Table of Contents

[Problem description: 1](#_Toc5291324)

[Architect solution (simplified) 1](#_Toc5291325)

[Preview 1](#_Toc5291326)

[Software used 1](#_Toc5291327)

[basic setup 1](#_Toc5291328)

[Missing features 2](#_Toc5291329)

[Enhancements 2](#_Toc5291330)

[Copyright notice 2](#_Toc5291331)

## Problem description:

Create an API that takes a string as input and returns a list of text insights from the top 5 google results of the query generated by those keywords.

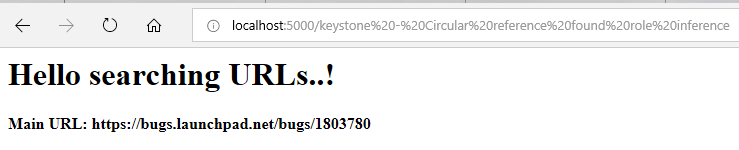
## Architect solution (simplified)

Flask API + web scraping using python BeautifulSoap

Ref: <https://github.com/miguelluiz/vanhack-quiz-C/blob/master/Simplified%20architecture.pptx>

An API has been provided using Flask on [http://localhost:5000/<url](http://localhost:5000/%3curl)>

## Preview



## Software used

* It has been tested against Microsoft Edge and Google Chrome

Virtual Environment under Windows 10 (64-bit) using **Python 3.7.3**



Required packages (included in requirements.txt)

beautifulsoup4==4.7.1

bs4==0.0.1

builtwith==1.3.3

Flask==1.0.2

Flask-Compress==1.4.0

google==2.0.2

Jinja2==2.10

requests==2.21.0

soupsieve==1.9

## basic setup

Make sure Python version is 3.7 or higher, then:

* Download or clone from [GitHub](https://github.com/miguelluiz/vanhack-quiz-C), then run the following command:
* pip install -r requirements.txt

Note that the google package has been used in order to **search top 5 results** as shown below:

from googlesearch import search

def getTopUrls(url, n):

'''

Returns google search after reaching 'n' urls

'''

\_urls=[]

for url in search(url, stop=n):

\_urls.append(url)

return \_urls

## Missing features

* Missing getting source code and images for a given URL  
  If the webpage has any source code and/or any image that might help the customer it will not show at the results.

## Enhancements

* Layout needs improvements  
  The given layout does a very basic task, therefore it has room to be improved.   
  Ex: We do provide lots of links, so it would be worth to limit the number the links and add a bottom to help the user to decide whether he or she wants to see or not  
  Ex: Links could also be a real link and it is presented just as a text
* Although BeautifulSoup can do most of the tasks at daily bases, Scrapy allow the solution to go even further

Copyright notice

* Since there are sites that have strong restrictions against scrapping, in a real solution, this issue should be taken care