Joke Generator

Table of Contents

[Projects 3](#_Toc62429410)

[Dependencies 3](#_Toc62429411)

[Interfaces 3](#_Toc62429412)

[Concrete Classes 3](#_Toc62429413)

[Custom Exception 3](#_Toc62429414)

[Application Configuration 3](#_Toc62429415)

[Code Flow 3](#_Toc62429416)

[Logic Engine (JokeApplication.cs -> RunProgram method) 5](#_Toc62429417)

[Changes 6](#_Toc62429418)

# Projects

1. ConsoleApp
2. TestConsoleApp

# Dependencies

|  |
| --- |
| Microsoft.Extensions.Hosting |
| Newtonsoft.Json |
| System.Net.Http |
| Microsoft.Extensions.Logging |
| Xunit |
| Moq |

# Interfaces

1. IJokeApplication
2. IJokeFeed
3. INameFeed
4. IWriter

# Concrete Classes

1. JokeApplication
2. JokeFeed
3. NameFeed
4. Writer

# Custom Exception

1. APIErrorException

# Application Configuration

1. appsettings.json

# Code Flow

1. Program.cs
   * Main class of the application
   * Reads configurable values from appsettings.json ex. API URLs
   * Injects dependencies
     + Injects Writer, JokeFeed, NameFeed dependencies in JokeApplication class
     + Injects HttpClient in JokeFeed and NameFeed
   * Triggers JokeApplication service asynchronously
2. JokeApplication.cs
   * Implements BackgroundService and IJokeApplication interfaces
   * async await pattern used
   * ExecuteAsync method runs “RunProgram” method asynchronously
3. JokeFeed.cs
   * Implements IJokeFeed interface
   * Contains methods to fetch all categories of jokes and get (1-9) jokes from any random category or a category provided by user
4. NameFeed.cs
   * Implements INameFeed interface
   * Contains method to fetch a random name
5. Writer.cs
   * Implements IWriter interface
   * Contains method to print and log messages
6. APIErrorException
   * Custom exception class

# Logic Engine (JokeApplication.cs -> RunProgram method)

Diagram, schematic

Description automatically generated

# Changes

|  |  |
| --- | --- |
| Dependency Injection | Maintainability, Usability, Readability, Loose coupling, Extensibility |
| TAP (Task Based Asynchronous Pattern) | non-blocking way to execute the application. Less CPU usage |
| Exception handling | Reliability |
| appsettings.json for app configuration | Configurability |
| Logger | Debug Issues |
| Made program engine flow user friendly (Flow chart above)   * + Validate user keys   + Prompt user repeatedly until they press correct key   + Option to exit/quit   + Display categories when user wants to enter a category   + Validate user entered category   + Validate number of jokes that can be fetched | UX and Usability |
| Unit testing using Xunit and Moq | Quality (WIP) |