

Image Converter

An image converter converts image having tables to excel file

By Tejesh Agrawal
B00804661
Graduating May 2021

INDEX

Sr. No.	Particulars	Page No.
1	Introduction	3
2	Technologies Used	3
3	Users	3
4	Install Tools	3
5	Folder Structure	4
6	Backend	4
7	Frontend	7
8	Screenshots of Website	13
9	Run Locally	25
10	Future Scope	26
11	URLs	26
12	References	26

1.INTRODUCTION

This project is about converting image having tables to an excel files. It is a web application named = "" using Eureka Service discovery as a discovery server to detect microservices developed in Java Spring boot, python flask, and ReactJS.

This web application has following features:

1. Login using OAuth 2.0
2. Uploading an Image
3. Process the uploaded image
4. Display the converted excel sheet
5. Download the converted excel sheet

2.TECHNOLOGIES USED

Eureka service discover server - Discovery server developed by Netflix OSS using Spring boot framework.

Spring boot application framework - To build REST a microservice for calling other microservices endpoints.

Flask – To host python machine learning code, on REST endpoints.

ReactJS – For frontend, user can login, upload an image and then display the converted Excel file.

Auth0 – User can login using google credentials.

3. USER

User – User will have to login using Google credentials, then user can upload image, convert it into excel and then display, download the excel file.

4. INSTALL TOOLS

Code Editor – VS Code, Atom to write code and run it on

Web Browser – Google Chrome, OSX Safari

Languages – JAVA SE 1.8, python 3(using virtual environment), JavaScript for frontend

NPM – To install Node dependencies, ReactJS and NodeJS

POSTMAN – platform for API development, to test POST, GET API endpoints to check

5. FOLDER STRUCTURE

Discovery server – Discovers all the backend microservices (discovery-server)

Microservices – Spring boot application (catalog-service), flask backend to process image (table)

Frontend – ReactJS application, contains all the JS, CSS, HTML code for user interface (frontend)

bu-final-project				
Name	^	Date Modified	Size	Kind
__pycache__		Apr 25, 2021 at 12:22 PM	--	Folder
app.py		May 5, 2021 at 5:43 PM	4 KB	Python Script
catalog-service		May 14, 2021 at 2:29 AM	--	Folder
discovery-server		Apr 21, 2021 at 7:02 PM	--	Folder
dns.py		Apr 25, 2021 at 12:22 PM	189 bytes	Python Script
dnspython		Apr 25, 2021 at 12:18 PM	--	Folder
frontEnd		May 8, 2021 at 5:12 PM	--	Folder
table		May 14, 2021 at 12:25 AM	--	Folder
trash		May 9, 2021 at 4:18 PM	--	Folder

The table folder having **app.py**, it is flask backend microservice.

table	
Name	Date Modified
> __pycache__	May 5, 2021 at 5:47 PM
app.py	May 14, 2021 at 12:32 AM
cv_inverted.png	May 9, 2021 at 2:17 PM
> env	Apr 29, 2021 at 5:04 PM
horizontal.jpg	May 9, 2021 at 2:17 PM
img_vh.jpg	May 9, 2021 at 2:17 PM
requirements.txt	May 5, 2021 at 5:46 PM
table_new.py	May 5, 2021 at 5:45 PM
table.py	May 5, 2021 at 5:44 PM
testcv.png	Apr 27, 2021 at 10:06 PM
Untitled.ipynb	Apr 29, 2021 at 6:12 PM
> uploads	May 5, 2021 at 12:47 AM
vertical.jpg	May 9, 2021 at 2:17 PM
writer.xlsx	Today at 6:24 PM

6. What, why, how?

What?

This project is about converting images to an excel files.

Why?

In real world, if we talk about places like warehouses, labs etc. data entries are still done manually, to transform those entries into computer readable excel files this tool will be helpful.

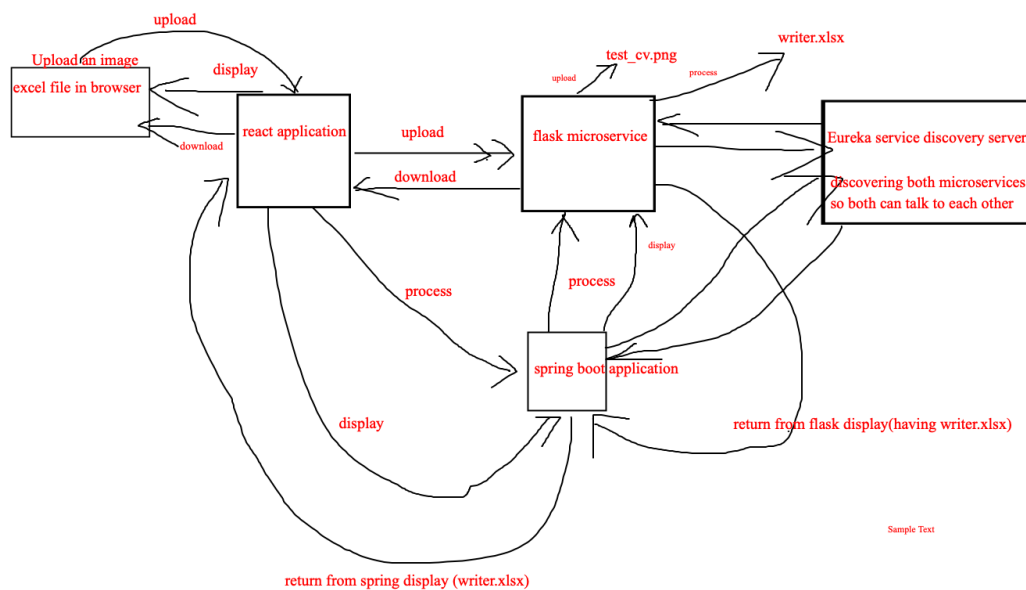
How?

By using machine learning algorithms, libraries (pytesseract, OpenCV) to transform an image to an Excel file.

ReactJS for user interface to upload file, process file and display file.

Flask and Spring Boot to enable microservices to perform REST operations.


Eureka service discovery server to help communicate microservices to each other.



- Above arrows REST api calls

7. Service Discovery Server

mvnw	Jan 26, 2019 at 3:11 PM	9 KB	Unix Ex...ab
mvnw.cmd	Jan 26, 2019 at 3:11 PM	6 KB	Document
pom.xml	Jan 26, 2019 at 3:11 PM	2 KB	XML Text
src	May 7, 2021 at 11:25 PM	--	Folder
eureka-curl-payload.json	May 7, 2021 at 11:25 PM	449 bytes	JSON file
main	May 7, 2021 at 11:25 PM	--	Folder
java	Apr 25, 2021 at 11:57 PM	--	Folder
io	Apr 25, 2021 at 11:57 PM	--	Folder
resources	Apr 25, 2021 at 12:05 PM	--	Folder
application.properties	Apr 26, 2021 at 5:45 PM	127 bytes	Java pr...rti
application.yml	May 7, 2021 at 11:32 PM	207 bytes	YAML docu
test	Jan 26, 2019 at 3:11 PM	--	Folder
java	Jan 26, 2019 at 3:11 PM	--	Folder
target	Apr 21, 2021 at 6:54 PM	--	Folder
classes	May 9, 2021 at 4:23 PM	--	Folder
test-classes	May 9, 2021 at 4:23 PM	--	Folder

HOME LAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2021-05-16T18:22:34 -0400
Data center	default	Uptime	1 day 04:05
		Lease expiration enabled	false
		Renews threshold	5
		Renews (last min)	4

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
CATALOG-SERVICE	n/a (1)	(1)	UP (1) - tejeshs-air:catalog-service:8085
PYTHON-MODULE-1	n/a (1)	(1)	UP (1) - 127.0.0.1:python-module-1:5000

General Info

Name	Value
total-avail-memory	154mb
environment	test
num-of-cpus	4
current-memory-usage	76mb (49%)
server-uptime	1 day 04:05
registered-replicas	

8. Microservices

Spring boot REST backend:

Name	Date Modified	Size
mvnw	Jan 26, 2019 at 3:11 PM	
mvnw.cmd	Jan 26, 2019 at 3:11 PM	
pom.xml	Apr 26, 2021 at 7:22 PM	
src	Apr 25, 2021 at 11:57 PM	
main	Apr 25, 2021 at 11:57 PM	
java	Apr 25, 2021 at 11:57 PM	
resources	Jan 26, 2019 at 3:11 PM	
test	Jan 26, 2019 at 3:11 PM	
target	Today at 11:00 PM	
classes	Today at 11:00 PM	
application.properties	Today at 11:00 PM	
io	Today at 11:00 PM	
dependencies.txt	Apr 26, 2021 at 7:17 PM	
test-classes	May 9, 2021 at 4:26 PM	
io	May 9, 2021 at 4:26 PM	
writer.xlsx	May 14, 2021 at 2:29 AM	

The screenshot displays an IDE with the following components:

- EXPLORER:** Shows the project structure for 'catalog-service', including 'src/main/java/io/javabrain/moviedatacatalogservice' and 'src/main/resources/application.properties'.
- EDITOR:** Displays the source code for 'CatalogServiceApplication.java'. The code includes imports for Spring Boot, Eureka, and RestTemplate, and defines a 'main' method and a 'getRestTemplate' method.
- OUTPUT/DEBUG CONSOLE:** Shows the execution logs of the application. The logs indicate that the application is running successfully and resolving Eureka endpoints via configuration.

```
package io.javabrain.moviedatacatalogservice;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.web.client.RestTemplateCustomizer;
import org.springframework.cloud.client.loadbalancer.LoadBalanced;
import org.springframework.context.annotation.Bean;
import org.springframework.http.client.HttpComponentsClientHttpRequestFactory;
import org.springframework.web.client.RestTemplate;

@SpringBootApplication
@EnableEurekaClient
public class CatalogServiceApplication {
















    public static void main(String[] args) {
        SpringApplication.run(CatalogServiceApplication.class, args);
    }

    @LoadBalanced
    @Bean
    public RestTemplate getRestTemplate() {
        return new RestTemplate();
    }

    // @LoadBalanced
    // @Bean
    // public RestTemplateCustomizer readTimeoutCustomizer() {
    //     //RestTemplate restTemplate = new RestTemplate();
    //     return restTemplate -> {
    //         HttpComponentsClientHttpRequestFactory clientRequestFactory = new HttpComponentsClientHttpRequestFactory();
    //         clientRequestFactory.setReadTimeout(12345);
    //         restTemplate.setRequestFactory(clientRequestFactory);
    //     };
    // }
}
```

2021-05-16 23:02:27.510 INFO 82740 [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2021-05-16 23:07:27.644 INFO 82740 [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2021-05-16 23:12:27.657 INFO 82740 [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2021-05-16 23:35:04.159 INFO 82740 [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2021-05-16 23:40:04.180 INFO 82740 [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2021-05-16 23:45:04.196 INFO 82740 [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2021-05-16 23:58:04.213 INFO 82740 [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2021-05-16 23:59:04.279 INFO 82740 [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2021-05-17 00:00:04.294 INFO 82740 [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration

Flask ML backend microservice:

table			
Name		Date Modified	
>  __pycache__		May 5, 2021 at 5:47 PM	
 app.py		May 14, 2021 at 12:32 AM	
 cv_inverted.png		May 9, 2021 at 2:17 PM	
>  env		Apr 29, 2021 at 5:04 PM	
 horizontal.jpg		May 9, 2021 at 2:17 PM	
 img_vh.jpg		May 9, 2021 at 2:17 PM	
 requirements.txt		May 5, 2021 at 5:46 PM	
 table_new.py		May 5, 2021 at 5:45 PM	
 table.py		May 5, 2021 at 5:44 PM	
 testcv.png		Apr 27, 2021 at 10:06 PM	
 Untitled.ipynb		Apr 29, 2021 at 6:12 PM	
✓  uploads		May 5, 2021 at 12:47 AM	
>  test_docs		May 9, 2021 at 2:32 PM	
 vertical.jpg		May 9, 2021 at 2:17 PM	
 writer.xlsx		Today at 6:24 PM	

```

(env) Tejeshs-Air:table tejeshagrawal$ python3 app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:27:09] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:28:23] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:31:28] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:32:21] "GET /display HTTP/1.1" 200 -
^CTraceback (most recent call last):
  File "app.py", line 353, in <module>
    flask_cors.CORS(app, expose_headers='Authorization')
NameError: name 'flask_cors' is not defined
(env) Tejeshs-Air:table tejeshagrawal$ python3 app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:33:18] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:34:24] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:34:35] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:35:09] "GET /object%20object HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:35:30] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:36:14] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:37:35] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 00:43:18] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 02:24:09] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 02:24:54] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 02:26:03] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 02:26:19] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 02:29:54] "GET /display HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 02:33:02] "GET /download HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 02:33:58] "GET /download HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 02:38:37] "GET /download HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 02:41:40] "GET /download HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 02:43:23] "GET /download HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [14/May/2021 02:48:55] "GET /download HTTP/1.1" 200 -

```















9. FRONTEND

TECHNOLOGICAL COMPONENTS USED

React - Components, Props, Events, Hooks, Router

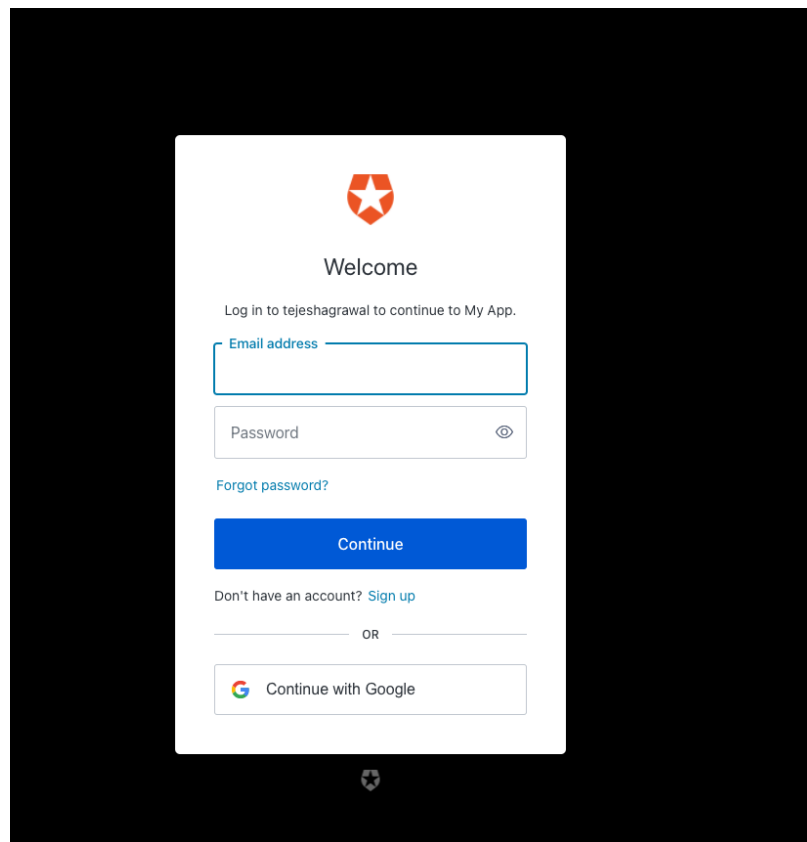
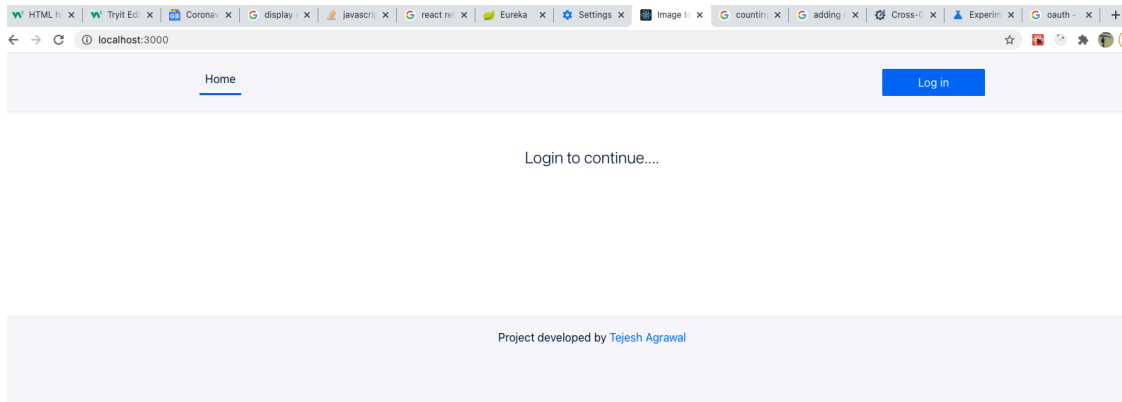
Axios - library to send ajax request to backend microservices server

Auth0 – To login using user's google credentials

frontEnd		
Name		Date Modified
 api-server.js		Apr 13, 2021 at 8:52 AM
 Dockerfile		Apr 13, 2021 at 8:52 AM
 exec.ps1		Apr 13, 2021 at 8:52 AM
 exec.sh		Apr 13, 2021 at 8:52 AM
>  node_modules		May 11, 2021 at 2:38 AM
 package-lock.json		May 10, 2021 at 7:59 PM
 package.json		May 10, 2021 at 8:59 PM
>  public		Apr 13, 2021 at 8:52 AM
 README.md		Apr 13, 2021 at 8:52 AM
 server.js		Apr 13, 2021 at 8:52 AM
>  src		May 8, 2021 at 5:12 PM
 webpack.config.js		May 8, 2021 at 5:12 PM
 yarn.lock		May 10, 2021 at 7:59 PM

10. SCREENSHOTS OF WEBSITE

Login Page:



Home Page:

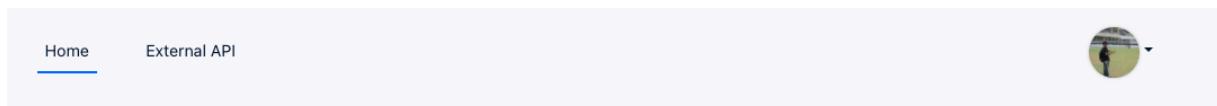


Image to Excel table conversion, click to upload file...

Choose File No file chosen

Enter the desired name o

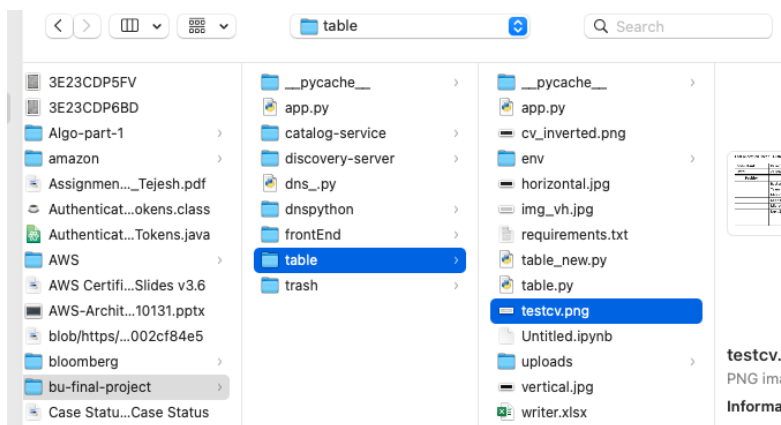
Upload...

Process...

This is a web application that converts an image to an excel file, using Eureka server as a discovery server for microservices written in Sprint boot(Java), flask (python3), ReactJS (Javascript) and user authentication using Auth0.

Display... Download...

Choosing file from file directory after clicking on **Choose File** button,



Click on upload button to upload file in server backend folder.

Choose File testcv.png
Enter the desired name o
Upload...

Click on process button:

**Image to Excel table
conversion, click to upload
file...**

Choose File testcv.png
Enter the desired name o
Upload...
Process...
Remaining
54
seconds

Below screenshot gives the api 200 call, success call:

```
INFO:HELLO WORLD:welcome to upload`  
INFO:werkzeug:127.0.0.1 - - [17/May/2021 00:34:19] "POST /upload HTTP/1.1" 200 -  
INFO:werkzeug:127.0.0.1 - - [17/May/2021 00:40:18] "GET /process HTTP/1.1" 200 -  
□
```

Clicking on display button:

This is a web application that converts an image to an excel file, using Eureka server as a discovery server for microservices written in Sprint boot(Java), flask (python3), ReactJS (Javascript) and user authentication using Auth0.

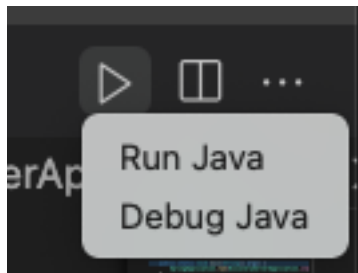
Display...		file.xlsx		Download...	
A	B	C	D		E
0		0	1		2
1	0	Consultant:	I Project: [Consultant: Kaliro Siduco [Tech Alive Date: 21 May 2020 Customer-ID: 443228-XY [Position [Service [Factor [Amount [Unt [[— [Backend Support ours { [_ — s*([TeamMeetings — — ae st— ("(*zs)SOW!w!!!!U!U!U!UdhLmLmLU!UmrIU 8 fours {eres Ee Migration — — \$= l cL [30 hous l Next Steps Project: Kaliro Siduco		XGTR
2	1	Date:	Team: 21 May 2020		
3	2	Position	Service		
4	3		Backend Support		
5	4		Team Meetings		
6	5		Milestone Presentation		
7	6		Code Refactoring		
8	7		Migration		
9	8		Next Steps		
10	9				
11	10				Total

Click on download button, downloads the file:



11. RUN LOCALLY

Starting the Eureka Service discovery server on port 8761:



```
2021-05-17 01:49:28.920 INFO 42071 --- [Thread-12] c.n.e.r.PeerAwareInstanceRegistryImpl : Renew threshold is: 1
2021-05-17 01:49:28.920 INFO 42071 --- [Thread-12] c.n.e.r.PeerAwareInstanceRegistryImpl : Changing status to UP
2021-05-17 01:49:28.957 INFO 42071 --- [Thread-12] e.s.EurekaServerInitializerConfiguration : Started Eureka Server
2021-05-17 01:49:29.126 INFO 42071 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8761 (http) with context path '/'
2021-05-17 01:49:29.127 INFO 42071 --- [main] .s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 8761
2021-05-17 01:49:29.138 INFO 42071 --- [main] i.j.d.DiscoveryServerApplication : Started DiscoveryServerApplication in 19.782 seconds (JVM running for 23.567)
```

Starting flask microservice:

We are using python3 to run our microservice, creating virtual environment:

```
$ python3 -m venv env
```

Activate virtual environment:

```
[Tejeshs-Air:table tejeshagrawal$ . env/bin/activate
(env) Tejeshs-Air:table tejeshagrawal$
```

Running pip3.7 to install dependencies:

```
((env) Tejeshs-Air:table tejeshagrawal$ ls
Untitled.ipynb      app.py              env                  img_vh.jpg          table.py            testcv.png          vertical.jpg
__pycache__         cv_inverted.png    horizontal.jpg       requirements.txt     table_new.py        uploads             writer.xlsx
((env) Tejeshs-Air:table tejeshagrawal$ pip3.7 install -r requirement.txt
ERROR: Could not open requirements file: [Errno 2] No such file or directory: 'requirement.txt'
WARNING: You are using pip version 20.1.1; however, version 21.1.1 is available.
You should consider upgrading via the '/Users/tejeshagrawal/Desktop/bu-final-project/table/env/bin/python3 -m pip install --upgrade pip' command.
((env) Tejeshs-Air:table tejeshagrawal$ pip3.7 install -r requirements.txt
Requirement already satisfied: editdistance==0.5.2 in ./env/lib/python3.7/site-packages (from -r requirements.txt (line 1)) (0.5.2)
Requirement already satisfied: openpyxl in ./env/lib/python3.7/site-packages (from -r requirements.txt (line 2)) (3.0.7)
Requirement already satisfied: Jinja2 in ./env/lib/python3.7/site-packages (from -r requirements.txt (line 3)) (2.11.3)
Requirement already satisfied: pandas in ./env/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (1.2.4)
Requirement already satisfied: Image in ./env/lib/python3.7/site-packages (from -r requirements.txt (line 5)) (1.5.33)
Requirement already satisfied: pytesseract in ./env/lib/python3.7/site-packages (from -r requirements.txt (line 6)) (0.3.7)
Requirement already satisfied: lmbd==1.0.0 in ./env/lib/python3.7/site-packages (from -r requirements.txt (line 7)) (1.0.0)
Requirement already satisfied: matplotlib==3.2.1 in ./env/lib/python3.7/site-packages (from -r requirements.txt (line 8)) (3.2.1)
Requirement already satisfied: numpy==1.19.5 in ./env/lib/python3.7/site-packages (from -r requirements.txt (line 9)) (1.19.5)
```

```
[(env) Tejeshs-Air:table tejeshagrawal$ cat requirements.txt
editdistance==0.5.2
openpyxl
jinja2
pandas
Image
pytesseract
lmbd==1.0.0
matplotlib==3.2.1
numpy==1.19.5
opencv-python==4.4.0.46
path==15.0.0
tensorflow
XlsxWriter
flask_cors
flask_eureka
Flask
py_eureka_client
```

Running the flask microservice:

```
[(env) Tejeshs-Air:table tejeshagrawal$ python3 app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

It registers on Eureka service discovery server:

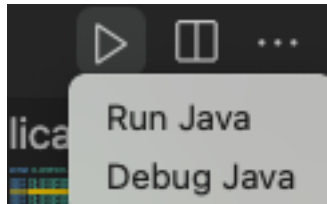
```
2021-05-17 02:07:56.207 INFO 42767 [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 0ms
2021-05-17 02:07:56.212 INFO 42767 [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 4ms
2021-05-17 02:07:10.268 INFO 42767 [nio-8761-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2021-05-17 02:07:10.268 INFO 42767 [nio-8761-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2021-05-17 02:07:10.294 INFO 42767 [nio-8761-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 26 ms
2021-05-17 02:07:10.495 INFO 42767 [nio-8761-exec-1] c.n.e.registry.AbstractInstanceRegistry : Registered instance PYTHON-MODULE-1/127.0.0.1:python-module-1:5000 with status UP (replication=false)
2021-05-17 02:07:56.215 INFO 42767 [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 1ms
```

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
PYTHON-MODULE-1	n/a (1)	(1)	UP (1) - 127.0.0.1:python-module-1:5000

Starting Spring boot microservice:



```
2021-05-17 02:11:39.232 INFO 43057 --- [InfoReplicator-0] com.netflix.discovery.DiscoveryClient : DiscoveryClient_CATALOG-SERVICE/tejeshs-air:catalog-service:8085: registering service...
2021-05-17 02:11:39.302 INFO 43057 --- [InfoReplicator-0] com.netflix.discovery.DiscoveryClient : DiscoveryClient_CATALOG-SERVICE/tejeshs-air:catalog-service:8085 - registration status: 204
2021-05-17 02:11:39.375 INFO 43057 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8085 (http) with context path '/'
2021-05-17 02:11:39.376 INFO 43057 --- [main] .s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 8085
2021-05-17 02:11:39.380 INFO 43057 --- [main] i.j.m.CatalogServiceApplication : Started CatalogServiceApplication in 14.349 seconds (JVM running for 16.647)
```

Now we have both applications running and registered on Eureka server:

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
CATALOG-SERVICE	n/a (1)	(1)	UP (1) - tejeshs-air:catalog-service:8085
PYTHON-MODULE-1	n/a (1)	(1)	UP (1) - 127.0.0.1:python-module-1:5000

General Info

```
2021-05-17 02:07:56.215 INFO 42767 --- [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 1ms
2021-05-17 02:08:56.217 INFO 42767 --- [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 0ms
2021-05-17 02:09:56.219 INFO 42767 --- [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 1ms
2021-05-17 02:10:56.266 INFO 42767 --- [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 2ms
2021-05-17 02:11:39.300 INFO 42767 --- [nio-8761-exec-4] c.n.e.registry.AbstractInstanceRegistry : Registered instance CATALOG-SERVICE/tejeshs-air:catalog-service:8085 with status UP (replication=false)
2021-05-17 02:11:56.282 INFO 42767 --- [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 7ms
2021-05-17 02:12:56.287 INFO 42767 --- [a-EvictionTimer] c.n.e.registry.AbstractInstanceRegistry : Running the evict task with compensationTime 2ms
```

Starting front end:

Installing dependencies:

```
Tejeshs-Air:frontEnd tejeshagrawal$ npm install

up to date, audited 1932 packages in 12s

1 package is looking for funding
  run `npm fund` for details

100 vulnerabilities (4 low, 83 moderate, 13 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible (including breaking changes), run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
Tejeshs-Air:frontEnd tejeshagrawal$
```


Starting the frontend server:

```
[Tejeshs-Air:frontEnd tejeshagrawal$ npm start

> auth0-react-sample@0.1.0 start
> npm-run-all --parallel spa api-server

> auth0-react-sample@0.1.0 api-server
> node api-server.js

> auth0-react-sample@0.1.0 spa
> react-scripts start

Exiting: Please make sure that auth_config.json is in place and populated with valid domain and audience values
```

Server is now started:

```
Compiled successfully!

You can now view image-excel-converter in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.1.239:3000

Note that the development build is not optimized.
To create a production build, use yarn build.
```

12. FUTURE SCOPE

- Use Apache Kafka as messaging server for handling multiple users.
- Use Cache service to process request faster.
- Implement it on AWS, on AWS use S3 for folder structure, Autoscaling and load balancers for more availability and efficiency.
- Right now, it is as a web application. Need to make a mobile web application having camera access feature, so user can take real time photos.
- Need to implement handwriting recognition, so users can convert even handwriting tabular data to an excel sheet.
- Enable proxy gateway server for more security in transition.

13. URLs

- GitHub :: <https://github.com/TEJESH/bu-final-tejesh>

14. REFERENCES

Image to excel, python ML using OpenCV and pytesseract

<https://towardsdatascience.com/a-table-detection-cell-recognition-and-text-extraction-algorithm-to-convert-tables-to-excel-files-902edcf289ec>

Eureka server, spring boot tutorial:

<https://youtu.be/y8lQb4ofjDo>

Auth0:

<https://manage.auth0.com/dashboard/us/tejeshagrawal/applications/Gg4jpbbyqw1mgLTtockxsXr2ZyecXNNVz/settings>

Python Eureka client:

<https://github.com/keijack/python-eureka-client>

Creating virtual environment for python:

<https://medium.com/@tejeshagrawal/why-using-virtual-environment-for-python-project-is-good-for-system-and-project-4196476213d6>

Flask with Machine learning model:

<https://towardsdatascience.com/how-to-easily-deploy-machine-learning-models-using-flask-b95af8fe34d4>