Titrate Technical Specification

Document Control

Revision History

Version	Date	Updated By	Description
V1.0	11-02-2024	Jyoshna	Initial Version
V1.1	08-03-2024	Jyoshna	Updated Annotation Flow
V1.2	14-03-2024	Jyoshna	Removed confounding objects flow

Document Approval

Date	Name	Role	Status
	Manohar	Manager	In Progress

Table of Contents

Objective

The following document covers the design for a specific business requirement. It would cover the process flows, happy path scenario, error and exceptional scenarios for the systems that are in scope.

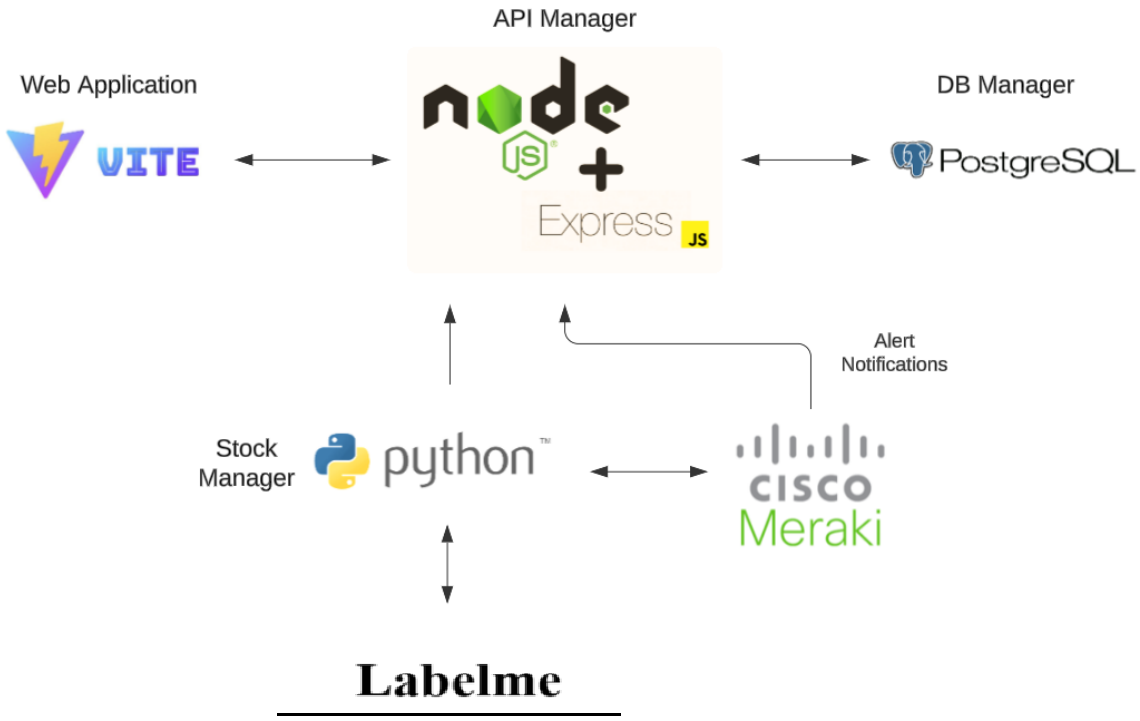
This document would highlight the dependency on external systems/third party systems like MERAKI etc. The solution or design for those systems would not be part of this document.

High level Interface information/dependency would be captured here.
This document may have reference to supplement documents for UI/UX, BRDs, MDDs, Product Taxonomy etc.

Functional Overview

Overview

Below is the high-level application flow with the details of systems/services involved along with the technical stack.



Below are the components and technical stack used for the development and integration,

- **Web Application (VITE Js):**
  Web application is developed using VITE Js (which is an enhanced and evolved version of React Js)

- **API Manager (Node and Express Js):**
  Backend service layer is built and developed by using Node and Express Js. This system acts as an integrator service for all internal and external integrations.
  It reads the alert notifications sent by Meraki and captures the same in the platform followed by populating them in the web application.

- **Stock Manager (Python):**
  Stock Manager is a sub system which is developed in Python. This system is integrated with Meraki to fetch, sync and update the stock back to the platform DB via API Manager.
    - Cisco Meraki detects the camera moments and sends the alerts to the API manager via integrated web-hook.
    - Labelmeis the tool used for annotating the images (by defining the part of the objects) and providing the statistics for the given image based on the annotations defined.

  Stock Manager (Python application) periodically checks the updated images in Meraki and send it to Labelme for scanning based on the defined annotations, followed by getting the stock status of each product from labelme.

- **DB Manager**
  PostgreSQL is used for capturing the users, analytics, products etc. and API Manager is integrated with the DB.

Users/Actors

There are two types of users who can access the application,

1. Admin
2. Imprest User

Admin

Admin is the user who can login and access the Imprests and products along with the stock status.

Admin user can create the application users and manage them as well.

Admin can create and manage the roles and role configurations for the users (User – Imprest configurations)

Admin can create the products and assign them to the Imprests (Imprest – Product configurations)

Admin can also manage the stock status and raise the POs accordingly.

Imprest User

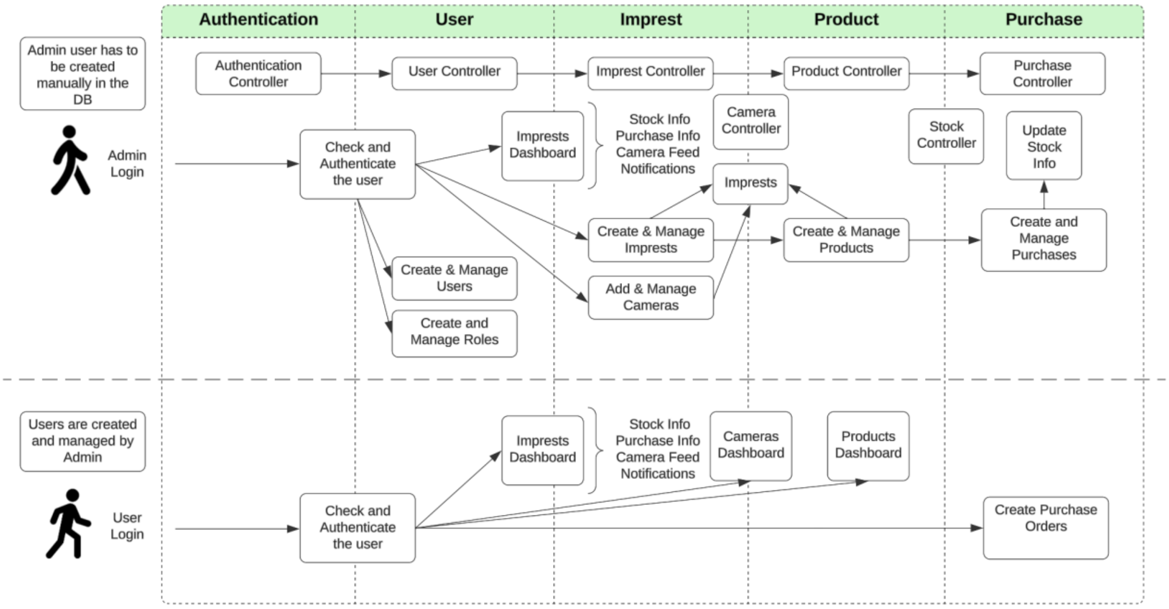
These are the application users who can login and access the assigned Imprests and monitor the products and stock info.

These users can also initiate the purchase orders in the application.

Technical Overview

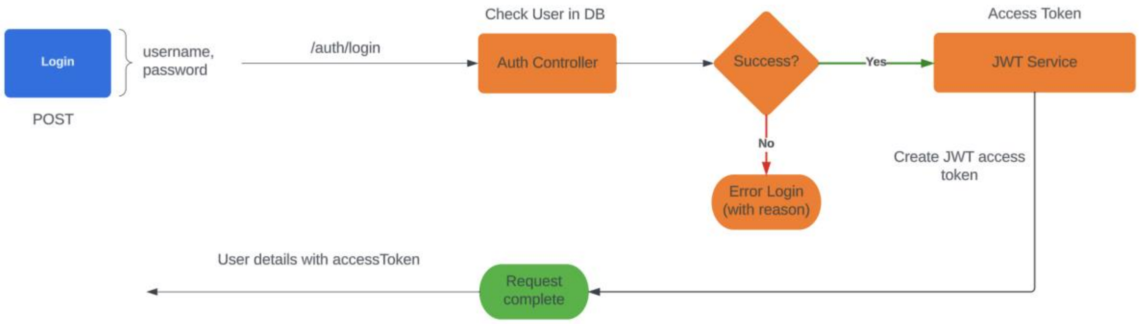
Technical Design

Below is the overview of controllers and backend implementation,



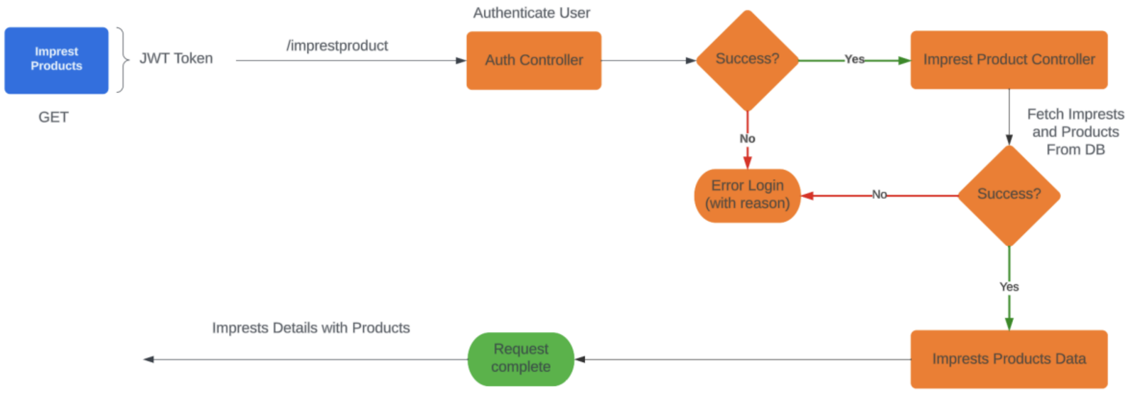
Login

This API is used for authenticating the users with username and password. Same API is used for admin and normal user.



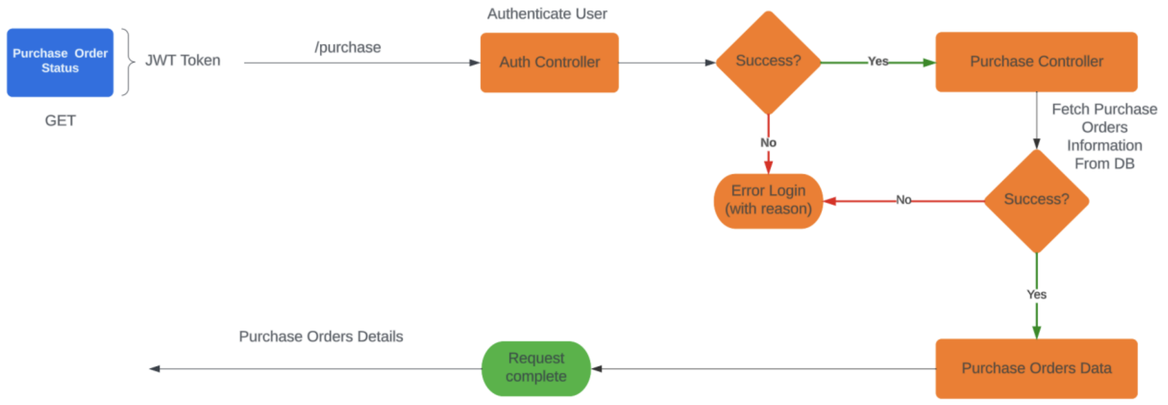
Imprests Products

This API is used to fetch the list of Imprests along with products from the database.



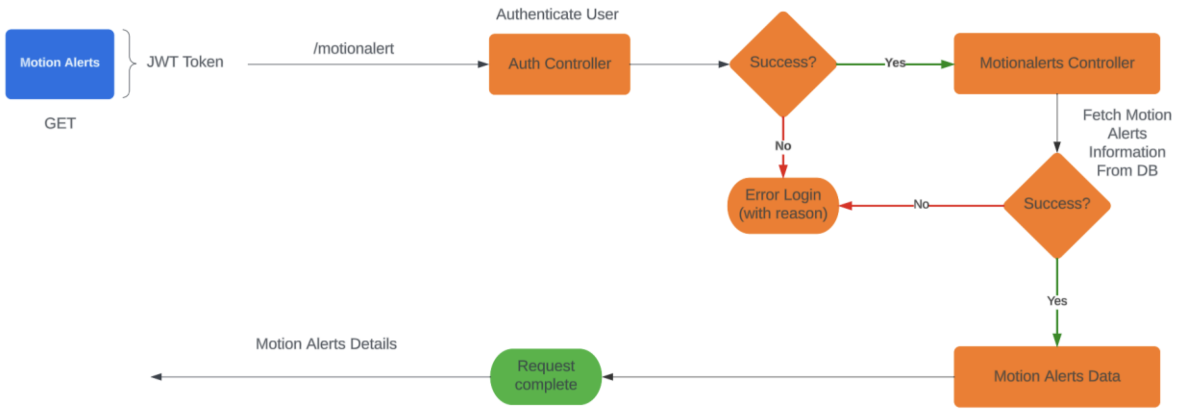
Purchase Order Status

This API is used to fetch the list of purchase orders from the database.



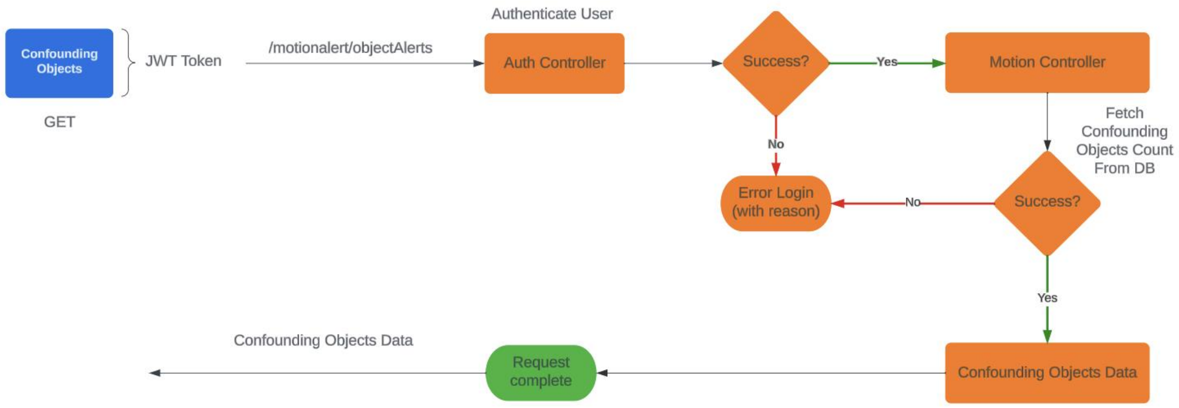
Motion Alerts

This API is used to fetch the motion alerts from the database.



Confounding Objects

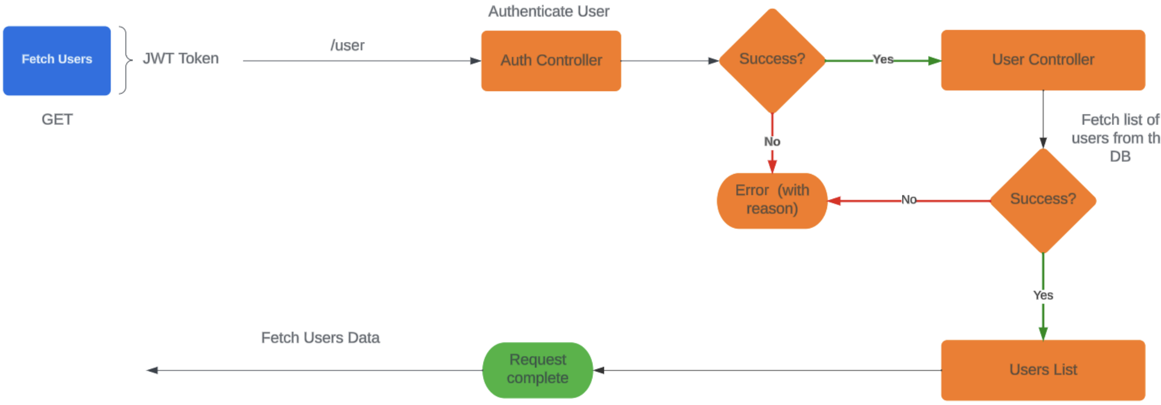
This API is used to fetch the confounding objects from the database.



User Module

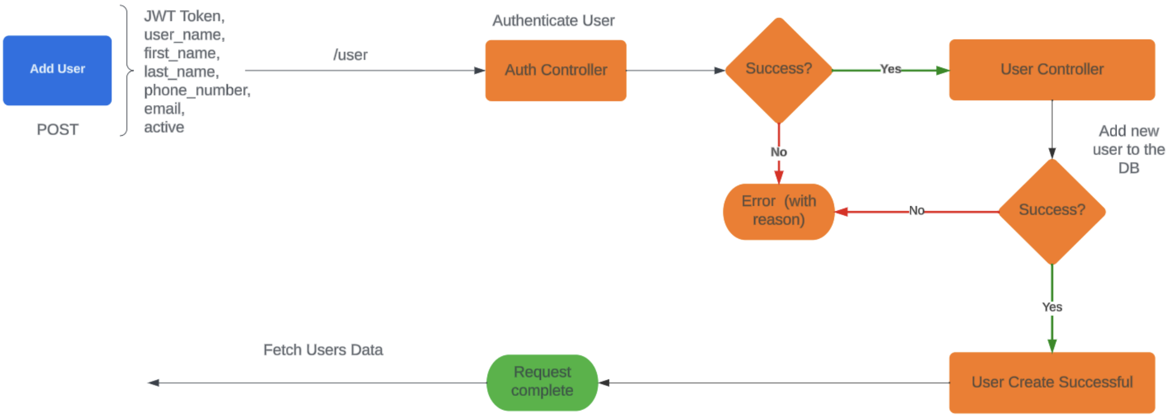
Fetch Users

This API is used to fetch list of users from the database.



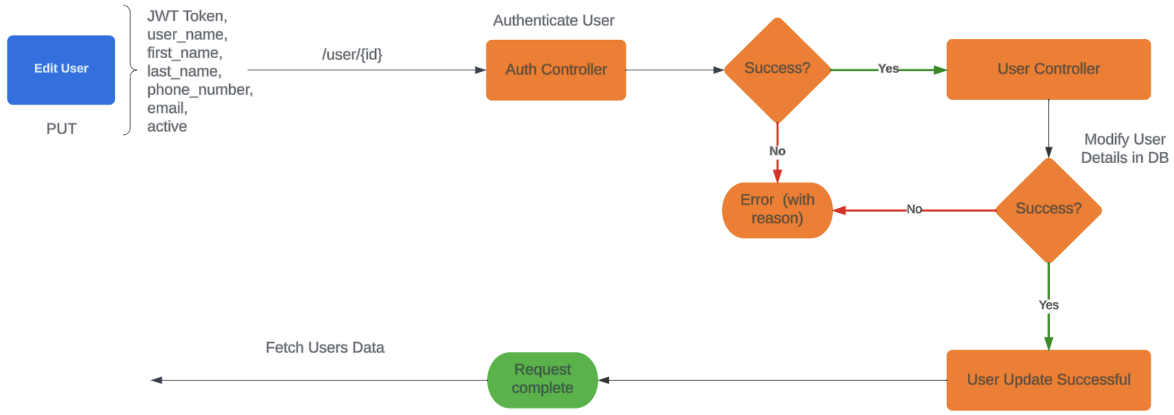
Add User

This API is used to create/add new user to the database.



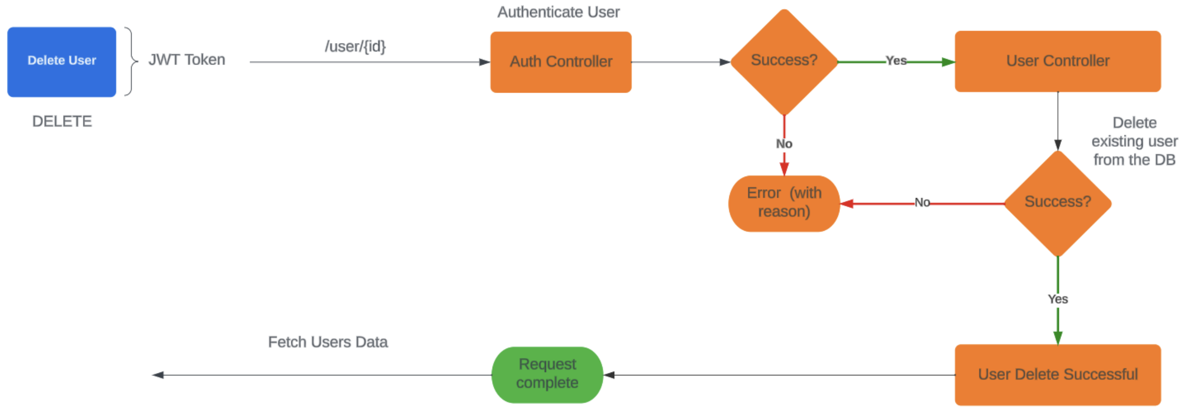
Edit User

This API is used to modify the existing user in the database.



Delete User

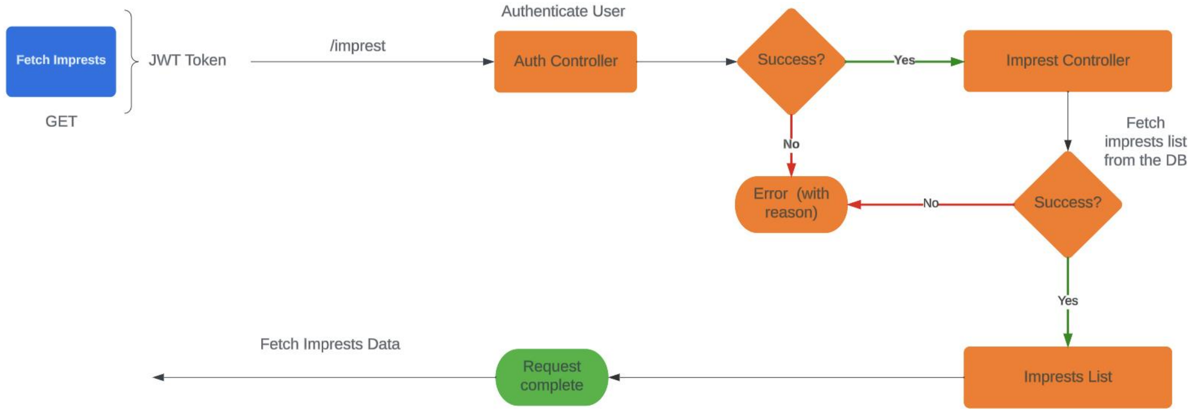
This API is used to delete the user from the database.



Imprest Module

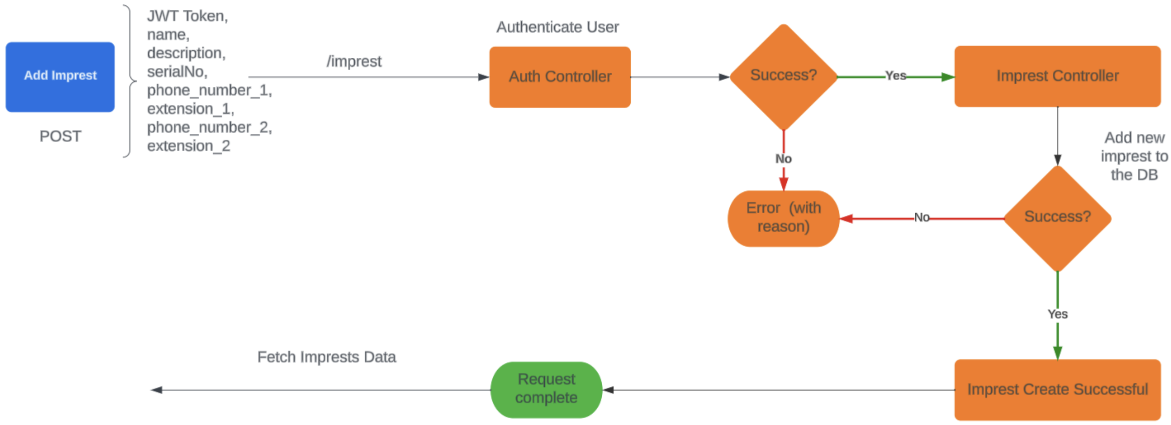
Fetch Imprests

This API is used to fetch list of Imprests from the database.



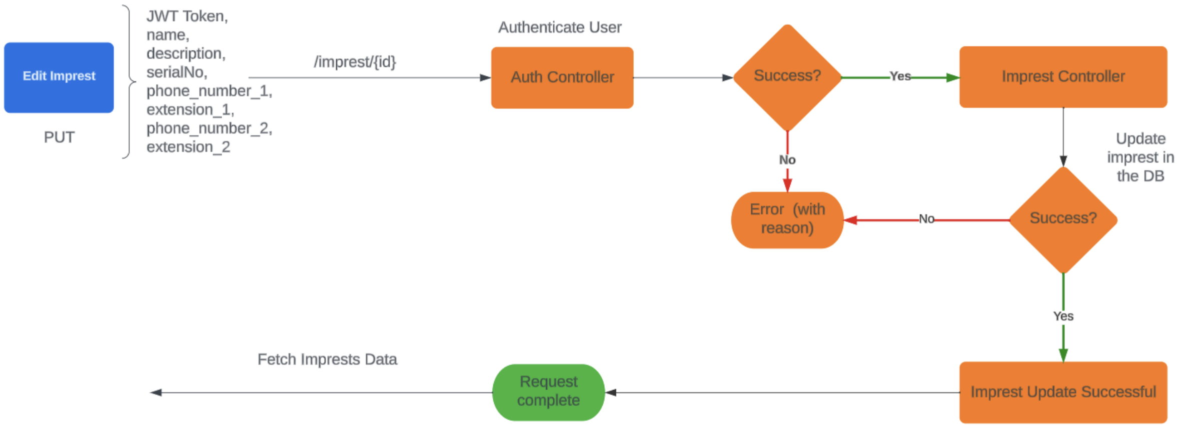
Add Imprest

This API is used to create/add new Imprest to the database.



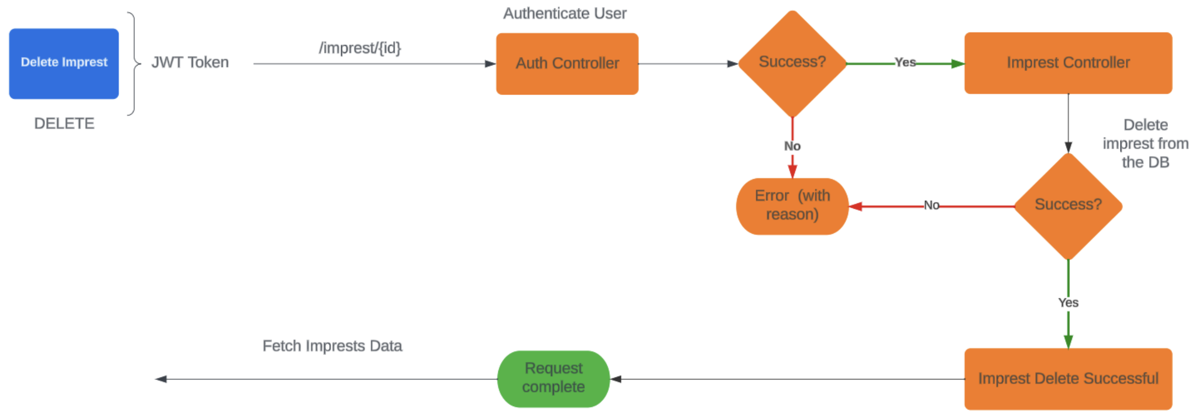
Edit Imprest

This API is used to modify and update the Imprest information in the database.



Delete Imprest

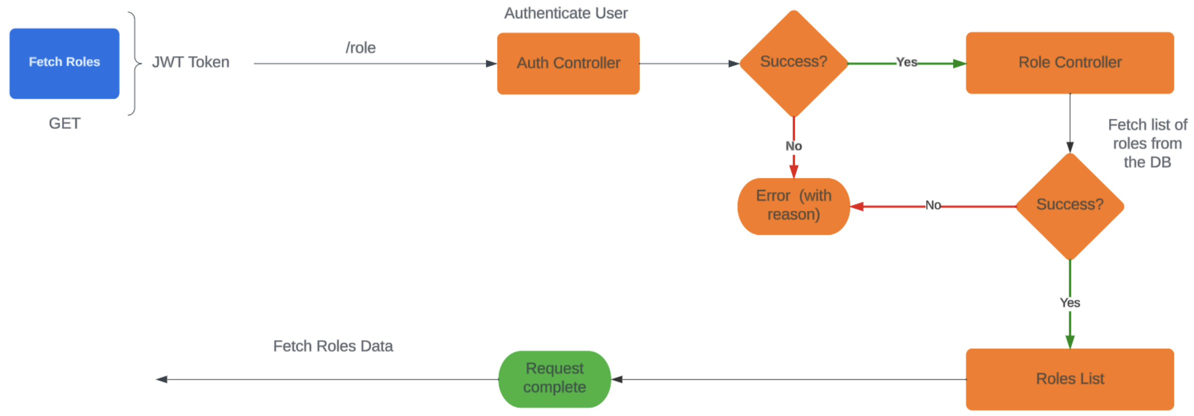
This API is used to delete an Imprest from the database.



Roles Module

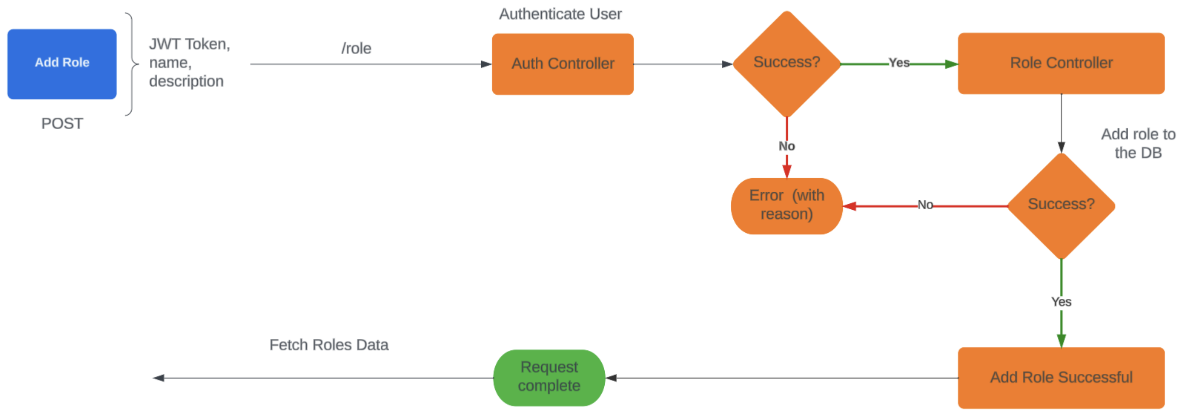
Fetch Roles

This API is used to fetch the list of roles from the database.



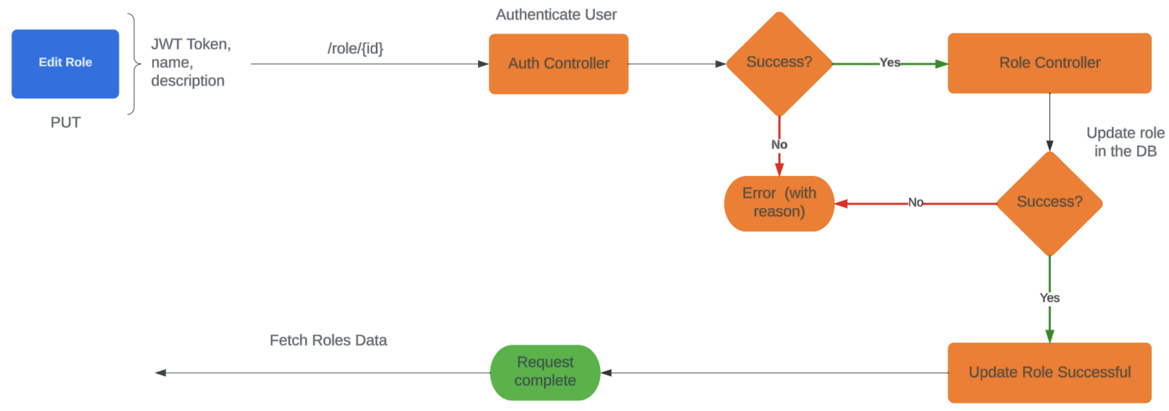
Add Role

This API is used to create/add a role to the database.



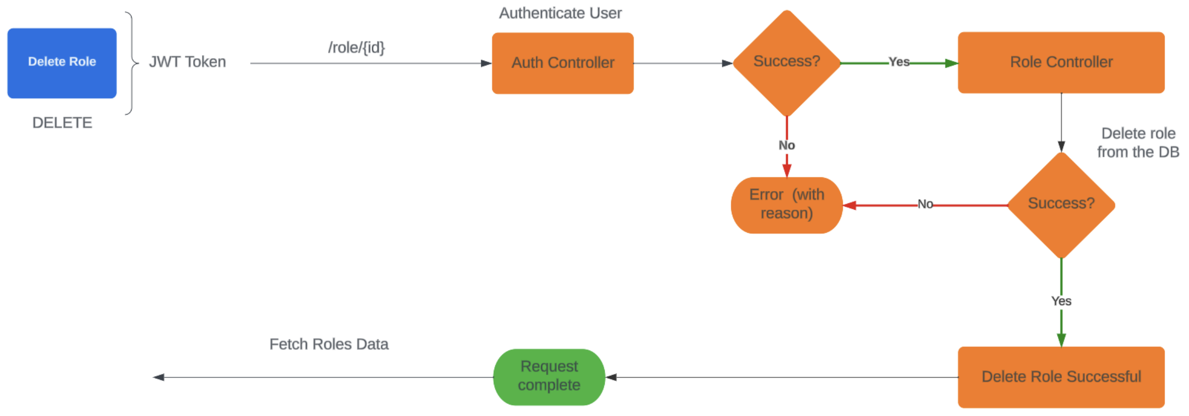
Edit Role

This API is used to modify the existing role in the database.



Delete Role

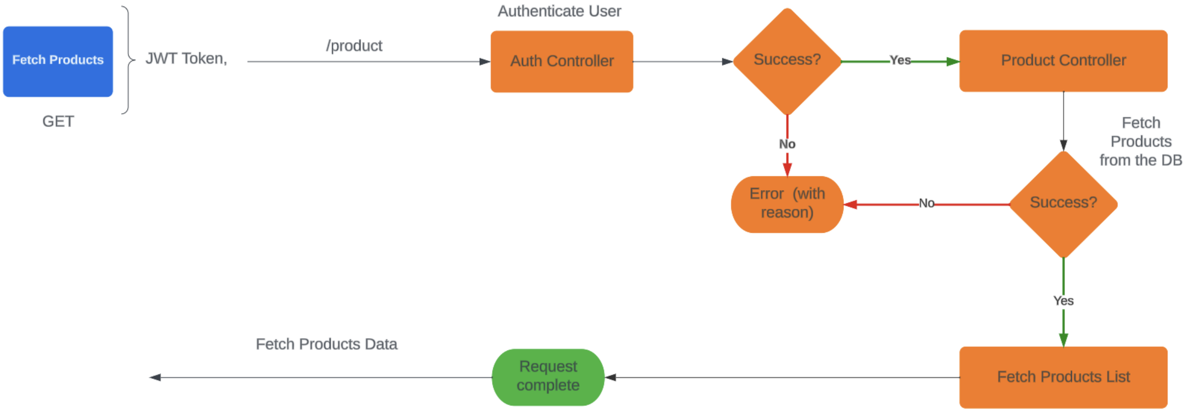
This API is used to delete the role from the database.



Products Module

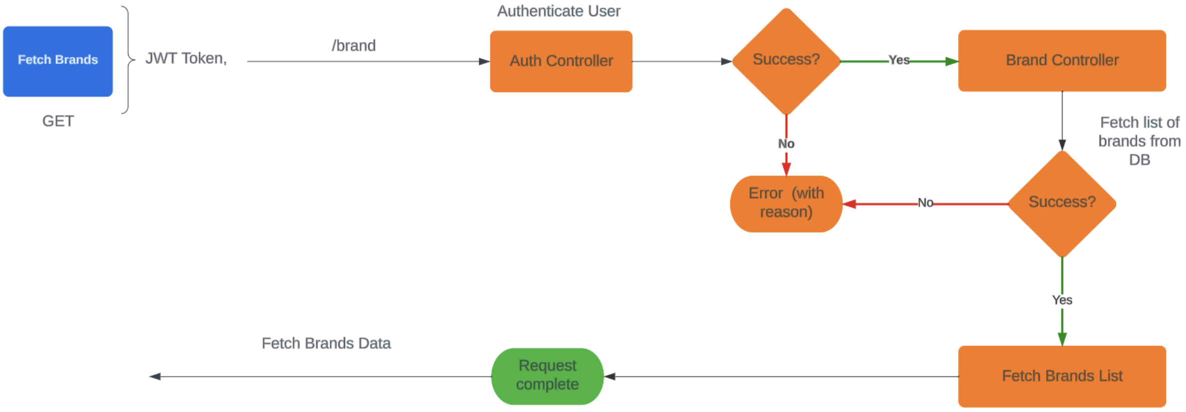
Fetch Products

This API is used to fetch the list of products from the database.



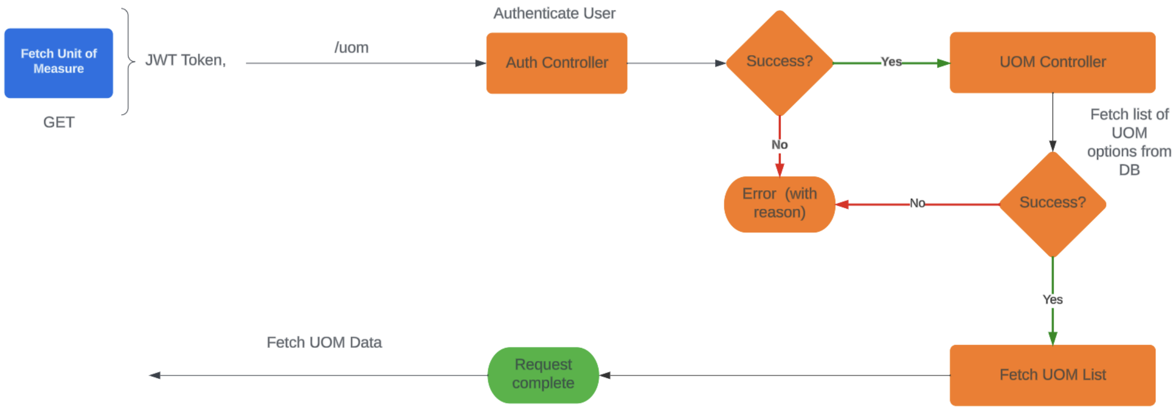
Fetch Brands

This API is used to fetch the list of brands from the database.



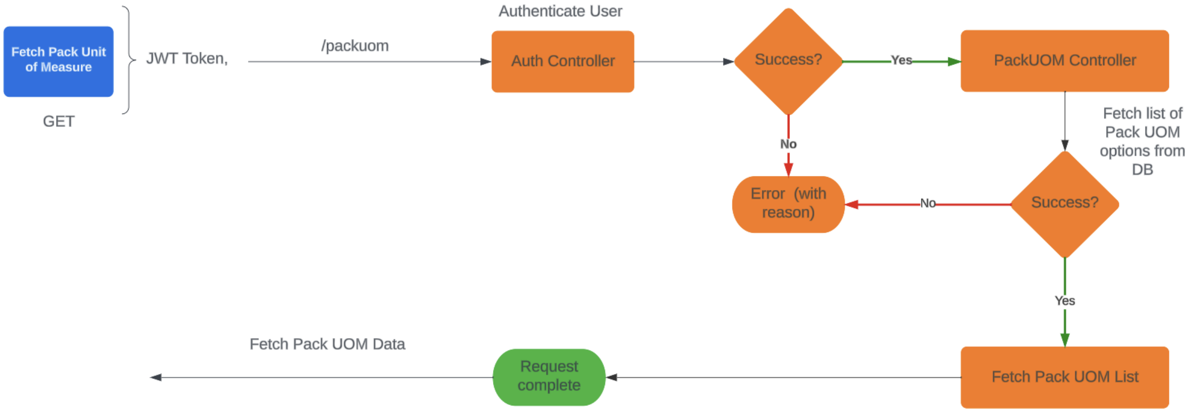
Fetch Unit of Measure

This API is used to fetch unit of measure options from the database.



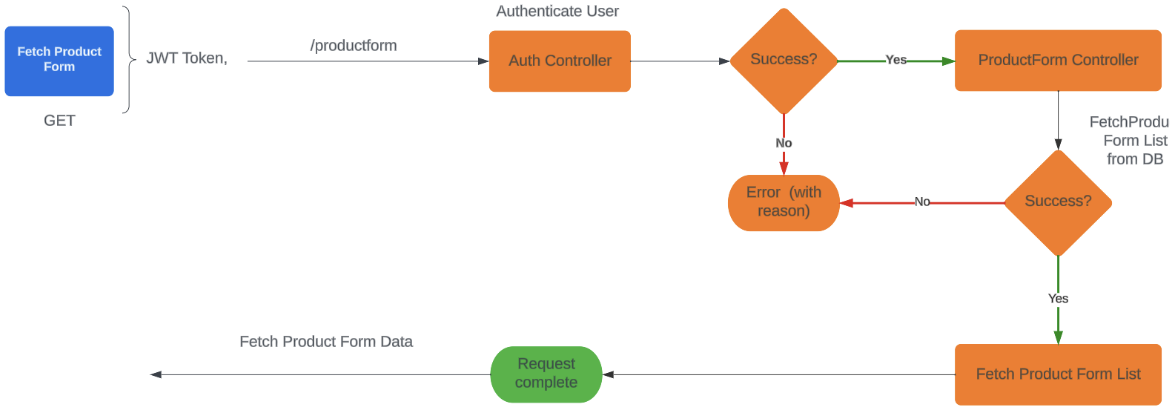
Fetch Pack Unit of Measure

This API is used to fetch all the pack unit of measure options from the database.



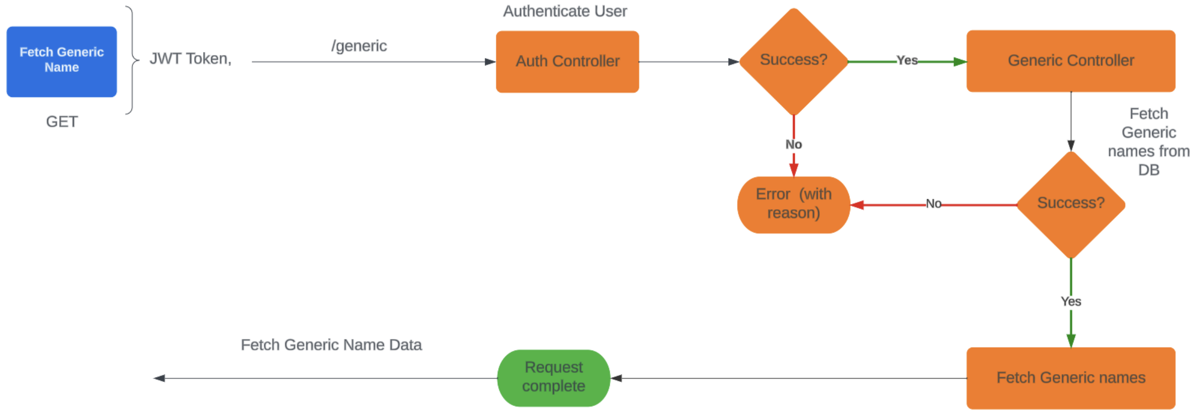
Fetch Product Form

This API is used to fetch all the product forms from the database.



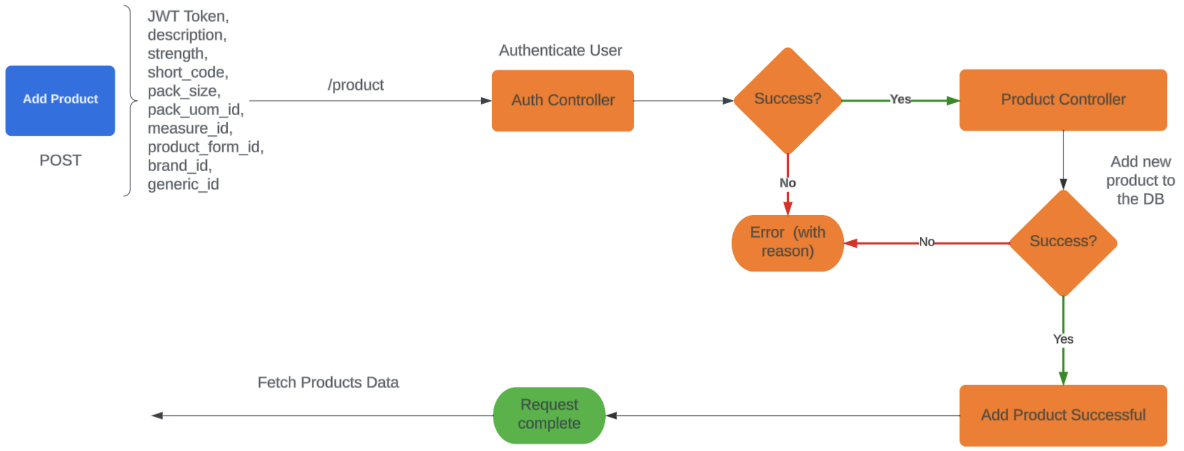
Fetch Generic Name

This API is used to fetch all the generic names from the database.



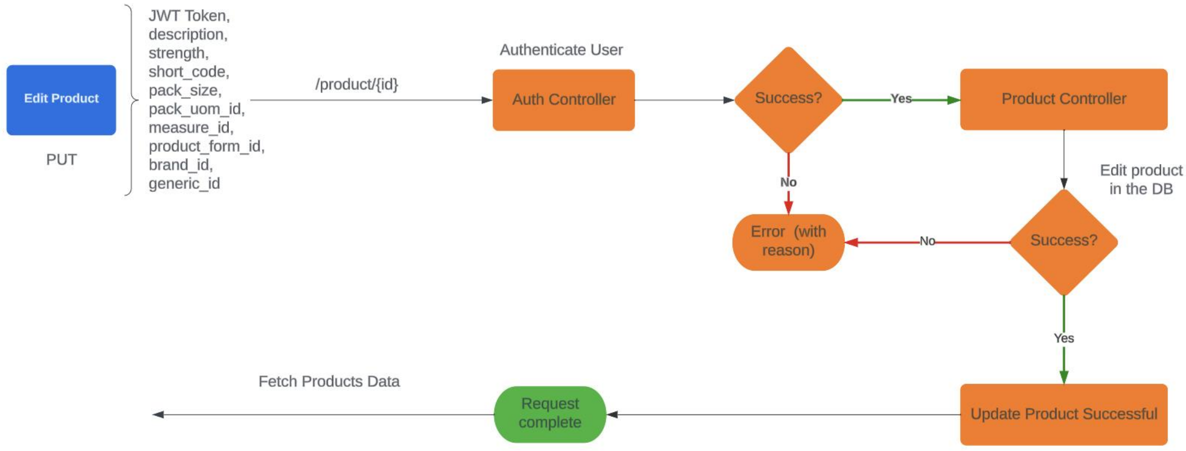
Add Product

This API is used to create/add a new product to the database.



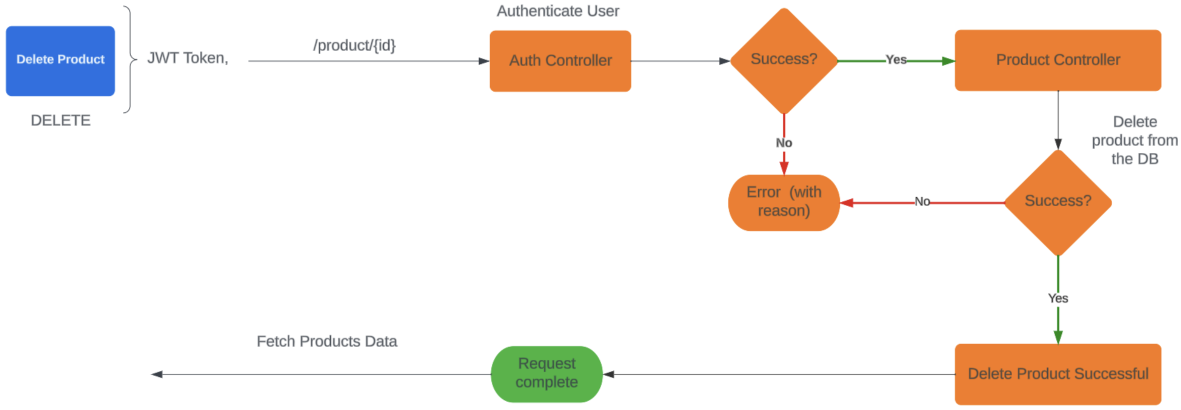
Edit Product

This API is used to modify and update any product information in the database.



Delete Product

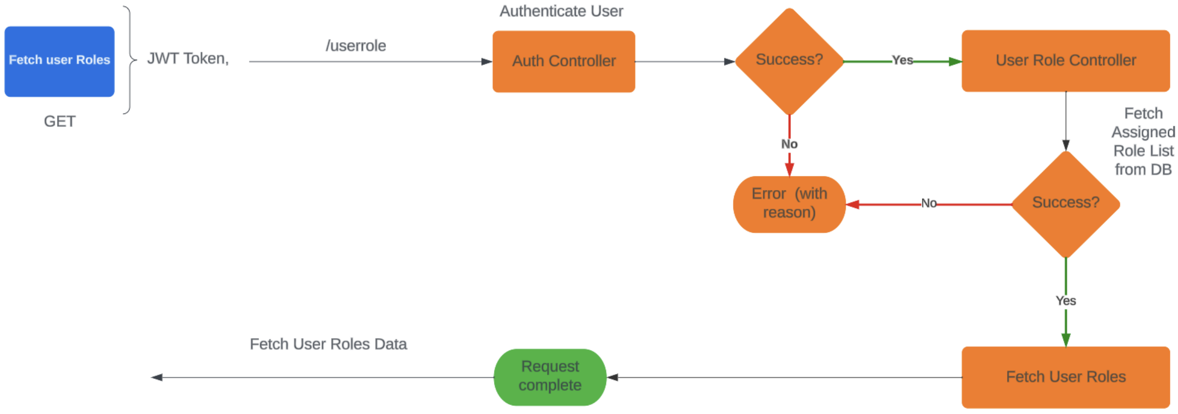
This API is used for removing any product from the Database.



User Roles Module

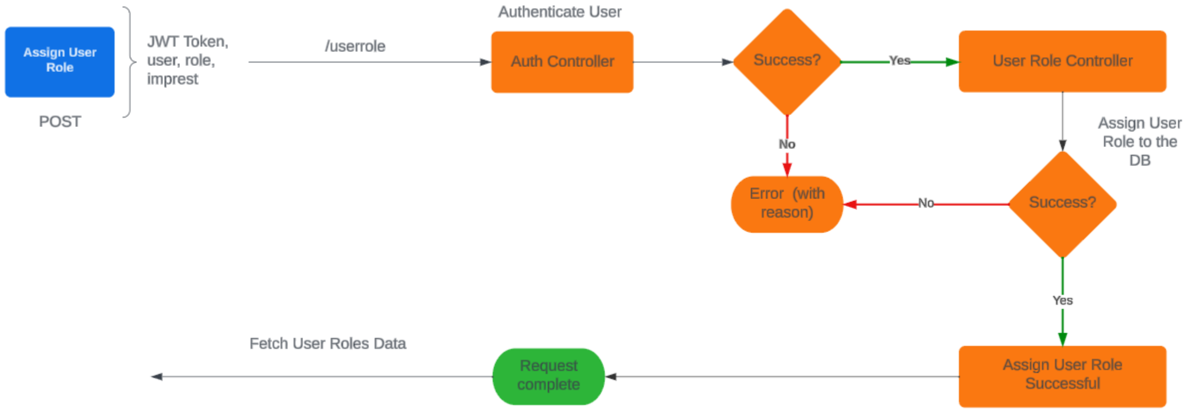
Fetch User Roles

This API is used for fetching all the user role configurations present in the application.



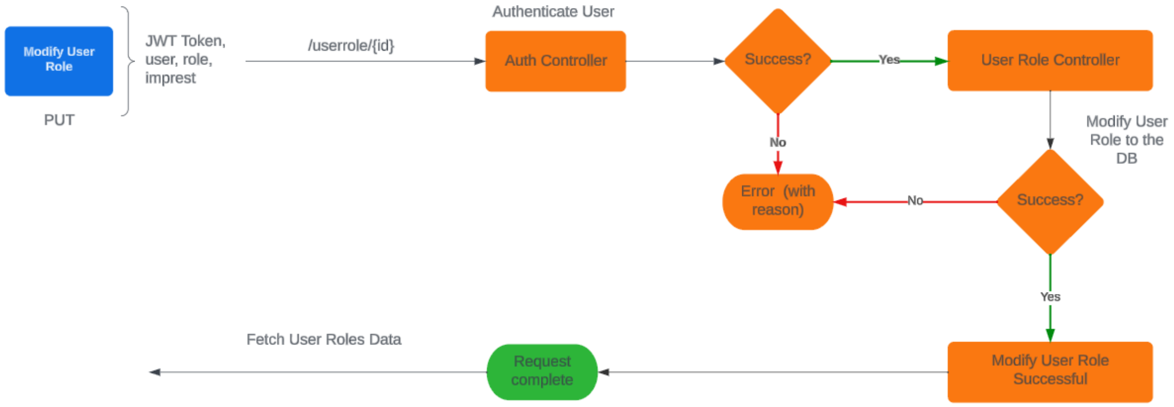
Assign User Role

This API is used to create and assign the User Role Configuration in the database.



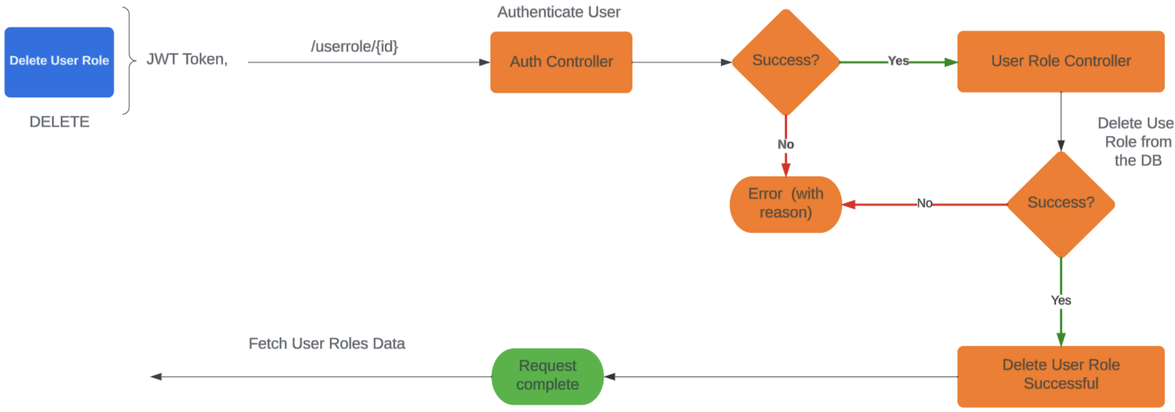
Modify User Role

This API is used to update the User Role Configuration in the database.



Delete User Role

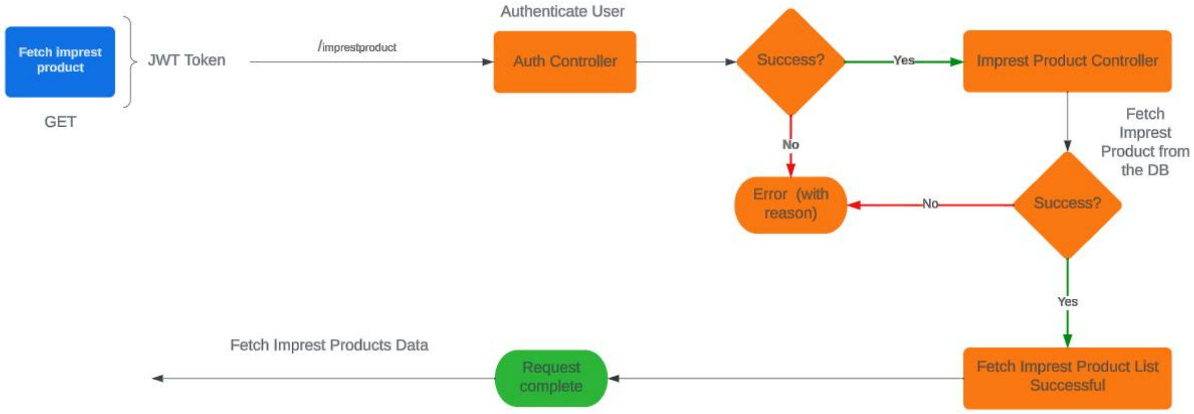
This API is used to remove any User Role Configuration from the application/database.



Imprest Products Module

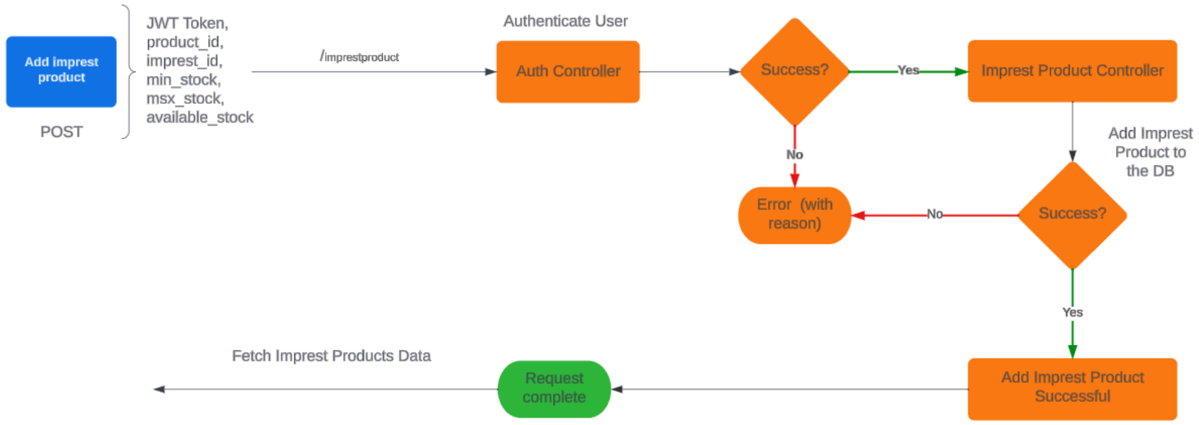
Fetch Imprest Products

This API is used to fetch all the Imprest product configurations created in the application.



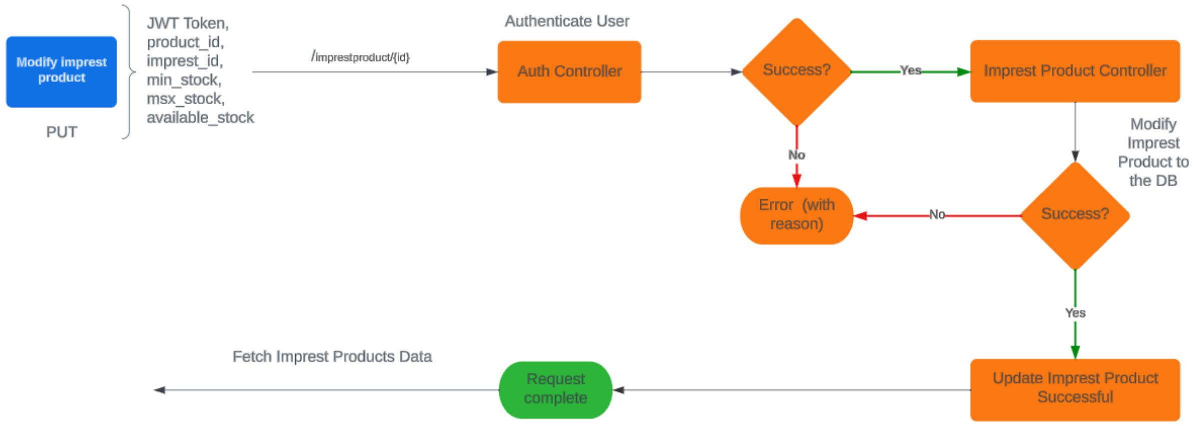
Add Imprest Product

This API is used for configuring the Imprest to a product with stock configuration.



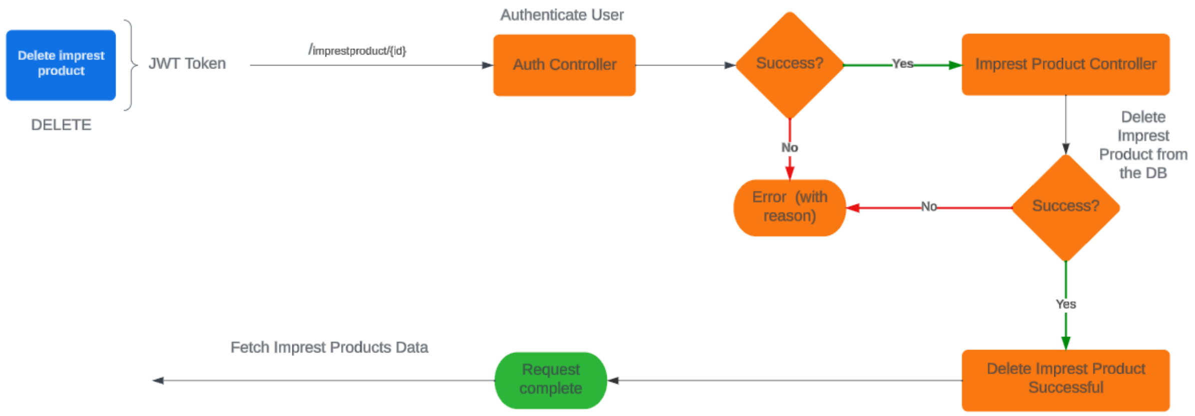
Edit Imprest Product

This API is used to modify the existing Imprest product configuration and update the data back to the Database.



Delete Imprest Product

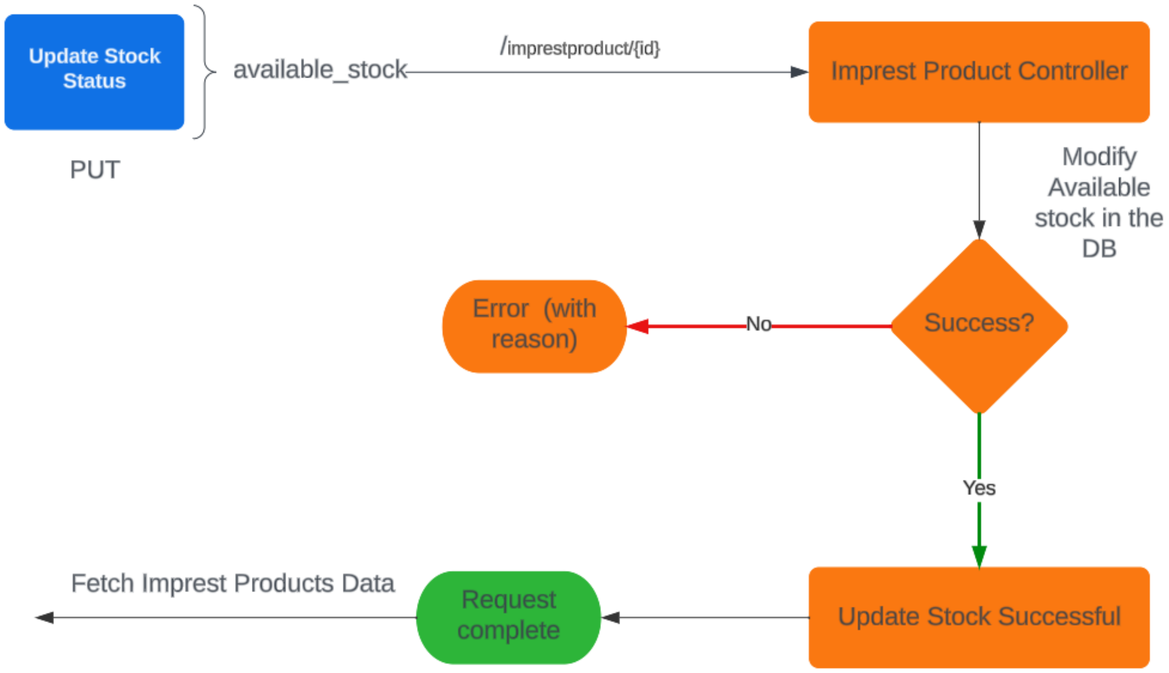
This API is used to delete any Imprest product configuration.



Stock Update Module

Update Stock Count

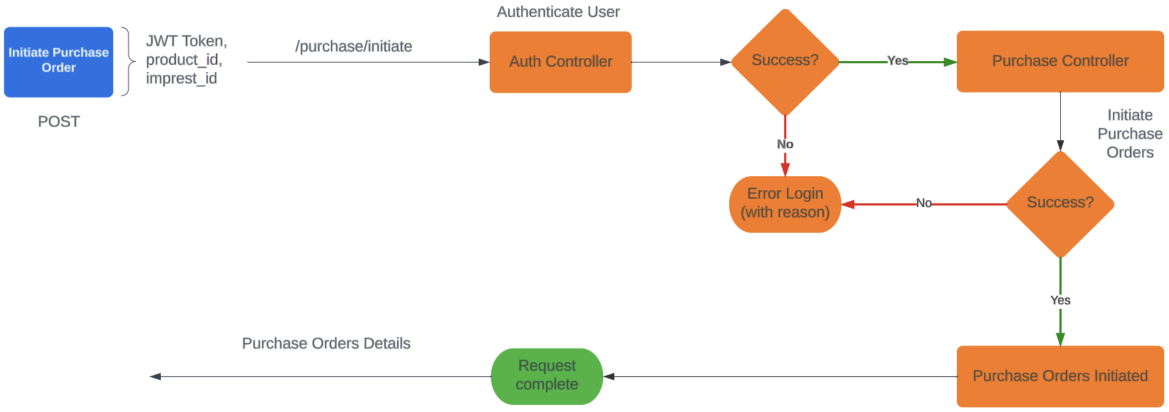
This API is used in the Python Scripts for updating the stock status to the Database.



Purchase Module

Initiate Purchase Order

This is the API used for initiating the purchase order for the products of an imprest.

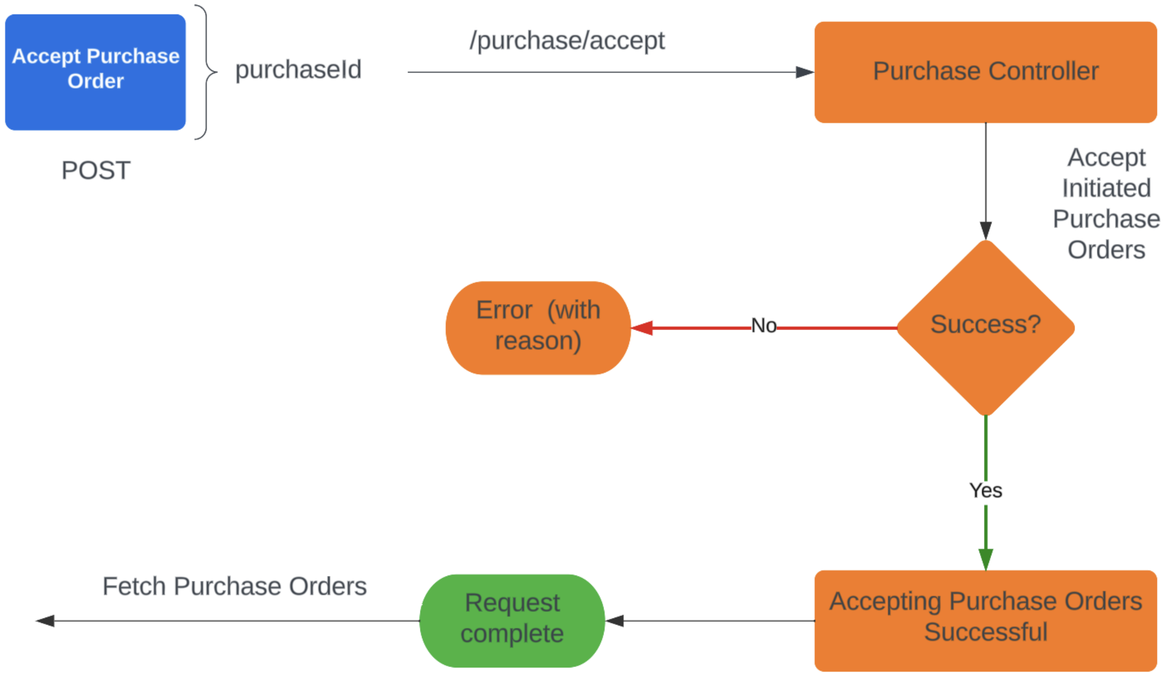


Request Payload as follows,

```
{
  "data": [
    {
      "imprest_id": 1,
      "product_id": 10
    },
    {
      "imprest_id": 1,
      "product_id": 12
    }
  ]
}
```

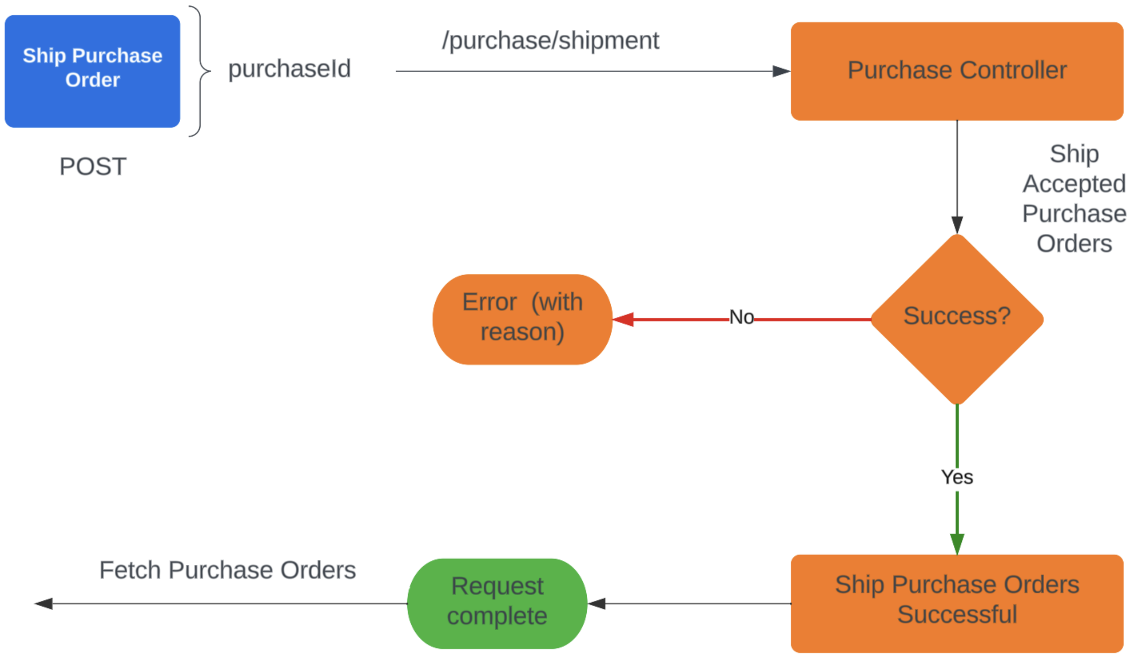
Accept Purchase Order

This API is used for accepting the purchase orders which are initiated from the application



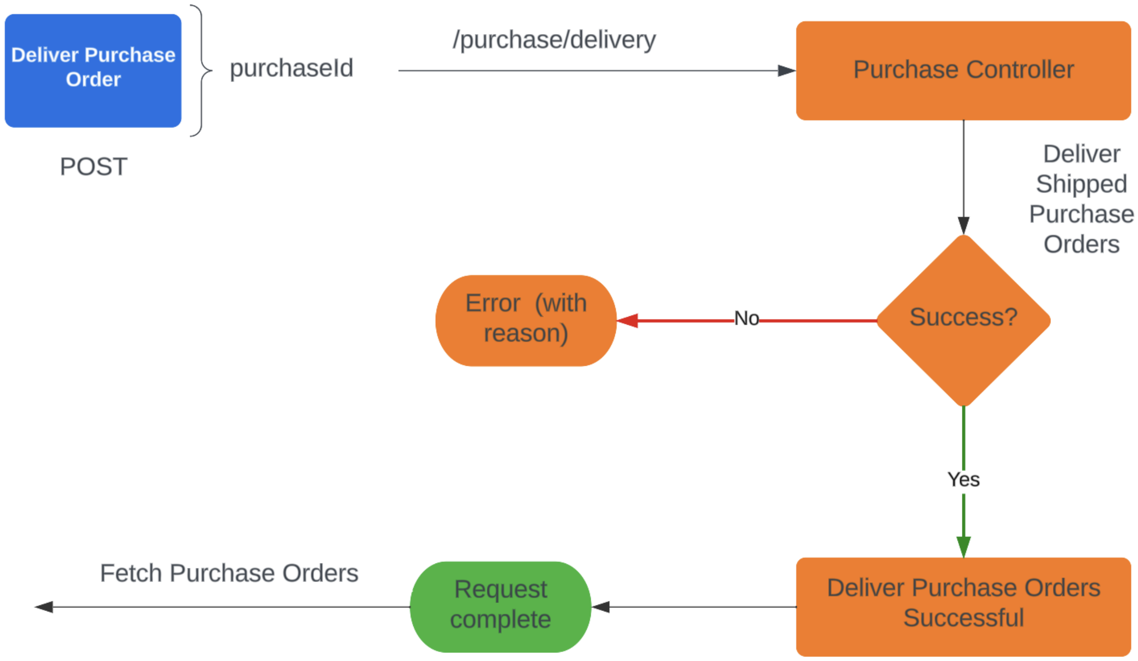
Ship Purchase Order

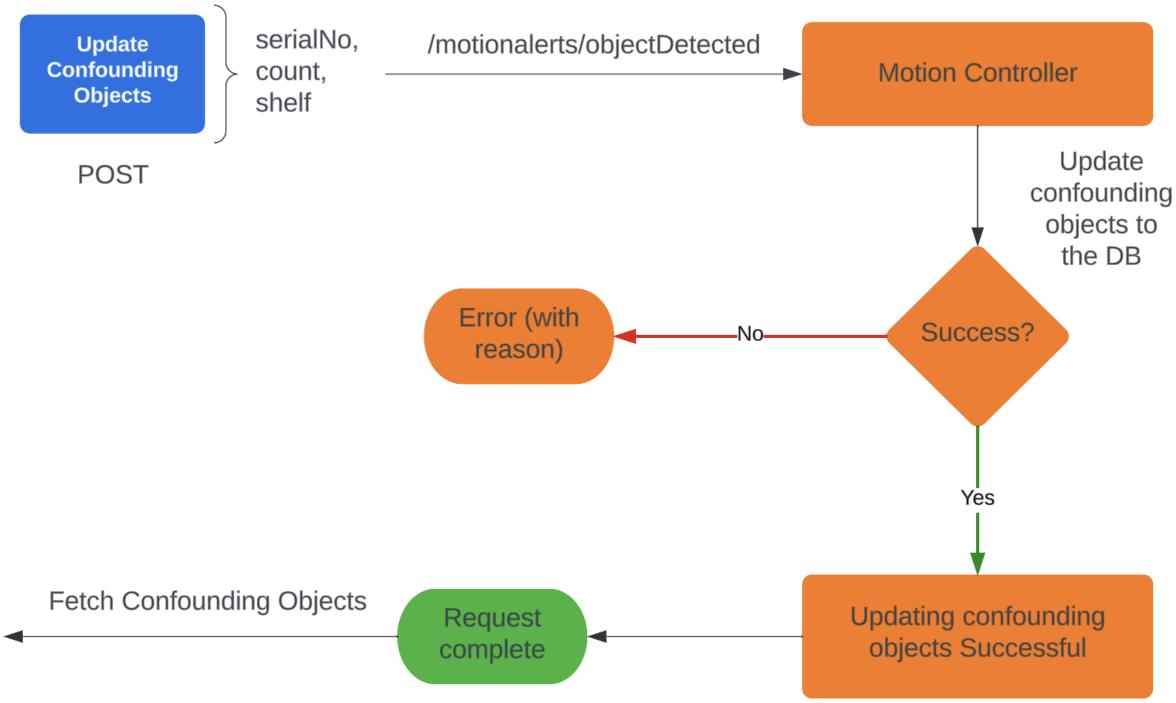
This API is used to initiate the shipment for the purchase orders accepted.



Deliver Purchase Order

This API is used to initiate the delivery for the purchase orders shipped.




Update Confounding Objects
POST
serialNo, count, shelf
/motionalerts/objectDetected
Motion Controller
Update confounding objects to the DB
Success?
No
Error (with reason)
Yes
Updating confounding objects Successful
Request complete
Fetch Confounding Objects