RETURN ORDER MANAGEMENT SYSTEM

* Project Members
* Mrunal Yogesh Patil
* Abhishek Arvind Gandhi
* Satyam Chaudhary
* M Harivivek
* Project Overview
* A leading Supply chain Management Organization wants to automate the return orders, by classifying them to repair or replacement. Repair is for all main or integral part of their product. Replacement is for accessories.
* Project Details
* Microservices
  1. Authorization

The Authorization microservice is used to create JWT tokens. It used for authorization of other microservices.

* 1. Component Processing

The intent of this Microservice is to determine the Component processing detail. It interacts with packaging and delivery microservice to get the consolidated cost for the processing.

* 1. Packaging and Delivery

The Packaging and Delivery microservice should get the component type and count to determine the packaging and delivery charge.

* 1. Payment

The Payment microservice gets the credit card details of the user and the total processing charge. Deducts the processing charge from the current amount and returns the balance amount.

* 1. Return Order Portal

The Return Order Portal microservice consists of all jsp pages which are connected to various other microservices.

* Port Number
  1. Authorization – 8084
  2. Packaging and Delivery – 8083
  3. Payment – 8082
  4. Component Processing – 8081
  5. Return Order Portal – 8080
  6. Discovery Server(Eureka) – 8085
* Amazon Web Services(AWS) – Elastic Beanstalk
  + All the Microservices are deployed on AWS Elastic Beanstalk.
  + AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.
  + For more information about AWS Beanstalk, refer the following link :- <https://aws.amazon.com/elasticbeanstalk/>
* AWS Elastic Beanstalk deployed microservices links
  + Authorization Microservice :-

<http://returnorderauth.ap-south-1.elasticbeanstalk.com:8084/>

* + Packaging and Delivery Microservice :-

<http://returnorderpackagedelivery.ap-south-1.elasticbeanstalk.com:8083/>

* + Payment Microservice :-

<http://returnorderpayment.ap-south-1.elasticbeanstalk.com:8082/>

* + Component Processing Microservice :-

<http://returnordercomponentprocessing.ap-south-1.elasticbeanstalk.com:8081/>

* + Return Order Portal :-

<http://localhost:8080/>

(These services must be consumed from an MVC app running in a local environment)

* Database
  + All the microservices are independently deployed.
  + An In-memory database (H2 database) has been used in the application.
* H2 Console links

1. <http://localhost:8080/h2-console>

Tables

1. PROCESS\_REQUEST
2. PROCESS\_RESPONSE
3. USER\_DETAILS
4. <http://localhost:8081/h2-console>

Tables

1. PAYMENT
2. PROCESS\_REQUEST
3. PROCESS\_RESPONSE
4. <http://localhost:8082/h2-console>

Tables

1. CREDIT\_CARD

* SONAR
  + SonarLint extension for Eclipse IDE has been used for fixing coding issues.
  + Like a spell checker, SonarLint highlights Bugs and Security Vulnerabilities as you write code, with clear remediation guidance so you can fix them before the code is even committed.