SDK
* (προαιρετικά) Docker για integration με εξωτερικά APIs

***Run***

dotnet build

dotnet run --project Aggregator.Api

### **Configuration**

appsettings.json (snippet):

{

"Auth": {

"Enabled": false,

"Audience": "api-aggregator",

"Authority": "https://localhost"

}

}

## **4. Endpoints**

### **/api/aggregate**

Επιστρέφει aggregated items από πολλαπλές πηγές.

**Query parameters**:

* query (string) → τι ψάχνουμε
* sources (comma-separated: hn,gh,meteo)
* limit (int) → μέγιστος αριθμός αποτελεσμάτων
* latitude, longitude (double) → για geo sources

**Example**:

curl "http://localhost:5134/api/aggregate?sources=hn,gh&query=dotnet&limit=5"

### **/health**

Health check endpoint (always open).

## **5. Optional Features**

* **JWT Bearer Authentication** (extra credits)
  + Dev tokens με dotnet user-jwts.
  + Swagger “Authorize” κουμπί για δοκιμή.
* **Swagger UI** → διαθέσιμο στο /swagger.
* **Stats** → monitoring hits/misses/errors (δεν εκτίθενται public, μόνο μέσω logs).

## **6. Testing**

### **Unit Tests**

* Framework: xUnit + Moq + FluentAssertions.
* Βασικά scenarios:
  + Merge & sort
  + Sources filter
  + Cache hits/misses
  + Error isolation

***Run***

dotnet test

## **7. Notes & Credits**

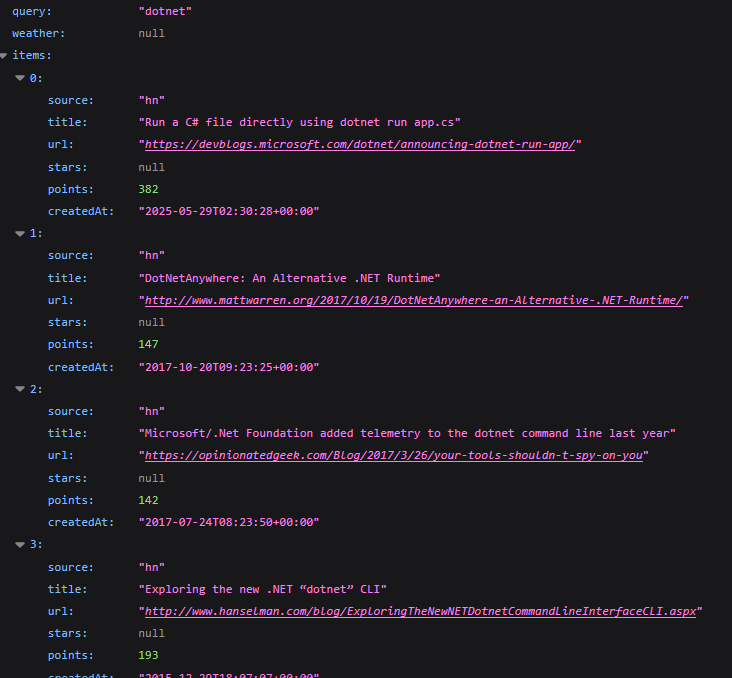
* Το project είναι self-contained, δεν απαιτεί database.
* **Optional JWT** security δίνει extra credits (δεν είναι υποχρεωτικό).
* Χρησιμοποιήθηκαν best practices για DI, caching και resilience.

*Extras:*

* Aggregation από HackerNews + GitHub

curl "http://localhost:5109/api/aggregate?sources=hn,gh&query=dotnet&limit=5"

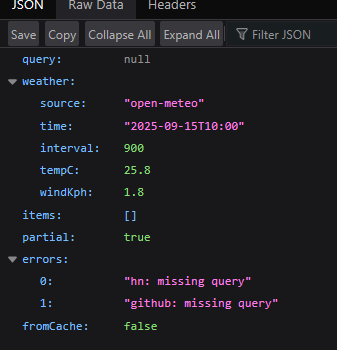
Αναμενόμενο αποτέλεσμα (JSON screenshot):



* Meteo connector με γεωγραφικές συντεταγμένες

"http://localhost:5109/api/aggregate?sources=meteo&latitude=37.98&longitude=23.72&limit=3"

Αναμενόμενο αποτέλεσμα (JSON screenshot):



### **Προστατευμένο endpoint με JWT (extra credits)**

Ενεργοποιημένο Auth:Enabled=true → /api/aggregate απαιτεί token.

curl -H "Authorization: Bearer <JWT>" \

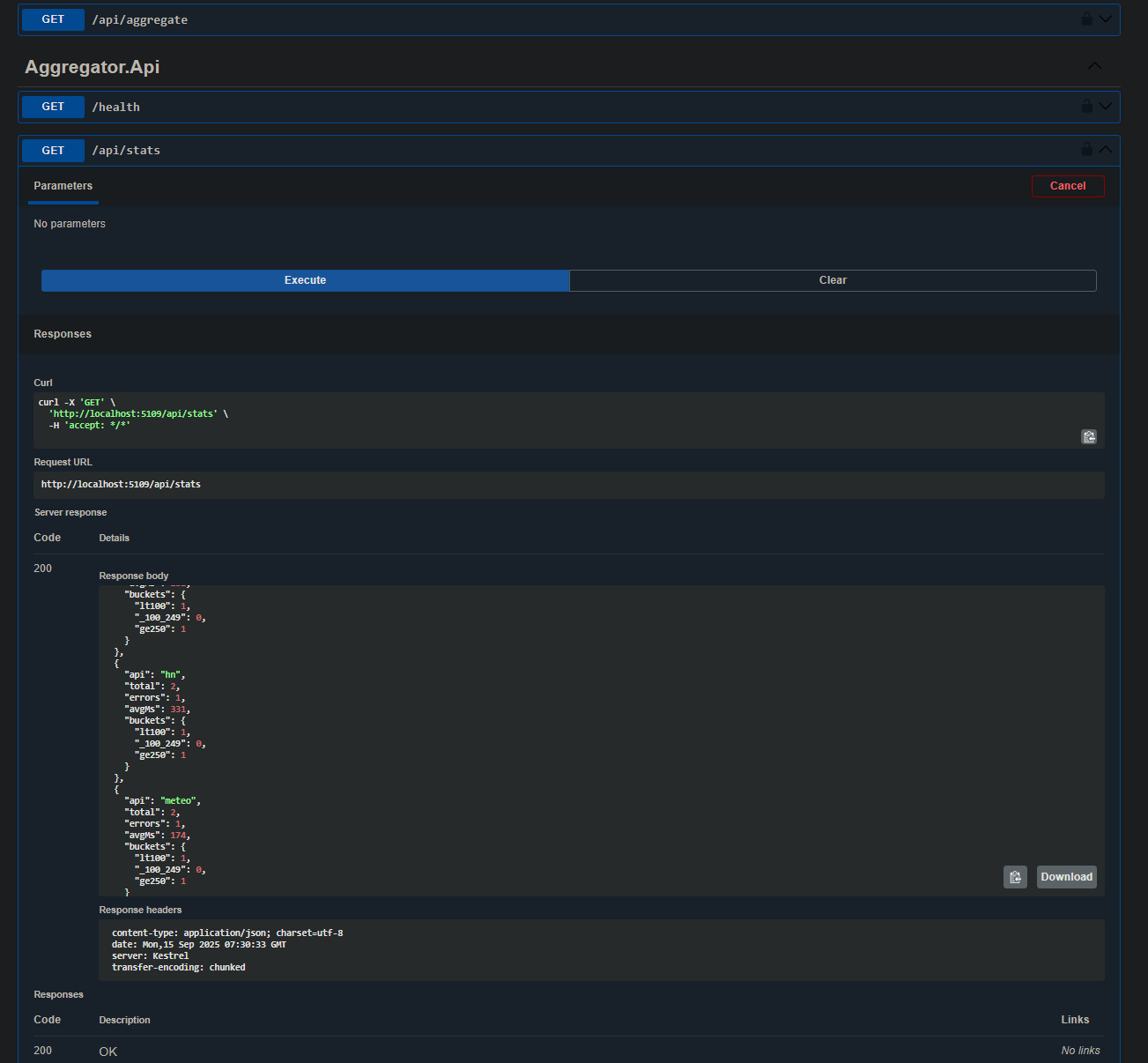
"http://localhost:5109/api/aggregate?sources=hn&query=async&limit=2"

**Χωρίς token:**

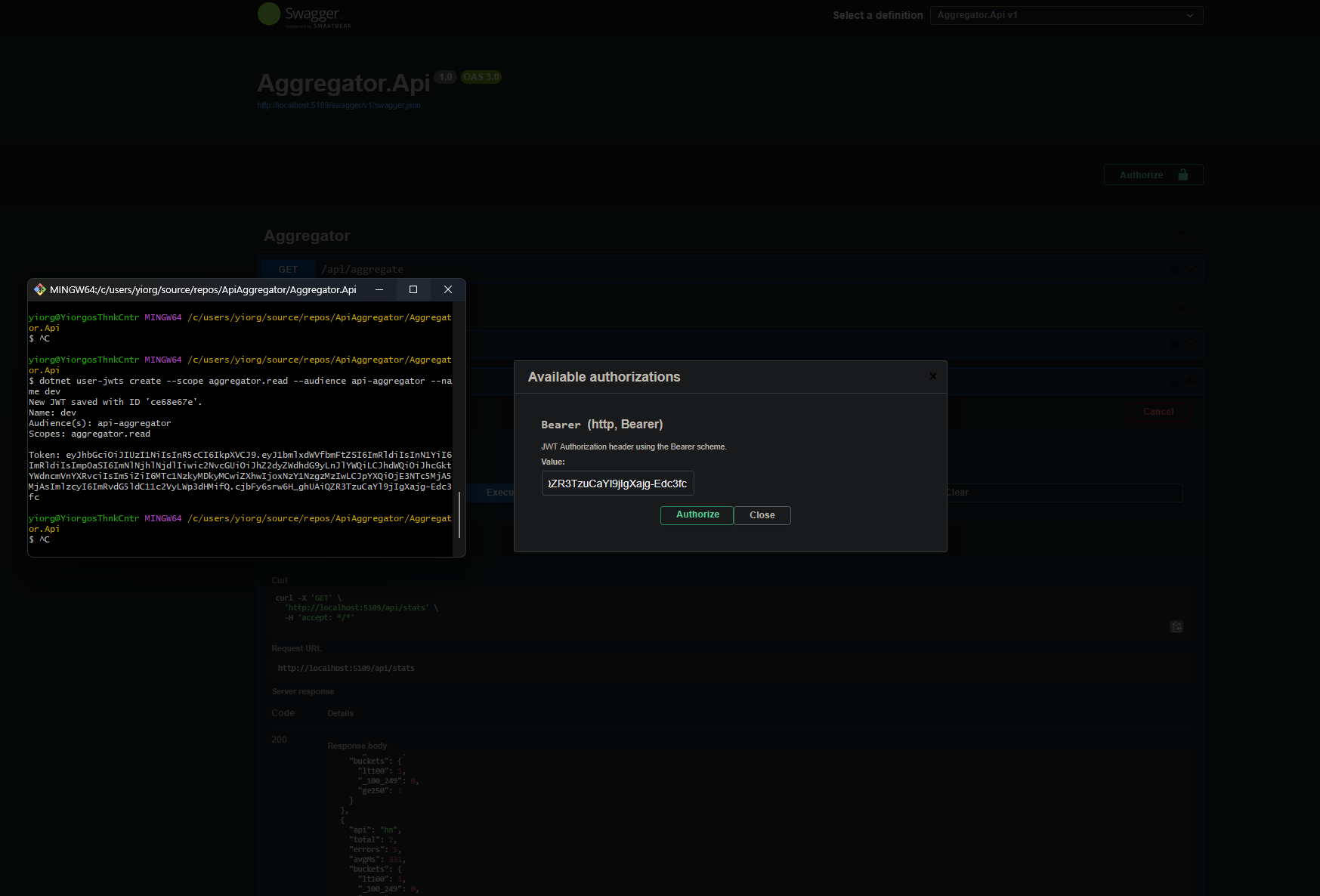
curl "http://localhost:5109/api/aggregate?sources=hn&query=async"

# → HTTP/1.1 401 Unauthorized

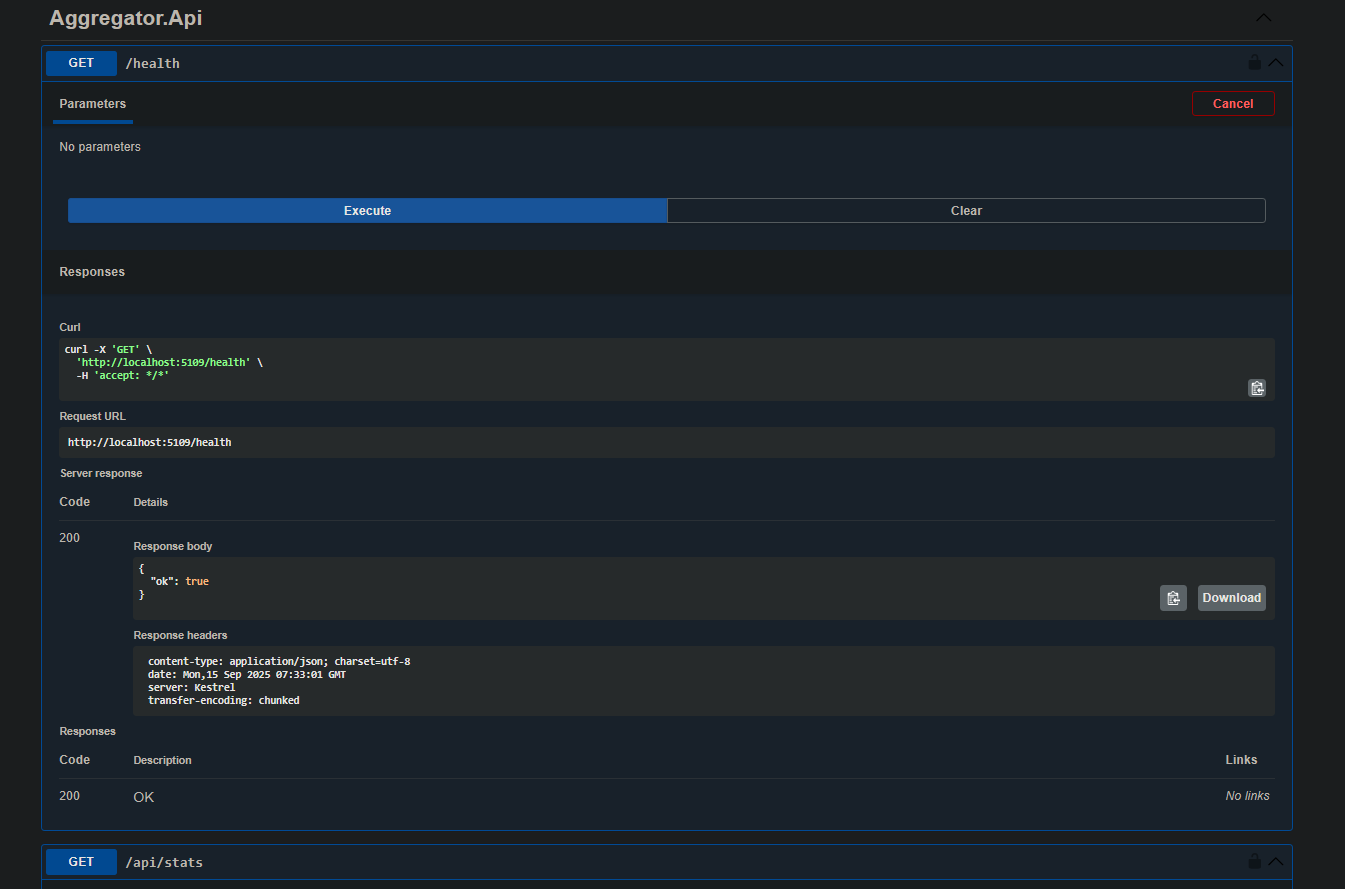
* Stats



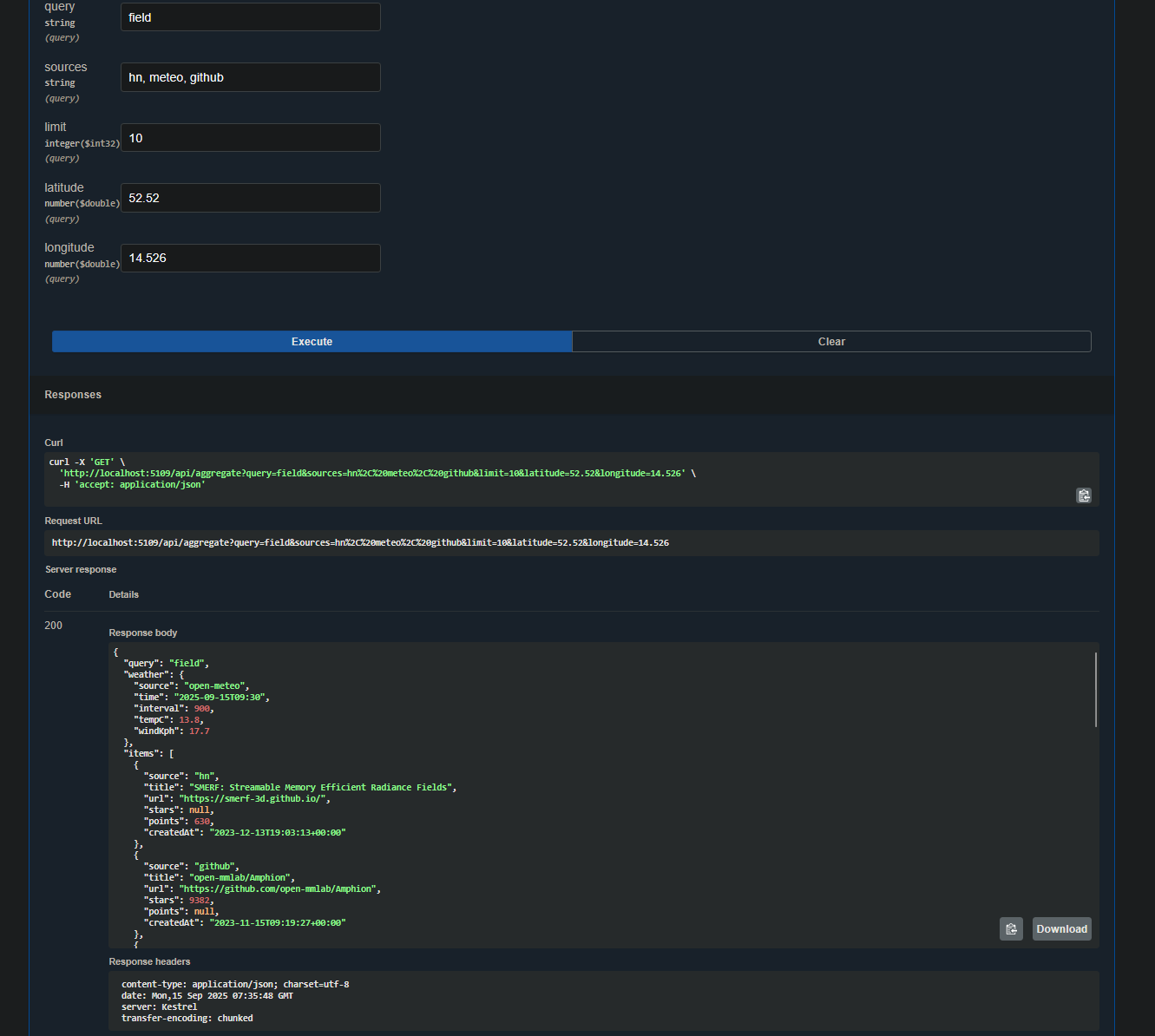
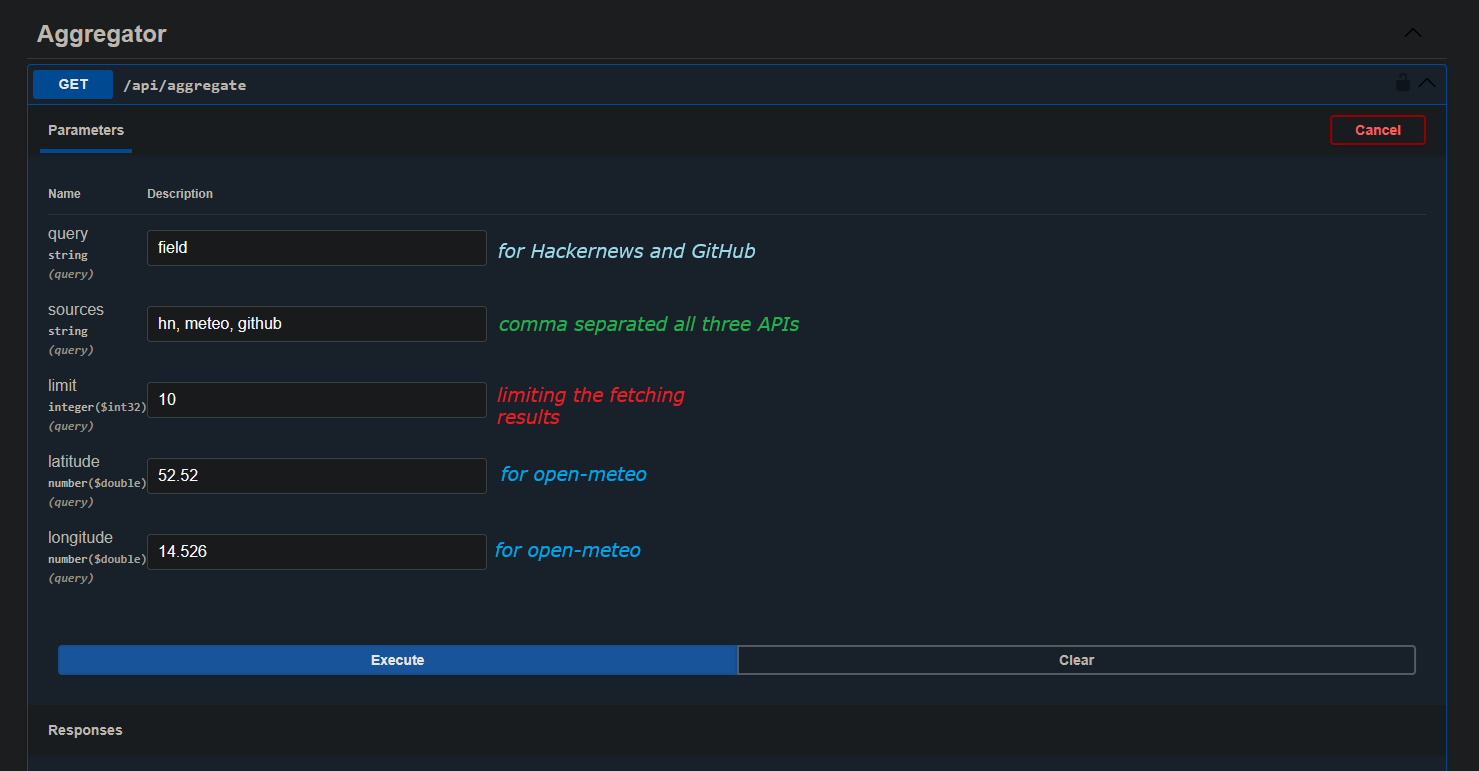
* Authorization



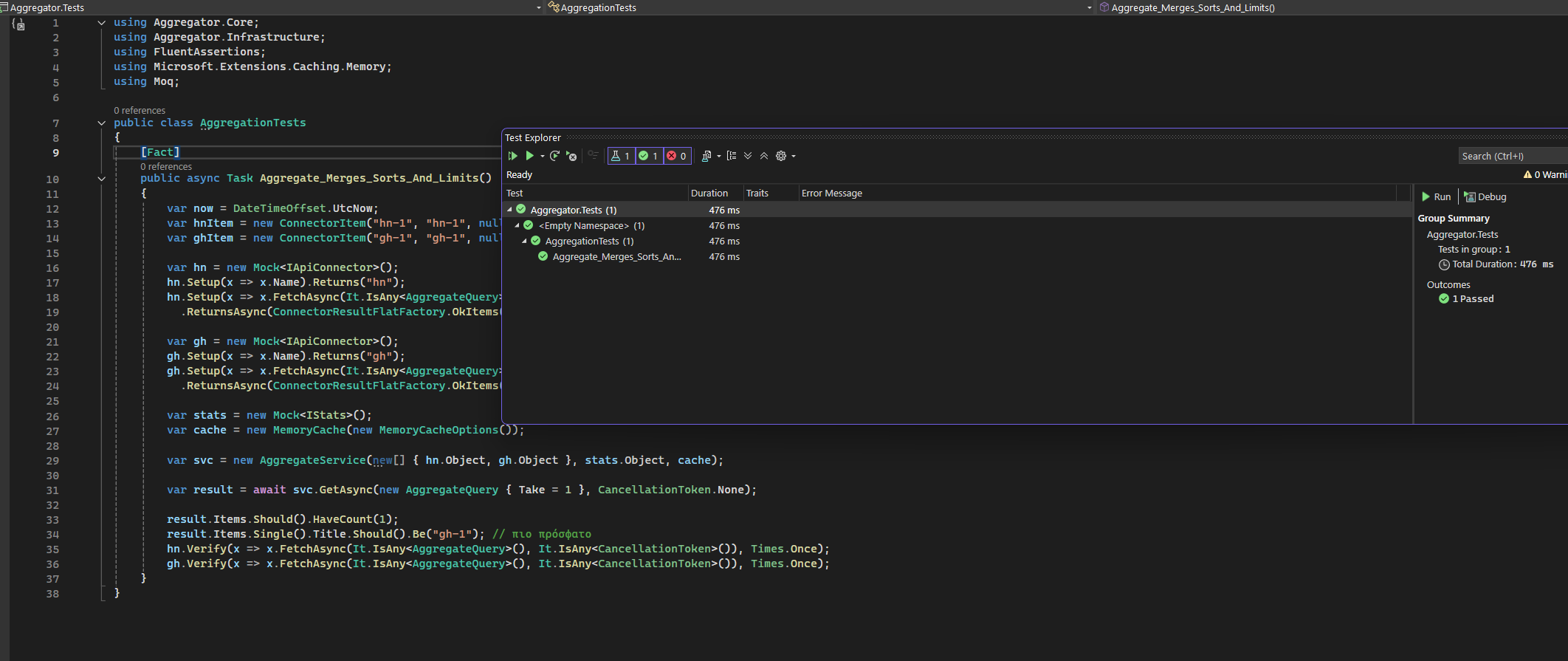
* Health



* From all three API results:



* Run tests:



The end.