**ExpressYourSelf (REST API)**

This document outlines the purpose, objectives, features, and key approaches adopted in the development of the **ExpressYourSelf** project.

**1. Project Objective**

The objective is to create a REST API that adheres to the following criteria:

1. Developed using the latest version of the .NET Core ecosystem.
2. All database interactions must use Entity Framework, except for the task described in section 2.3.
3. The database must be sequential (SQL). Using the provided schema and data is optional.

Below are the main functionalities and tasks of the API.

**2. Task Descriptions**

**2.1 Main Function**

**Description:** An endpoint that receives an IP as input and returns a structured dataset:

* **CountryName**
* **TwoLetterCode**
* **ThreeLetterCode**

**Data Retrieval Order:**

1. Memory cache.
2. Database (if not found in the cache).
3. External public API (IP2C) (if not found in the database).

**Additional Actions:**

* Data obtained from the external API must be stored in the database.
* Queries must be cached to improve performance.

**2.2 Data Update**

**Description:** An automated process that periodically verifies the consistency of the database and performs updates if necessary.

**Rules:**

* Frequency: every hour.
* Verifies in batches of 100 records at a time.
* Reference: IP2C external API, considered reliable.

**Additional Actions:**

* Inconsistent data must be updated in the database and invalidated in the cache.

**2.3 Report Endpoint**

**Description:** An endpoint that provides reports about the database.

**Requirements:**

* **Input:** An array of **TwoLetterCountryCode**. null values return complete data.
* **Output:** An array of structured objects with the following fields:
  + **CountryName** (string)
  + **AddressesCount** (int)
  + **LastAddressUpdate** (DateTime)

**Restrictions:**

* Entity Framework usage is prohibited for this task.

**3. Contextual Decisions**

**3.1 Main Function**

* **Cache Framework:** **IMemoryCache** was chosen for being native to .NET and easy to implement.
* **TTL (Time To Live):** Set to 1 hour, aligning with the update automation interval.
* **TTL Reset:** TTL is automatically reset when data is accessed in the cache.
* **IP2C API Response Format:** The external API returns data in CSV format, requiring appropriate parsing. Since it does not allow customized requests, optimization is limited.
* **Database:** PostgreSQL was adopted, with SQL syntax adjustments made to meet the project’s needs.

**3.2 Data Update**

* **Automation:** Implemented using **HostedService** in ASP.NET for simplicity and efficiency.
* **Assumptions:**
  1. **CountryName**, **TwoLetterCode**, and **ThreeLetterCode** must remain consistent.
  2. IP2C external API is deemed reliable.
* **Comparison Criteria:**
  1. Only **CountryId** will be compared.
  2. New countries will be added to the database automatically, respecting the 50-character limit for names.
* **Cache Handling:** Updated database records will also be updated in the cache to avoid extra costs from removing entries.

**3.3 Report Endpoint**

* **Database Structure:** This endpoint requires related tables, including the **LastAddressUpdate** attribute.
* **Input and Output:** Only null values will return complete data. Input and output types are fixed.
* **Technical Approach:** Raw SQL is mandatory, as per project restrictions.
* **Responses:** The endpoint will notify users if any provided codes do not exist in the database.

**4. Tools, Frameworks, and Techniques**

* **Development Ecosystem:** .NET Core 9.0.101 (latest version).
* **Approach:** Minimal API.
* **Database:** PostgreSQL 15.4.
* **Cache:** IMemoryCache.
* **Automation:** HostedService.
* **Versioning:** Git and GitHub.
* **Request Testing:** Postman.
* **Methodology:** Scrum.

**5. Tests**

**5.1 Request Test**

* Executes an IP request every 2 minutes.
* Utilizes HostedService, randomly selecting an IP from an array of 200 unique values.

**5.2 Endpoint Test**

* Every 15 minutes, attempts to access random endpoints from a predefined list, including:
  + Simple IP lookup.
  + Complete report retrieval.
  + Filtered report retrieval (valid and invalid codes).
  + Invalid IP queries.

**5.3 Update Test**

* Some records in the database have their data manually altered.
* During the next scheduled update, these records must be automatically corrected.

**Note:** For more details about sprints and developed classes, refer to the **Sprints** and **Classes** documents.