**Products API**Geoff Campbell-Brady

<https://www.linkedin.com/in/geoff-campbell-brady-594b9b18/>

**Table of Contents**

[1. Architecture diagram including Products and other components](#_Toc197287102)

[1.1 Explanation](#_Toc197287103)

[2. Unit and Integration Tests](#_Toc197287104)

[3. Docker Desktop](#_Toc197287105)

[4. Swagger UI](#_Toc197287106)

[5. Heath check](#_Toc197287107)

[6. Login](#_Toc197287108)

[7. Authorization in Swagger](#_Toc197287109)

[8. Adding a product](#_Toc197287110)

[9. Returning products based on colour](#_Toc197287111)

[10. Connecting via Web API via Postman](#_Toc197287112)

[11. Getting Bearer token via Postman](#_Toc197287113)

[12. Postman demonstrating validation](#_Toc197287114)

[13. Postman returning products](#_Toc197287115)

# Architecture diagram including Products and other components

**Shop Web Page**View products, place orders, pay, shipping

**Gateway API**

**Shipping API and DB**

**Payments API and DB**

**Orders API and DB**

**Products API and DB**

Publish ShippingProcessed Event

Subscribe OrdersCreated Event

Publish OrdersCreated Event

Publish PaidProcessed Event

Subscribe PaidProcessed Event

Subscribe OrdersCreated Event

Subscribe PaidProcessed Event

Subscribe ShippingProcessed Event

**Event broker**Topics

## Explanation

* Clients use the shopping web page to view products, place orders, pay and to deal with shipping.
* The page communicates with the Gateway API which routes requests to the relevant API
* Each API has its own database. The Products API database has a listing of inventory which must be updated when the users order (product items become reserved), pay (product items marked as sold) or shipped (product items removed from the inventory). It does this by subscribing to events published by the APIs via the Event Broker.
* The Order API publishes an OrdersCreated event when a user orders items.
* The Payments API subscribes to the OrdersCreated event, redirects the user to a payment section and when the user has paid raises a PaidProcessed event.
* The Shipping API subscribes to the PaidProcessed event so it arranges shipping and publishes a ShippingProcessed event.
* In Azure, the Gateway API would be Azure API Management and the message broker would be the most reliable one which is Azure Service Bus (though this is not the fastest one that Azure offers).

# Unit and Integration Tests

A screenshot of a computer

AI-generated content may be incorrect.

PC tested on has Windows Home which is not compatible with Hyper-V for Virtualisation so WSL2 and Docker Desktop was used.

# Docker Desktop

A screenshot of a computer

AI-generated content may be incorrect.

# Swagger UI

A screenshot of a computer

AI-generated content may be incorrect.

# Heath check

A screenshot of a computer

AI-generated content may be incorrect.

# Login

A screenshot of a computer

AI-generated content may be incorrect.

# Authorization in Swagger

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Adding a product

A screenshot of a computer

AI-generated content may be incorrect.

# Returning products based on colour

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Connecting via Web API via Postman

Getting bearer token biaA screenshot of a computer

AI-generated content may be incorrect.

# Getting Bearer token via Postman

A screenshot of a computer

AI-generated content may be incorrect.

# Postman demonstrating validation

A screenshot of a computer

AI-generated content may be incorrect.

# Postman returning products

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.