API Testing for LotService

Public endpoint: https://api.gronogolsen.dk
To test the LotController endpoints in sequence, follow the steps below using Postman or a similar API testing tool. Ensure you have a running instance of the LotService API.

Step 0: Create a user

Call POST http://localhost:2999/user/create with the following body request, replace email with your own to test email functionality:

```
{
  "Id": "myuserid123",
  "FirstName": "YourName",
  "LastName": "YourLastName",
  "Email": "youremailhere@gronogolsen.com",
  "Username": "username123",
  "Password": "password123",
  "Address": "789 Oak St, Springfield, USA",
  "PhoneNumber": "555-654-3210",
  "Verified": true
}
```

Step 0.5: Delete Lot

Make a DELETE request to http://localhost:2999/lot/6645e7795a52da654bd15566. To make sure no lot is currently existing

Step 1: Create Lot

Make a POST request to http://localhost:2999/lot. Set the request header Content-Type to application/json. The body of the request should be:

```
{
"lotId": "6645e7795a52da654bd15566",
"LotName": "Antique Painting",
"Location": "New York",
"OnlineAuction": true,
"StartingPrice": 100,
"MinimumBid": 10,
"LotCreationTime": "2023-01-01T00:00:00Z",
"LotEndTime": "2028-01-02T00:00:00Z",
"Open": true
}
```

Step 2: Get All Lots

Make a GET request to http://localhost:2999/lot. You should receive a response containing a list of LotModel objects.

Step 3: Get Lot by ID

Make a GET request to http://localhost:2999/lot/6645e7795a52da654bd15566. You should receive a response with the requested LotModel object.

Step 4: Update Lot

Make a PUT request to http://localhost:2999/lot. Set the request header Content-Type to application/json. The body of the request should be:

```
{
"LotId": "6645e7795a52da654bd15566",
"LotName": "Antique Painting",
"Location": "New York",
"OnlineAuction": true,
"StartingPrice": 150,
"MinimumBid": 10,
"LotCreationTime": "2023-01-01T00:00:00Z",
"LotEndTime": "2029-01-02T00:00:00Z",
"Open": true
}
```

You should receive a response containing the updated LotModel object.

Step 5: Update Lot Price

Make a PUT request to http://localhost:2999/lot/bid. Set the request header Content-Type to application/json. The body of the request should be:

```
{
"Amount": 200,
"BidderId": "myuserid123",
"LotId": "6645e7795a52da654bd15566",
"Timestamp": "2023-01-01T01:00:00Z"
}
```

You should receive a response containing the BidModel object.

Step 6: Close Lot

Make a PUT request to http://localhost:2999/lot/close. The body of the request should be:

```
"6645e7795a52da654bd15566"
```

You should receive a response containing the updated LotModel object with the status changed to closed.

Step 6.5

Receive the emails (if you inputted your own email) and follow the email button flow, so receive the invoice, “pay” it, and receive the parcel tracking link

Step 7: Delete Lot

Make a DELETE request to http://localhost:2999/lot/6645e7795a52da654bd15566. You should receive a response confirming the deletion of the lot.