## Steps to run

#### Start the application

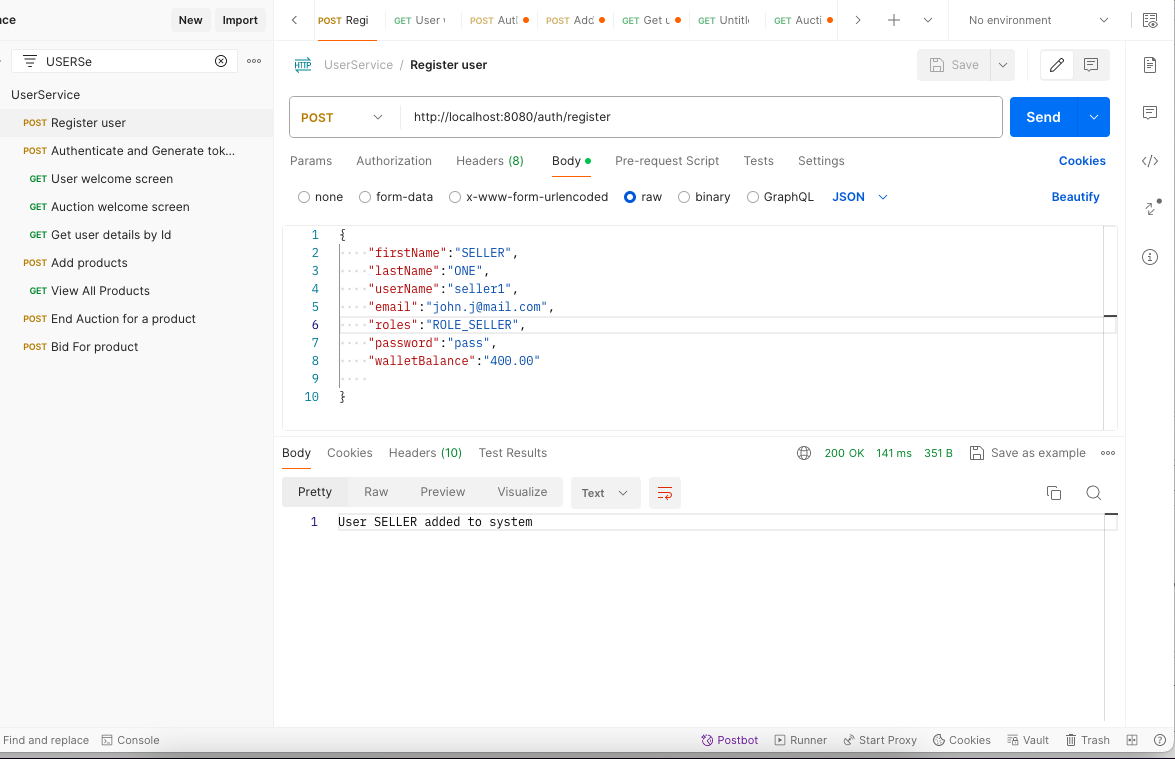
1. Run User service as a spring boot app
2. Run Auction Service as a spring boot app
3. check welcome API’ for both services should return welcome

#### Steps

1. Create users with

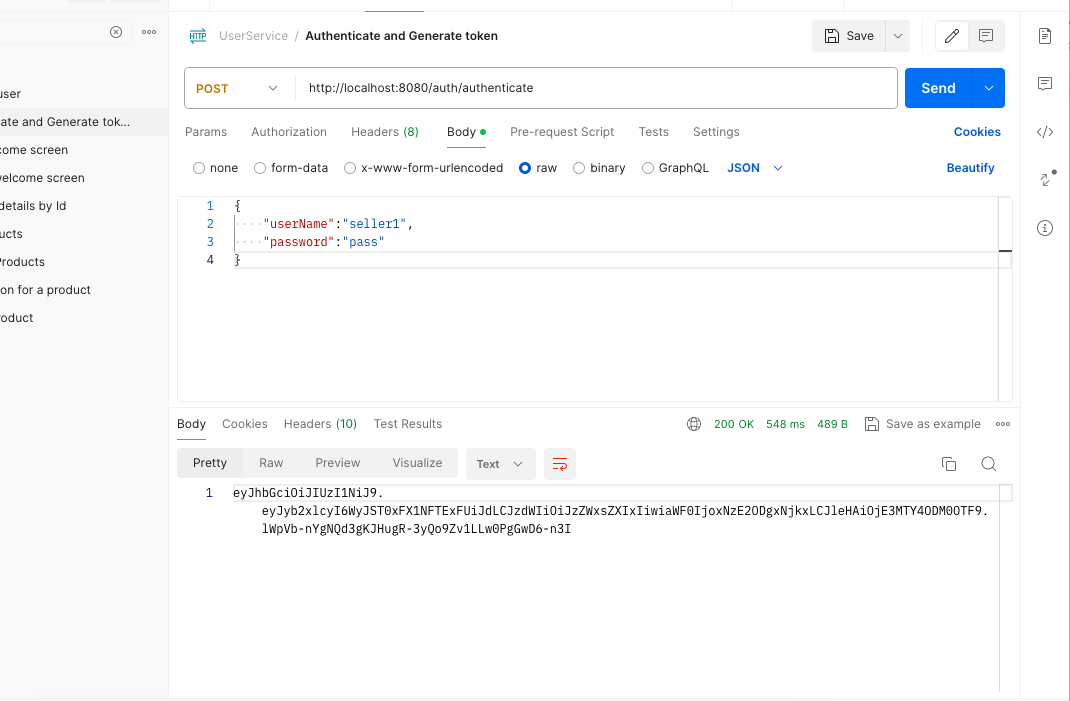
ROLES-->ROLE\_SELLER

<http://localhost:8080/auth/register>



1. User Authenticate API to generate JWT\_TOKEN - SAve the token for each user in notedpad

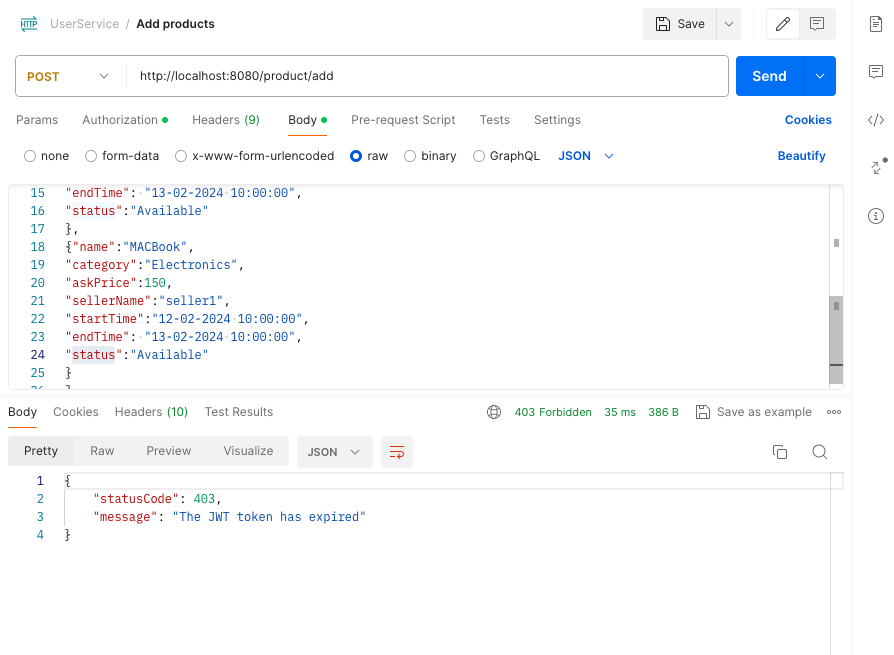
<http://localhost:8080/auth/authenticate>



Create 2 users with role ROLE\_BUYER (can create more users)

\*\*Save token for ech of them in local texteditor for use, pass the token for corresponding sellerName/buyerId is in request body

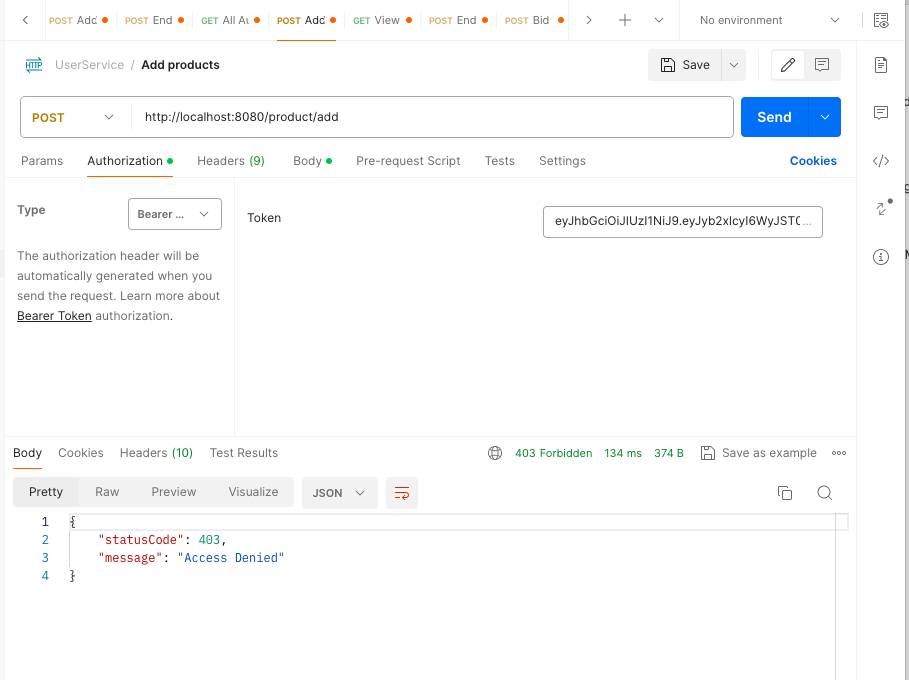
1. For each API request excluding registering a user and authentication we need to pass created JWT token as type Authentication and Bearer <tokenValue>
2. Invlaid or Expired token gives appropriate error message



1. Add products API

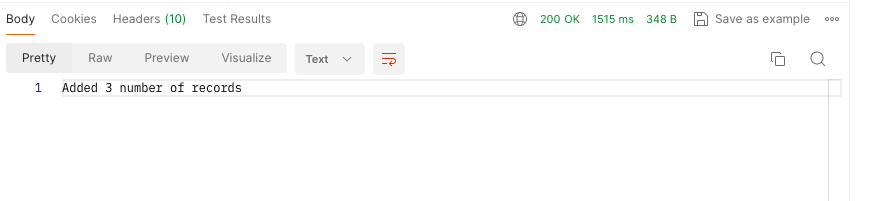
http://localhost:8080/product/add

* 1. Only seller can add products try addin BUYER token and calling API returns --> 403 Access Denied



* 1. Add seller token and call API, products gets added

Returns message



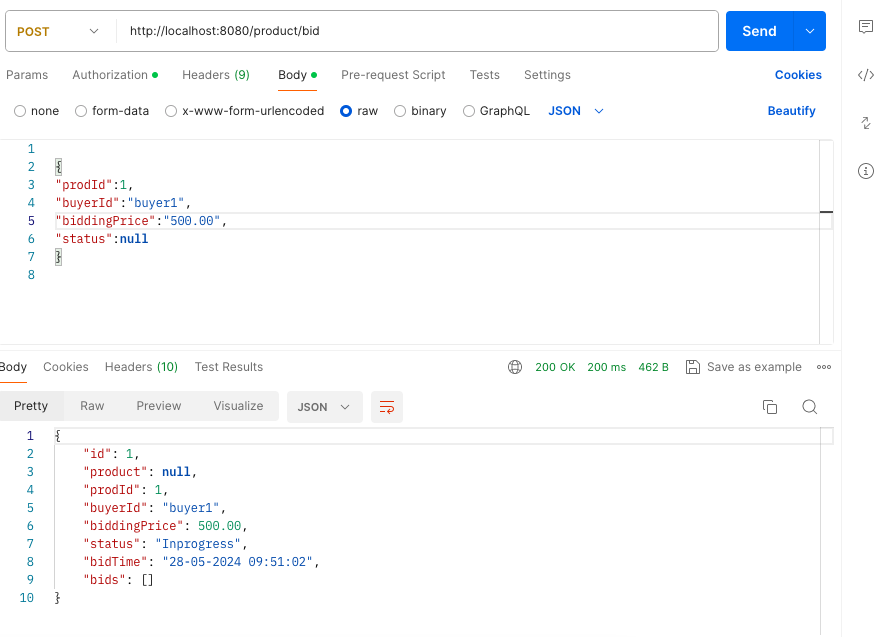
1. View Products -All users can acces this API

http://localhost:8080/product/all

1. Bid on a product.User with role BUYER can place bid.Pass BUYER token in authentication header.

http://localhost:8080/product/bid

* 1. Invalid user role gives 403 Access denied
  2. Valid token place a bid with bid request has user name, product id as main details, returns 200 ok with bid added in response body.



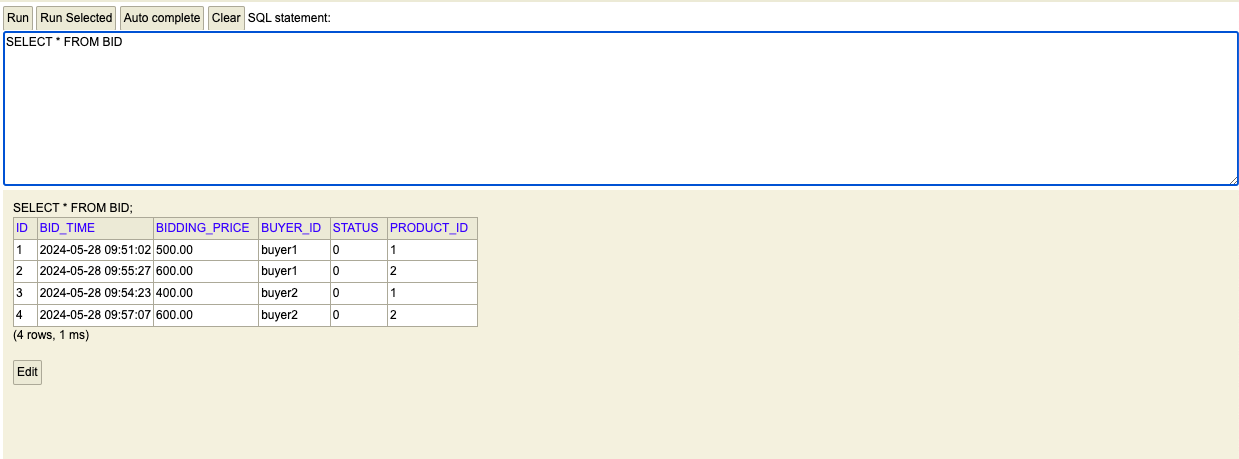
Database check all bids

1. End bid on product by passing product id .Only a SELLER can end a bid.

http://localhost:8080/product/end/{id}

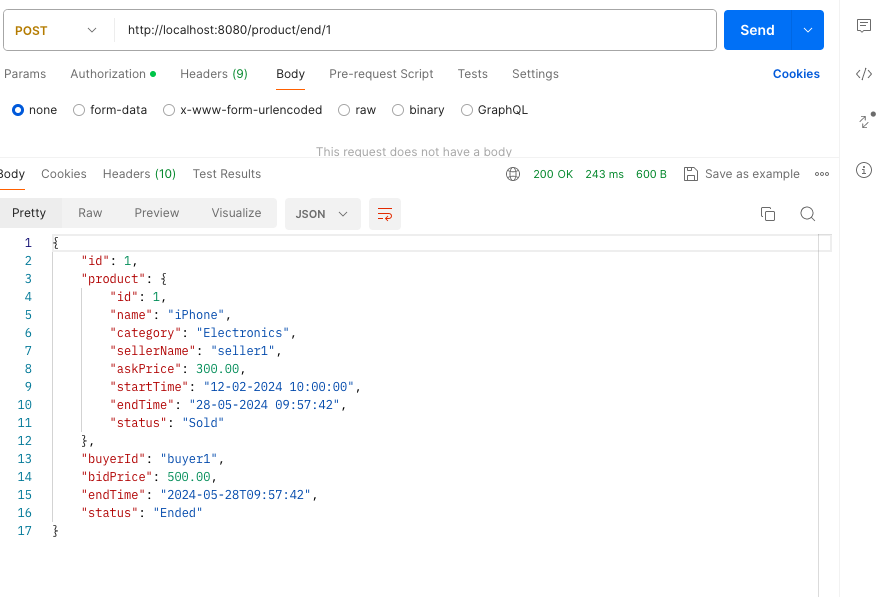
* 1. Invalid user role gives 403 Access denied
  2. End a bid
     1. Check is Auction is present and INPROGESS for given product
     2. Check if product is AVAILABLE
     3. Fetch all bids and pick the one with highest price
     4. If more than one bid with same price pick one which was places first
     5. Fill the name of buyer as buyer Id and bidding price he placed
     6. Return details to seller

Databse entry

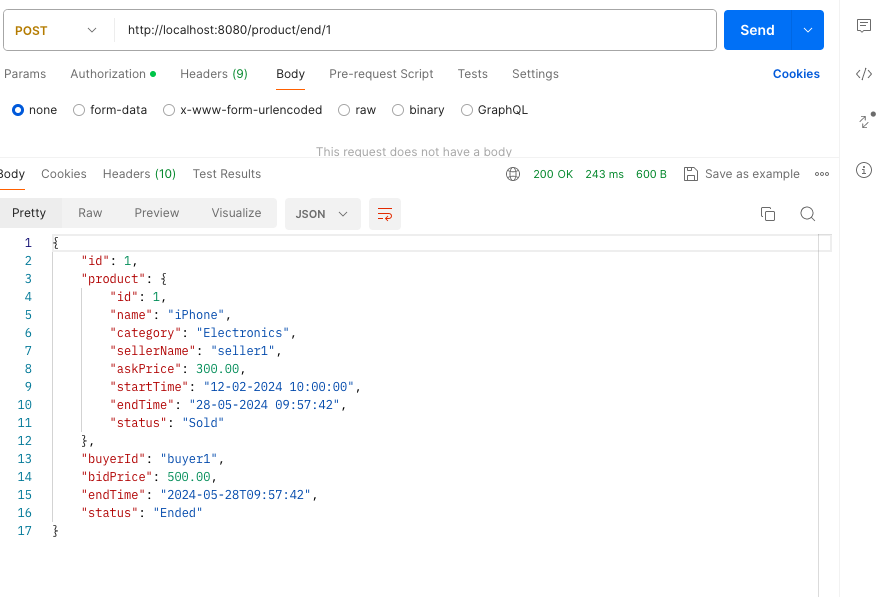


End for product sample output

different bidding price



Returns earliest bid in case same bidding price



DB Details

Auction service

<http://localhost:8081/h2-console/l>

username:sa

password:<no password>

User Service

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

username:sa

password:password