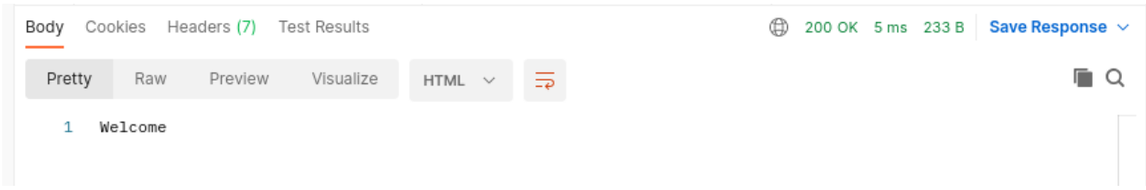
Documentation: Consists of Introduction, Routes, and Errors.
Introduction:
This API facilitates the storage of product info for a shop.
An item object is created and stored in the database. It contains the following structure:
ITEM

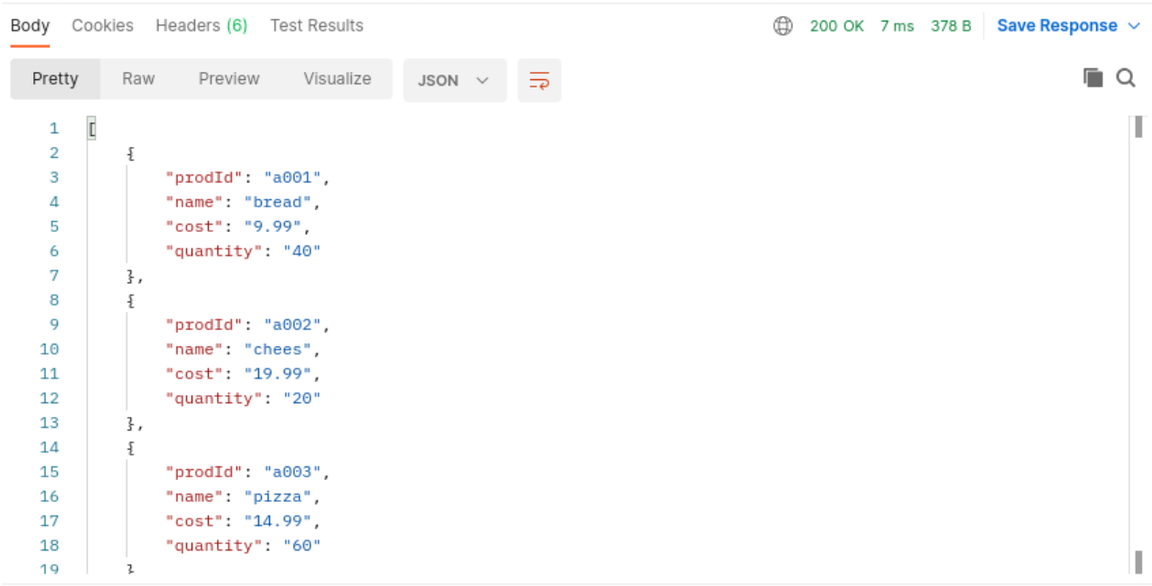
Property Name	Datatype	Comments
prodId	string	PRIMARY KEY, UNIQUE
name	string	
cost	float	
quantity	positive integer	

Routes:

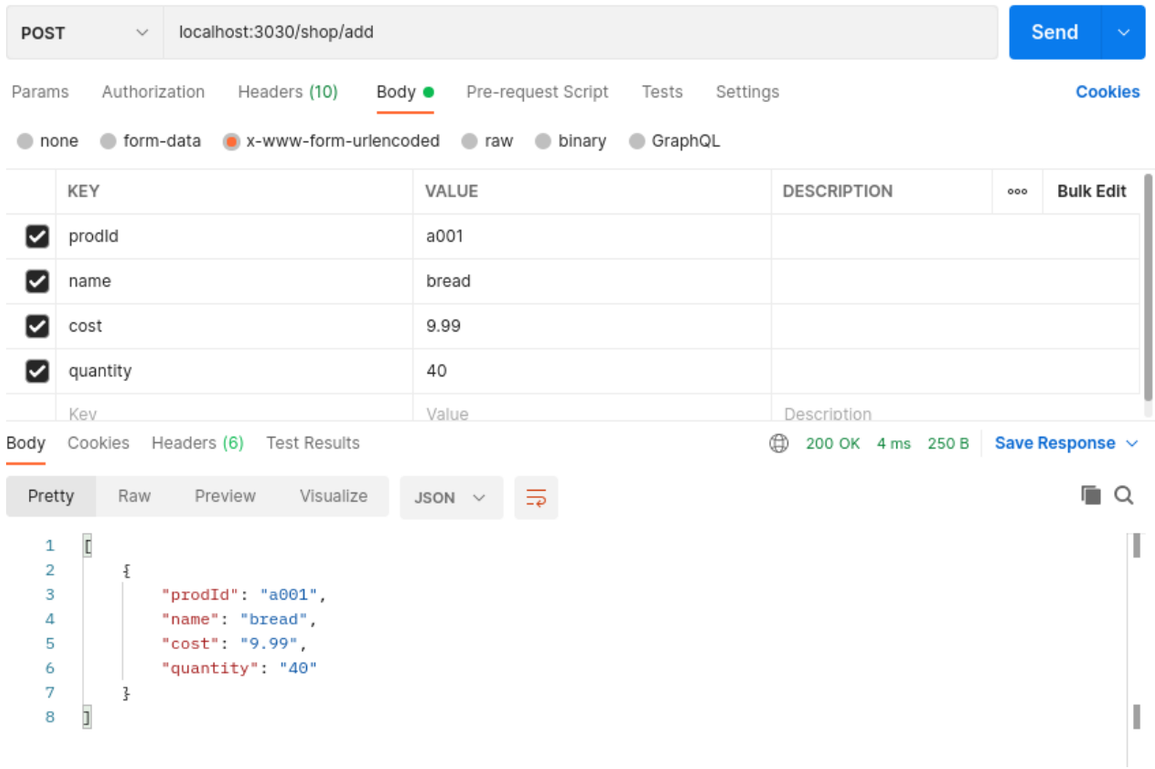
1) '/', Method: GET
   This route responds with "Welcome!". It is a test route to check if the server is online.



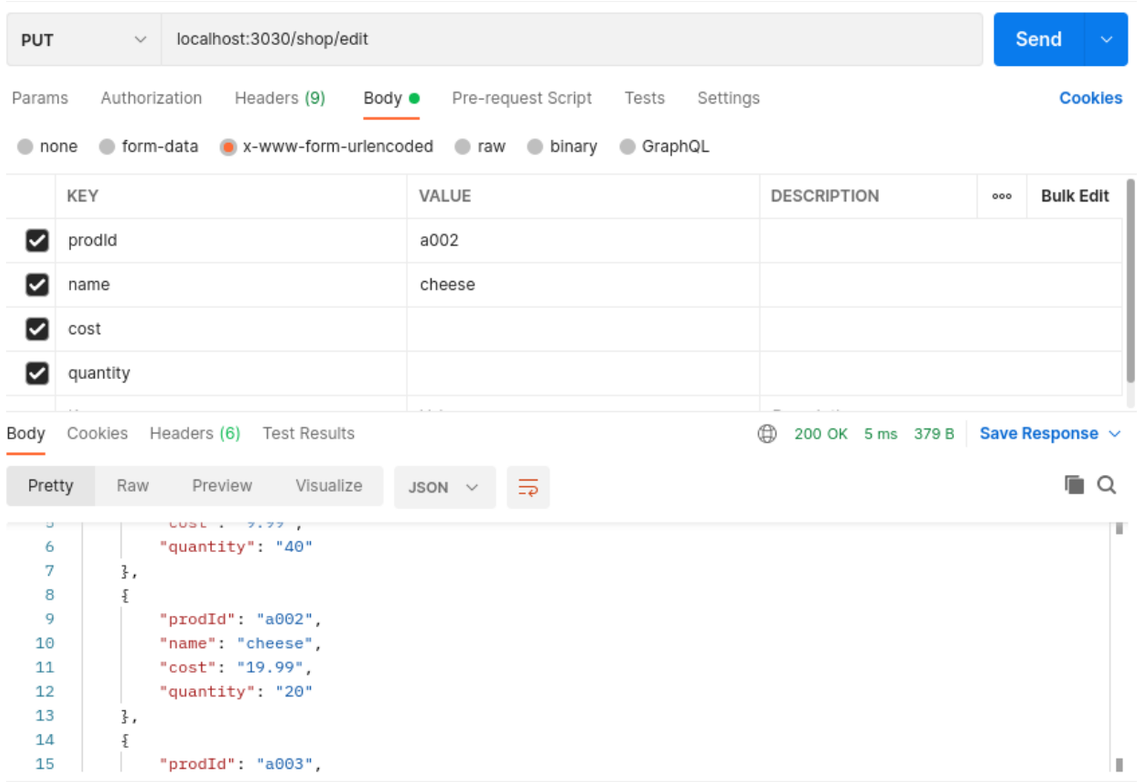
2) '/shop' Method: GET
   This route responds with an array containing all of the items in the stock database.



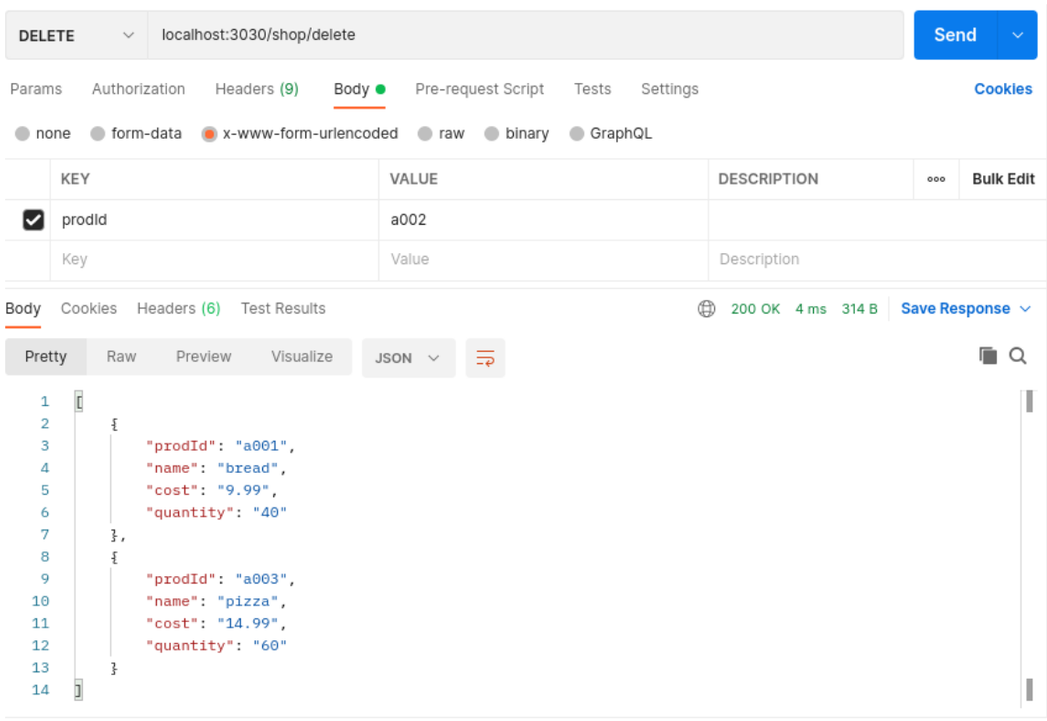
3) ‘/shop/add’ Method: POST
This route receives item data in the request body, checks for duplication, adds the item to the stock database, and responds with an array containing all of the items in stock.



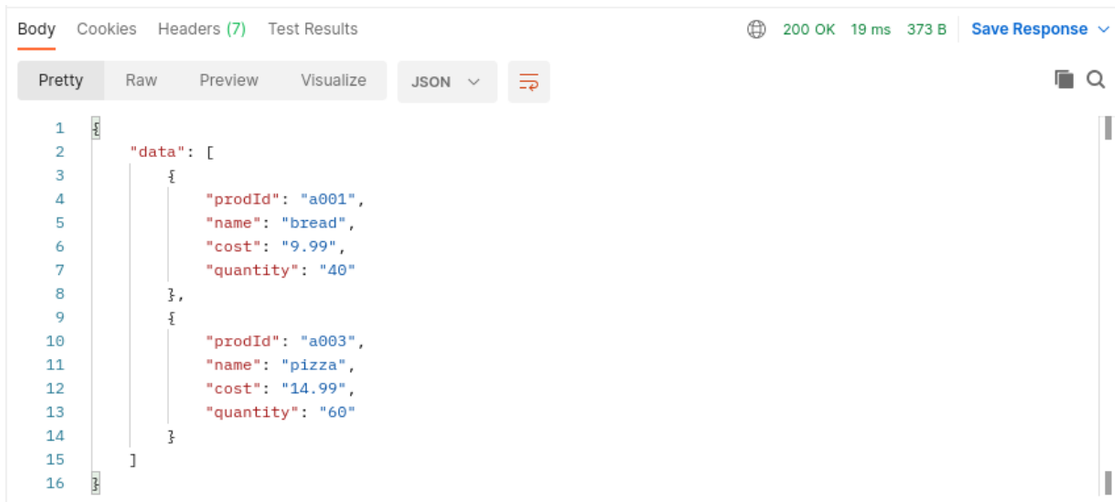
4) ‘/shop/edit’ Method: PUT
This route receives item data in the request body, checks if the item is found, edits the item info in the database, and responds with an array containing all of the items in stock.



5) ‘/shop/delete’ Method: DELETE
This route receives item prodId in the request body, checks if the item is found, deletes the item, and responds with an array containing all of the items in stock.

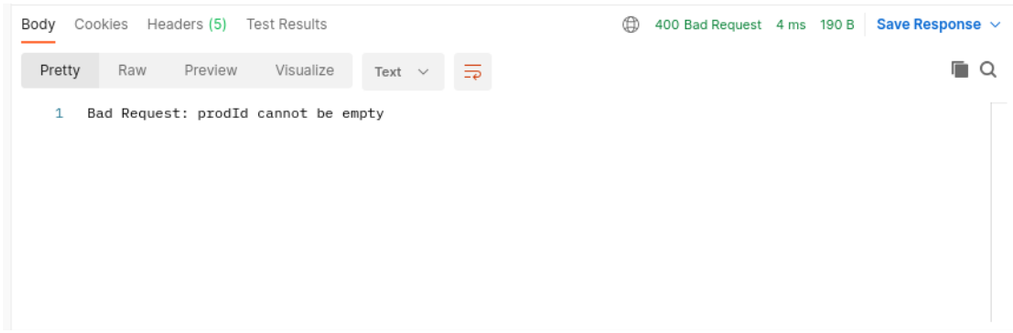


6) ‘/shop/csv’ Method: GET
This route generates a csv file, initiates a download request, and also responds with the generated csv.

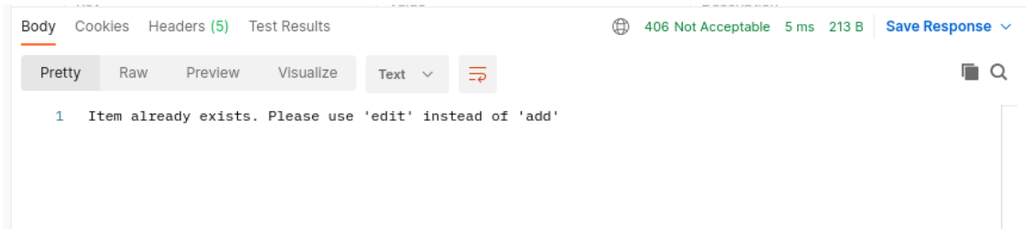


Errors:

1) If prodId is null: (/shop/add, /shop/edit, /shop/delete)



2) If item already exists (/shop/add)



3) If item doesn't exist (/shop/edit, /shop/delete)