# SearchRanking Documentation

Version 1.0

Last Updated: 2021-04-24

# How it works?

The design consists of two main operations:

1. Fetch HTML

This step involves requesting for the HTML content from the search engine, storing it into an AWS S3 bucket.

1. Process data fetched from step 1 and return rankings  
   This step processes all of the contents downloaded from step 1 and provide a summary of the processed data.

# Configurations

To see this solution in full action, you will require a set of AWS credentials where information can be found here: <https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html>. Undeniably it is bad that the credentials are currently coded in the library, but to see the solution running will require a working credentials updating in this file: Sympli.Common.AWS.S3Client.cs

# How to do sample runs?

The solution included an API implementation with two GET end points. It should be noted that these two endpoints are designed to work with an event/time triggered system that is not included in this solution (See sections below for more information). However, manual execution can simulate the desired design requirement specified in the section Extension 1 of the requirement document. To simulate, you may execute the following requests using a browser or through the generated Swagger API panel (http://localhost:{port}/swagger):

Execute the following requests with a minimum of 10 times:  
/api/SearchRanking/Fetch?SearchKeywords=e-settlements&SearchEngineType=Google   
  
To retrieve the results:  
/api/SearchRanking/Results?SearchEngine=Google&Keywords=e-settlements&TargetDate=2021-04-24&Match=sympli.com.au&ShowDetails=false

A couple of things to note here:

* When performing fetch, the data fetch will be recorded as “today”. Note that each fetch through Google only returns 10 results, the application is currently designed to only return results when there are more than 100 results found, hence the minimum of 10 executions of the Fetch request.
* Each execution of the Fetch request will fetch the next set of results according to what has already been “fetched”, in this case, it is determined by what is in the S3 bucket.
* Retrieving results will require a date, if no date is specified, “today” will be used as default.
* Retrieving results keywords will need to match the keywords used in the fetch. Otherwise no results will be available.
* Retrieving results will require at least 100 results to be present from the amount of Fetch requests executed. Otherwise no result will be available.
* Retrieving results does not require to have a “Match” parameter, if none provided, all results will be returned
* Retrieving results using “ShowDetails” as true will present the information of the URL detected from the Fetch results.

# TODOs

Due to time constraints, the following should be noted:

1. Move credentials for S3 client to proper storage (Sympli.Common.AWS.S3Client.cs) in accordance with AWS documentation
2. Bing implementations are for IoC demonstration purposes only and to fulfill the design requirement of Extension 2, full implementation for Bing has not been completed.
3. API response should be more standardised along with better handled error messages.
4. More error handlings should be considered.
5. More code comments should be added or hire me to add the comments for you

# Further Suggestions

Since the CEO is interested in the SEO rankings, just by seeing the rankings going up and down isn’t too much helpful. It can be suggested to add the functionality of fetching the HTML of each of the ranked pages and subsequently using cloud services to compare all of the HTML data to provide a much more in-depth analysis of “how did a site went up or down in the SEO ladder?” and “What was changed to the pages between dates?” which in turn will be much more valuable to the CEO than just a ranking.