**Technologies:**

* Nodejs
* Express.js
* Dockers Containers
* RabitMQ Messaging & Queuing, Producer and Consumer based architecture.

**Working:**

This Service uses RabitMQ to queue all the post requests and then there are 2 agents that work with this queue to complete the screenshot process

1. **Producer (web-Service.js) :** Receives the requests and pushes it to the queue while lets user know about the request id so that they can query the data in future
2. **Consumer (Process-Service):** takes each request from the queue and takes screenshot, saves it, updates the database and then acknowledges the Producer.

**Pre-Requisites:**

* Docker Engine must be installed on the computer.
* Node V. 10.16.0

**Installation Steps:**

* Run Run\_Dockers.bat ----- (Runs Dockers images of MySQL, RabitMQ & set up SQL DB)
* Run Run\_Producer.bat ---- ( Runs the Producer Services which listen to requests)
* Run Run\_Consumer.bat---- (Run Consumer Service that takes screenshots and stores data)

**Interface:** POSTMAN can be used as a front-end and to hit the API endpoints

**End Points:**

* [POST] <http://localhost:3000/api/v1/screenshot>

**input** : Json Array of URLs in Body.

{

"url": [

"https://facebook.com",

"https://google.com"

]

}

**Output:** Json Response with status and Request ID which will be used to query data.

{

"status": "success",

"requestId": "201910242151268"

}

* [GET] [http://localhost:3000/api/v1/getScreenshot/{RequestID}](http://localhost:3000/api/v1/getScreenshot/%7bRequestID%7d)

**Input:** Takes Request ID as input and returns the links to Screenshot Images.

**Output:**  a json Array containing URLs and link to the png image of the screenshot

[

{

"url": "https://facebook.com",

"link": "http://localhost:3000/201910242151268/201910242151268-1.png",

"requestId": "201910242151268"

},

{

"url": "https://google.com",

"link": "http://localhost:3000/201910242151268/201910242151268-2.png",

"requestId": "201910242151268"

}

]

**Screenshots:**







