GEOTAB

Console application - JokeGenerator

Code refactoring.

Change:.

Installed NuGetmicrosoft.extensions.hosting

Purpose:

Add Dependency Injection, Configuration and Logging ability to the project.

Created DI services container. Move it to the separate Startup file, where we can configure services.

Moved code from Main function to EntryPoint class.

Code remains in Main function is responsible for build DI Service Provider (from Startup file) and run the application (EntryPoint.Run function)

Purpose:

Introduce and configure DI Service Provider. Keep code clean in separate files (following MS standards)

Dependency injection results in maintainable, testable, readable, flexible, and extensible code.

Created folder Services and moved Web APIs calls into it and wrapped them in Typed HttpClient classes. (<https://docs.microsoft.com/en-us/aspnet/core/fundamentals/http-requests?view=aspnetcore-5.0#typed-clients>)

Purpose and benefits:

Provides IntelliSense and compiler help when consuming clients.

Provide a single location to configure and interact with a particular HttpClient. For example, a single typed client might be used:

For a single backend endpoint.

To encapsulate all logic dealing with the endpoint.

Work with DI and can be injected where required in the app.

Used Memory Cache to keep joke categories list in local memory as those are not supposed to change very often

Purpose and benefit.

Reduce external service calls, improve performance.

Analyzed Web API used by application and created Models for Person, Joke, Category in new Models folder

Purpose:

To use typed types instead of “dynamic” or/and clear json string. Also types can be extended in possible future application modernization or migration to Windows or Web application

New folder: Helpers

UI user prompts moved to Constant class.

Simple QueryString builder to be used to help build parameterized requests to Web API.

User Input Validator class to help implement requirements #4

New folder Exceptions contains custom exceptions that used to log real causes of exceptions and

also hide details from the user by substituting with custom user friendly messages for both security and UI/UX prospective.

New NUnit JokeGeneratorTest project.

Contains test classes for Http Client services and Input Validator class.

Purpose:

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

Unit testing finds problems early in the development cycle. This includes both bugs in the programmer's implementation and flaws or missing parts of the specification for the unit.

All service calls are async and wrapped into try-catch blocks using custom exceptions.

Purpose:

To reduce processor load, make applications more responsive during long running external operations.