|  |  |
| --- | --- |
| **Title**: | **Microservice 01**: Pull a web page |
| **Description** | Microservice should pull a whole web page while pretending to be a browser |
| **Input Format 1** | JSON |
| **Input schema 1** | {Input:  [  {url: <url>}  ]  } |
| **Output Format 1** | JSON |
| **Output Schema 1** | {Output:  [  {url: <url>},  {error:  [  errorno: <number>,  errormessage: <message>  ]  {data: <html> }  ]  } |
|  |  |
| **Input Format 2** | REST API |
| **Input schema 2** | url |
| **Output Format 2** | JSON via REST API return |
| **Output Schema 2** | Same as Output Schema 1 |
|  |  |
| **Language** | Python 3 |
| **Input channel(s)** | Message queue +  API call |
| **Test Harness** | A Django form to test this |
| **Container** | Docker |
| **References** | Escaping HTML code in json: https://stackoverflow.com/questions/963448/html-tags-within-json-in-python  Request like a browser:  https://stackoverflow.com/questions/27652543/how-to-use-python-requests-to-fake-a-browser-visit |
| **Logging** | Use Django logging:  https://docs.djangoproject.com/en/2.0/topics/logging/ |
| **Configuration** | Store all config values in a .env file. No hard-coding!!  Python Decoupler:  https://pypi.org/project/python-decouple/ |
| **Framework** | Use Django and have a webpage where you can enter a URL and then the result will be shown in a textarea box  Also, use Django url to call the API |
| **Testing** | Use Django unit test |
| **Docker image** | For the docker build file image – input parameters should be:   1. Port for showing the Django page |

|  |  |  |
| --- | --- | --- |
| **Test case** | **Input** | **Expected Output** |
| 1 | User goes to webpage, enters a url of an existing page | Return json with url, and html data. HTML data needs to be escaped. |
| 2 | User goes to webpage, enters a url of a non-existnat page | Return json with url, and errorno: MS0101  Error message: url does not exist |
| 3 | Call REST API directly on the microservice with an existing url | Return json with url, and html data. HTML data needs to be escaped. |