



QUOTE FINDER

Ultimate C# Masterclass Assignment

Overview

Implement the Quote Finder app, which can find a quote containing a given word in an online database.

Console App + API use

Links:

<https://pprathameshmore.github.io/QuoteGarden/> - Quote Garden

documentation (NOTE: there is around 75000 quotes available)

<https://quote-garden.onrender.com/api/v3/quotes?limit=100&page=1> -sample query that gets 1 page of results with max 100 quotes

Main application workflow

When the application starts, it shall ask the user the following:

1. The **word** they want to search. Only quotes containing this word will be finally shown. This should be validated to be a single word, without spaces, numbers, special characters, etc.
2. The **number of pages** of data to check. This number will match the number of quotes shown to the user as an output.
3. The **number of quotes** to be on **each page**. The more quotes, the bigger the chance of finding one containing the required word.
4. **(optional)** Whether the processing of the downloaded data should be performed in **parallel** or not.

Then, the app will **query the Quote Garden API** to get the quotes to be searched. There should be 1 request per page, and the number of quotes on each page should be limited to the value of the third parameter provided by the user.

Once data is downloaded, it should be **processed**. For each page, we search for a single quote containing the given word. The algorithm checking for word presence should be smart and, for example, don't include quotes with word **category** if the user searches for **cat**.

If more than one quote with this word is found on a page, the shortest should be chosen

and printed to the console. If no quote is found on a page, a proper message should also be printed.

Optional requirement:

The processing of the response includes JSON deserialization and the search for a matching word in the collection of quotes. As an optional requirement, we want to let the user choose whether they want this processing to be done in sequence or in parallel, using multithreading. The execution time should be measured so the results can be compared.

Finally, the app prints “Program is finished.” and after the user presses any key, the window closes.

Alternative data source

Since the QuoteGarder is an external service, we don't have control over whether it is available or not. In case it is down, the developer can use the mock data provider called **MockQuotesApiDataReader**. Its source code can be found in the Git repository, as well as in the resources of the “Assignment - Quotes Finder - Description and requirements” lecture.