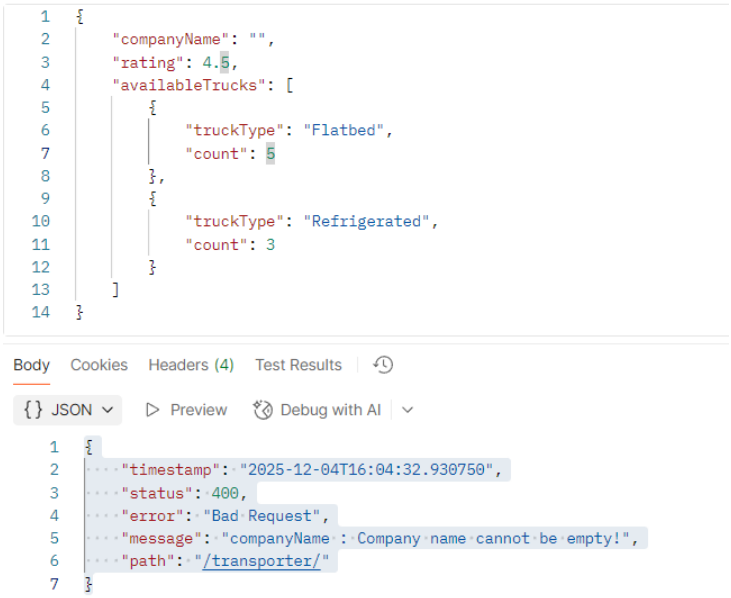
TESTING DOCUMENT

Transporter:

1] Invalid data:

1. Without company name Request
   Throws a custom Exception
   Request and Response is shown in the below image



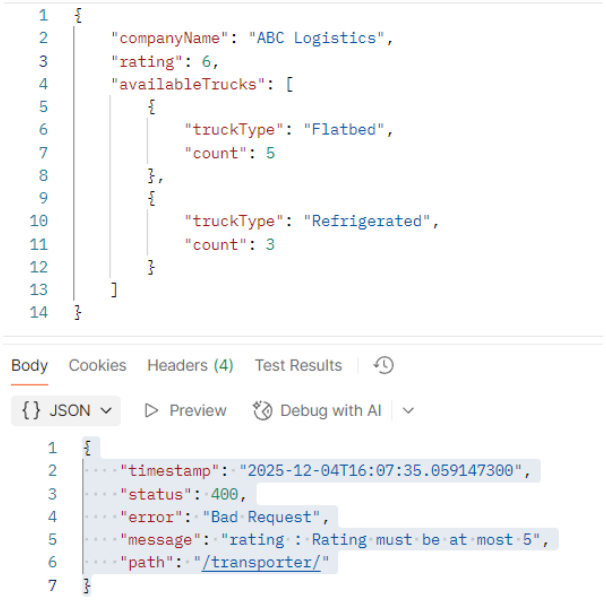
```
{
    "companyName": "",
    "rating": 4.5,
    "availableTrucks": [
        {
            "truckType": "Flatbed",
            "count": 5
        },
        {
            "truckType": "Refrigerated",
            "count": 3
        }
    ]
}
```

Body Cookies Headers (4) Test Results

{} JSON ▷ Preview Debug with AI

```
{
    "timestamp": "2025-12-04T16:04:32.930750",
    "status": 400,
    "error": "Bad Request",
    "message": "companyName : Company name cannot be empty!",
    "path": "/transporter/"
}
```

2. Rating out of boundary conditions
   Both negative and rating over 5 is shown in below images Both conditions request and Response is shown below.

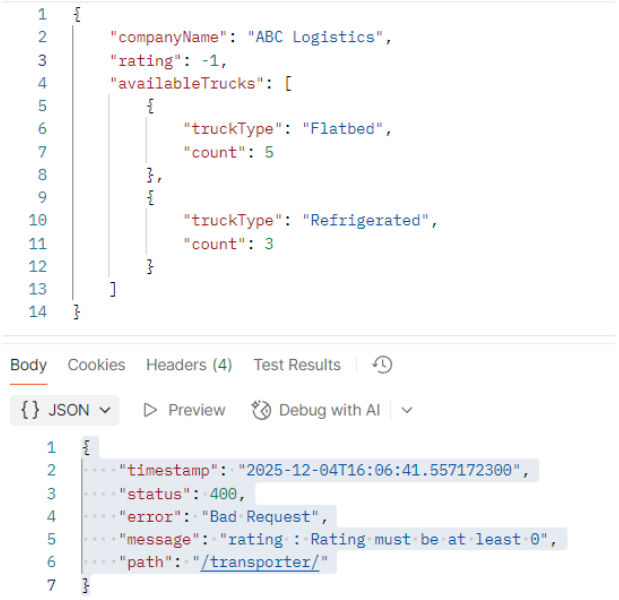


```
{
    "companyName": "ABC Logistics",
    "rating": 6,
    "availableTrucks": [
        {
            "truckType": "Flatbed",
            "count": 5
        },
        {
            "truckType": "Refrigerated",
            "count": 3
        }
    ]
}
```

Body Cookies Headers (4) Test Results

{} JSON ▷ Preview Debug with AI

```
{
    "timestamp": "2025-12-04T16:07:35.059147300",
    "status": 400,
    "error": "Bad Request",
    "message": "rating : Rating must be at most 5",
    "path": "/transporter/"
}
```



```
{
    "companyName": "ABC Logistics",
    "rating": -1,
    "availableTrucks": [
        {
            "truckType": "Flatbed",
            "count": 5
        },
        {
            "truckType": "Refrigerated",
            "count": 3
        }
    ]
}
```

Body Cookies Headers (4) Test Results

{} JSON ▷ Preview Debug with AI

```
{
    "timestamp": "2025-12-04T16:06:41.557172300",
    "status": 400,
    "error": "Bad Request",
    "message": "rating : Rating must be at least 0",
    "path": "/transporter/"
}
```

Create Transporter:

Post method → URL: http://localhost:8080/transporter/

2] Valid data: creating Transporter1 (1st company)

Request:

```
{
    "companyName": "ABC Logistics",
    "rating": 4.5,
    "availableTrucks": [
        {
            "truckType": "Flatbed",
            "count": 5
        },
        {
            "truckType": "Refrigerated",
            "count": 3
        }
    ]
}
```

Response:

```
{
    "message": "transporter created successfully!",
    "data": {
        "transporterId": "9d0f4031-9781-4084-b901-d6d2bdfd0057",
        "companyName": "ABC Logistics",
        "rating": 4.5,
        "availableTrucks": [
            {
                "id": 1,
                "truckType": "Flatbed",
                "count": 5
            },
            {
                "id": 2,
                "truckType": "Refrigerated",
                "count": 3
            }
        ]
    },
    "status": "CREATED"
}
```

creating Transporter2 as 2nd Company without truck Details

Post method → **URL:** http://localhost:8080/transporter/

```
{
    "companyName": "XYZ Logistics",
    "rating": 4.5
}
```

Body Cookies Headers (5) Test Results 201 Created 34 ms 365 B

{} JSON Preview Visualize

```
{
    "message": "transporter created successfully!",
    "data": {
        "transporterId": "f0ff8f74-810a-4563-b2b4-1c507755b16e",
        "companyName": "XYZ Logistics",
        "rating": 4.5,
        "availableTrucks": null
    },
    "status": "CREATED"
}
```

UpdatAvailableTrucks:

Request Method: PUT

URL: http://localhost:8080/transporter/{mention here transporterId} /trucks

Updating truck details to the transporter 2 (2nd company)

Request:

```
[
    {
        "truckType": "Flatbed",
        "count": 5
    },
    {
        "truckType": "Refrigerated",
        "count": 3
    }
]
```

Response:

```
{
    "message": "Available trucks updated successfully!",
    "data": {
        "transporterId": "f0ff8f74-810a-4563-b2b4-1c507755b16e",
        "companyName": "XYZ Logistics",
        "rating": 4.5,
        "availableTrucks": [
            {
                "id": 3,
                "truckType": "Flatbed",
                "count": 5
            },
            {
                "id": 4,
                "truckType": "Refrigerated",
                "count": 3
            }
        ]
    },
    "status": "UPDATED"
}
```

Updating Existing Trucks count:

Using GetTransporter check the transporter details first.

Request and Response: Before updating the transporter.

Body Cookies Headers (5) Test Results

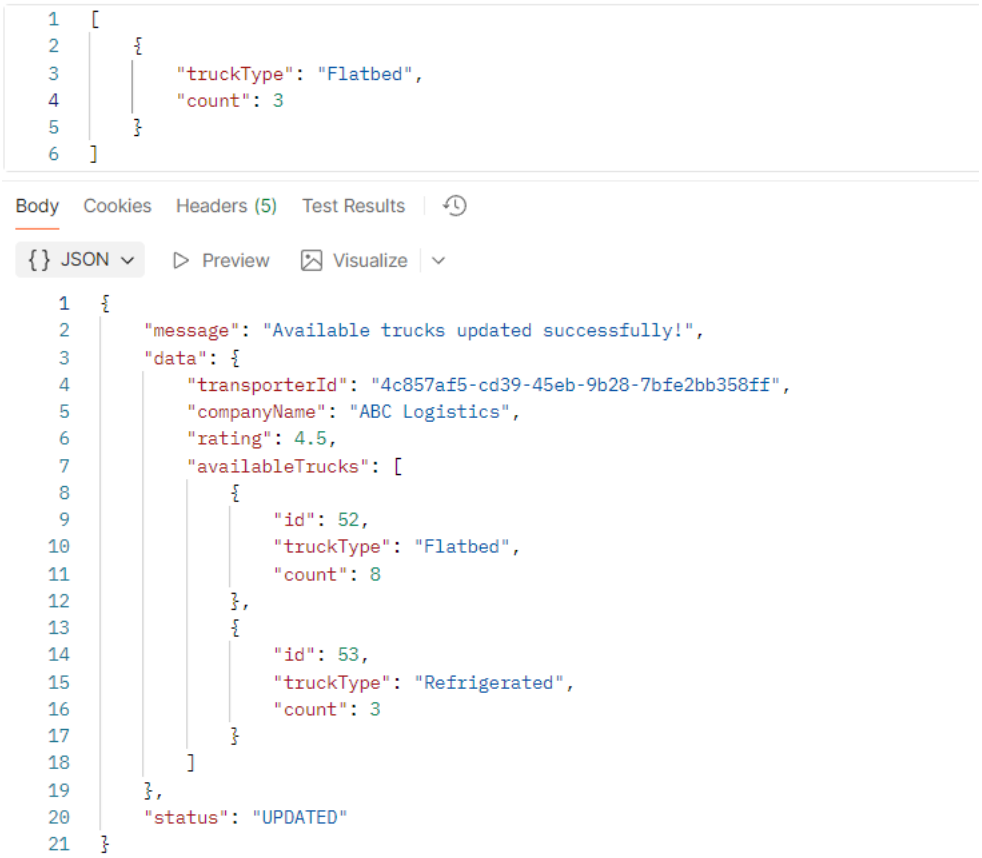
{} JSON Preview Visualize

```
{
    "transporterId": "4c857af5-cd39-45eb-9b28-7bfe2bb358ff",
    "companyName": "ABC Logistics",
    "rating": 4.5,
    "availableTrucks": [
        {
            "id": 52,
            "truckType": "Flatbed",
            "count": 5
        },
        {
            "id": 53,
            "truckType": "Refrigerated",
            "count": 3
        }
    ]
}
```

Then update specific truck count and check

Request and Response:

Request and Response show below after updating.



```
[
    {
        "truckType": "Flatbed",
        "count": 3
    }
]
```

Body Cookies Headers (5) Test Results

{} JSON Preview Visualize

```
{
    "message": "Available trucks updated successfully!",
    "data": {
        "transporterId": "4c857af5-cd39-45eb-9b28-7bfe2bb358ff",
        "companyName": "ABC Logistics",
        "rating": 4.5,
        "availableTrucks": [
            {
                "id": 52,
                "truckType": "Flatbed",
                "count": 8
            },
            {
                "id": 53,
                "truckType": "Refrigerated",
                "count": 3
            }
        ]
    },
    "status": "UPDATED"
}
```

Creating 3rd Transporter as Transporter3 (3rd company) Post method → URL: http://localhost:8080/transporter/

Request:

```
{
    "companyName": "MNC Logistics",
    "rating": 4.5,
    "availableTrucks": [
        {
            "truckType": "Flatbed",
            "count": 1
        },
        {
            "truckType": "Refrigerated",
            "count": 1
        }
    ]
}
```

Response:

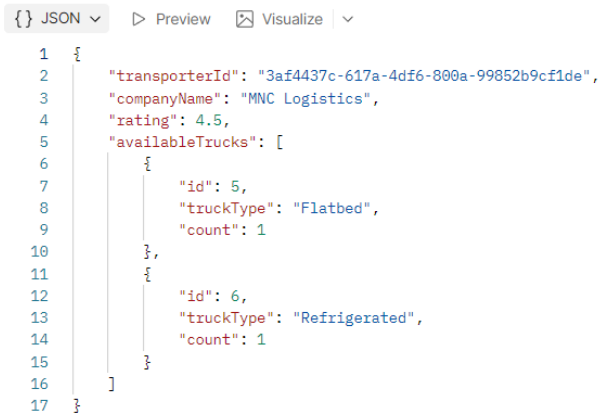
```
{
    "message": "transporter created successfully!",
    "data": {
        "transporterId": "3af4437c-617a-4df6-800a-99852b9cf1de",
        "companyName": "MNC Logistics",
        "rating": 4.5,
        "availableTrucks": [
            {
                "id": 5,
                "truckType": "Flatbed",
                "count": 1
            },
            {
                "id": 6,
                "truckType": "Refrigerated",
                "count": 1
            }
        ]
    },
    "status": "CREATED"
}
```

GetTransporter: getting the transporter using transporter id

Request method: “GET”

URL: http://localhost:8080/transporter/{transporterId}

Response:



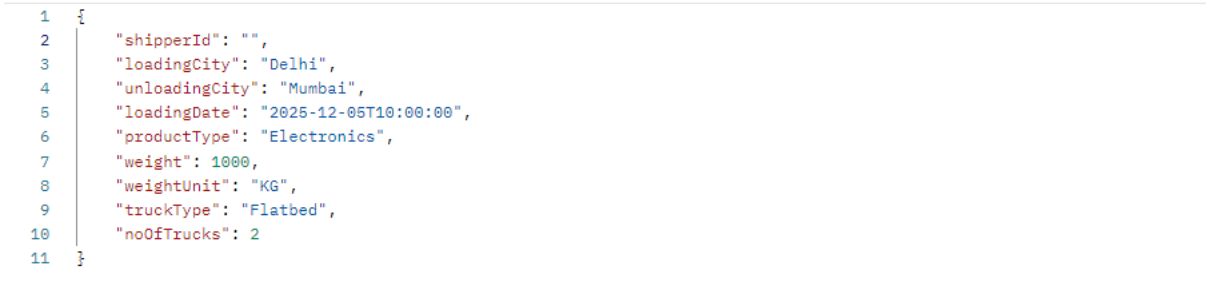
Load:

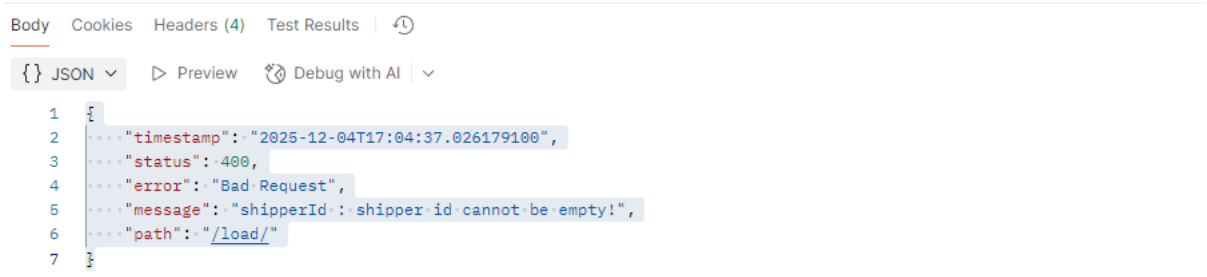
Request Mathod: POST

URL: http://localhost:8080/load/

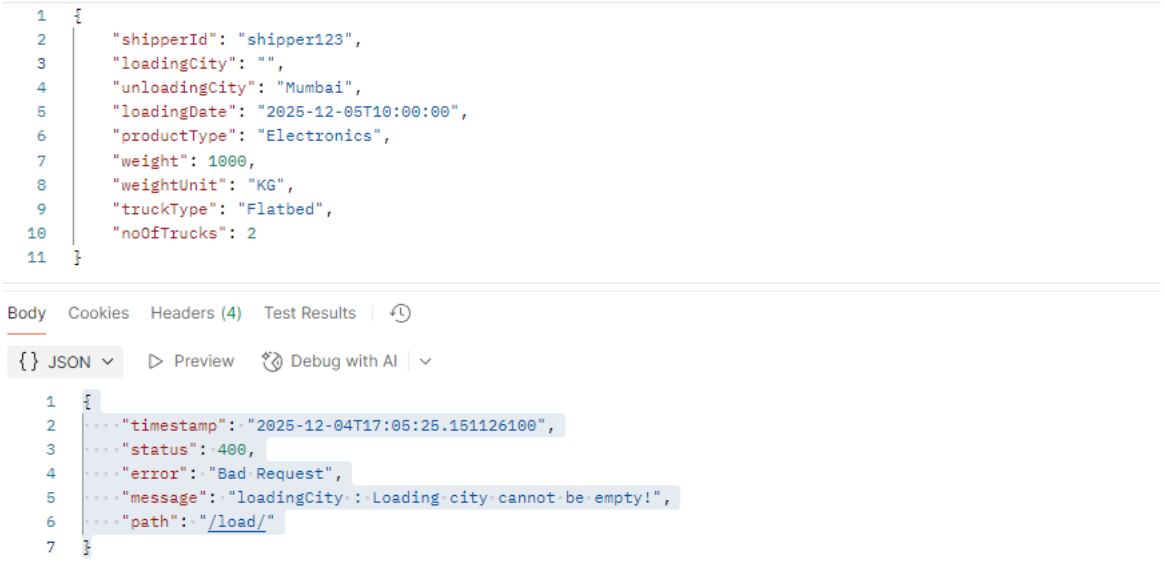
1] Invalid Data:

1. Invalid Shipper Id: Try to create load with blank shipperId or empty.or don’t pass shipperId
   Request and Response is shown below





2. Request without Loading city: without passing loading city try to create the load or keep the loading city blank or empty.
   Request & Response is shown below



```
{
    "shipperId": "shipper123",
    "loadingCity": "",
    "unloadingCity": "Mumbai",
    "loadingDate": "2025-12-05T10:00:00",
    "productType": "Electronics",
    "weight": 1000,
    "weightUnit": "KG",
    "truckType": "Flatbed",
    "noOfTrucks": 2
}
```

Body Cookies Headers (4) Test Results

{} JSON Preview Debug with AI

```
{
    "timestamp": "2025-12-04T17:05:25.151126100",
    "status": 400,
    "error": "Bad Request",
    "message": "loadingCity : Loading city cannot be empty!",
    "path": "/load/"
}
```

3. Request without unloading city: Try to create Load without unloading city. Or keep unloading city blank or empty.
   Request and Response :

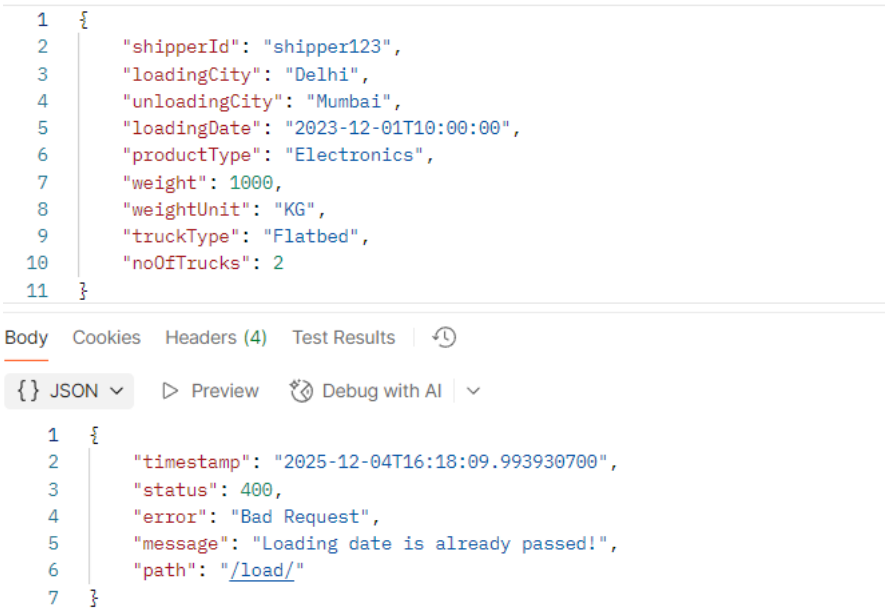
```
{
    "shipperId": "shipper123",
    "loadingCity": "Delhi",
    "unloadingCity": "",
    "loadingDate": "2025-12-05T10:00:00",
    "productType": "Electronics",
    "weight": 1000,
    "weightUnit": "KG",
    "truckType": "Flatbed",
    "noOfTrucks": 2
}
```

Body Cookies Headers (4) Test Results

{} JSON Preview Debug with AI

```
{
    "timestamp": "2025-12-04T17:06:21.124721700",
    "status": 400,
    "error": "Bad Request",
    "message": "unloadingCity : Unloading city cannot be empty!",
    "path": "/load/"
}
```

4. Request with Passed date: try to create Load with passed date as loading time
   Request and Response:



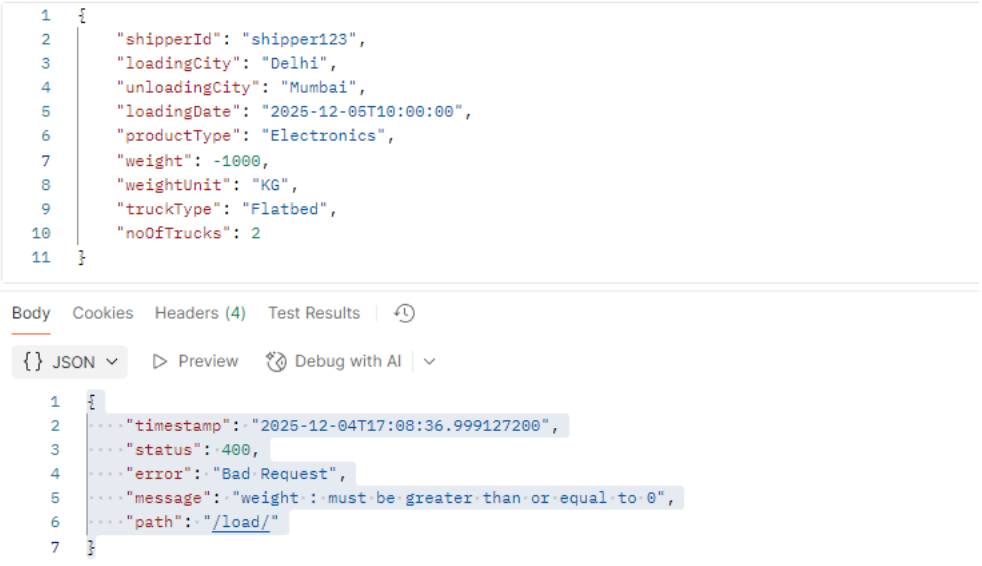
```
{
    "shipperId": "shipper123",
    "loadingCity": "Delhi",
    "unloadingCity": "Mumbai",
    "loadingDate": "2023-12-01T10:00:00",
    "productType": "Electronics",
    "weight": 1000,
    "weightUnit": "KG",
    "truckType": "Flatbed",
    "noOfTrucks": 2
}
```

Body Cookies Headers (4) Test Results

{} JSON ▷ Preview Debug with AI

```
{
    "timestamp": "2025-12-04T16:18:09.993930700",
    "status": 400,
    "error": "Bad Request",
    "message": "Loading date is already passed!",
    "path": "/load/"
}
```

5. Invalid weight: try to create with invalid weight like negative
   Request and Response:



```
{
    "shipperId": "shipper123",
    "loadingCity": "Delhi",
    "unloadingCity": "Mumbai",
    "loadingDate": "2025-12-05T10:00:00",
    "productType": "Electronics",
    "weight": -1000,
    "weightUnit": "KG",
    "truckType": "Flatbed",
    "noOfTrucks": 2
}
```

Body Cookies Headers (4) Test Results

{} JSON ▷ Preview Debug with AI

```
{
    "timestamp": "2025-12-04T17:08:36.999127200",
    "status": 400,
    "error": "Bad Request",
    "message": "weight : must be greater than or equal to 0",
    "path": "/load/"
}
```

6. Invalid weight Unit: try to create a load with invalid weight unit.
   Request and Response

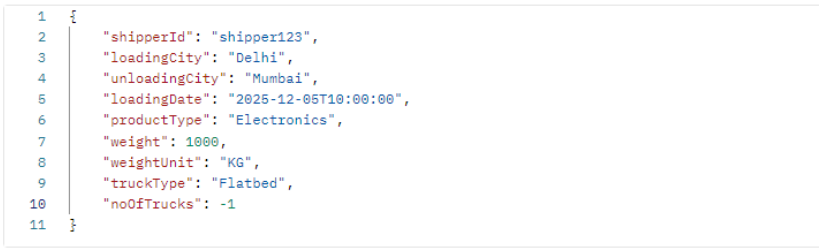
```
{
    "shipperId": "shipper123",
    "loadingCity": "Delhi",
    "unloadingCity": "Mumbai",
    "loadingDate": "2025-12-05T10:00:00",
    "productType": "Electronics",
    "weight": 1000,
    "weightUnit": "MG",
    "truckType": "Flatbed",
    "noOfTrucks": 2
}
```

Body Cookies Headers (4) Test Results 400 Bad Request 18 ms 412 B Save Res

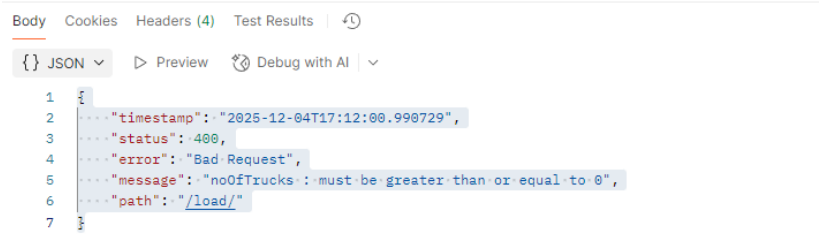
{} JSON ▷ Preview Debug with AI

```
{
    "timestamp": "2025-12-04T11:40:08.075+00:00",
    "status": 400,
    "error": "Bad Request",
    "message": "JSON parse error: Cannot deserialize value of type `com.tmc.model.load.WeightUnit` from String \"MG\": not one of the values accepted for Enum class: [TON, KG]",
    "path": "/load/"
}
```

7. Invalid noOfTrucks: try creating load with invalid noOfTrucks like negative number.
   Request & Response



```
{
    "shipperId": "shipper123",
    "loadingCity": "Delhi",
    "unloadingCity": "Mumbai",
    "loadingDate": "2025-12-05T10:00:00",
    "productType": "Electronics",
    "weight": 1000,
    "weightUnit": "KG",
    "truckType": "Flatbed",
    "noOfTrucks": -1
}
```



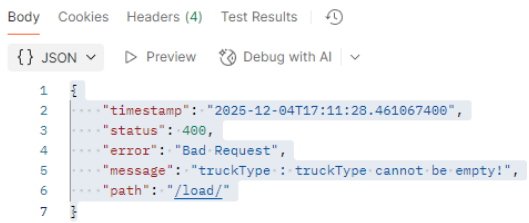
Body Cookies Headers (4) Test Results

{} JSON ▷ Preview Debug with AI

```
{
    "timestamp": "2025-12-04T17:12:00.990729",
    "status": 400,
    "error": "Bad Request",
    "message": "noOfTrucks : must be greater than or equal to 0",
    "path": "/load/"
}
```

8. Invalid truckType: Create load with empty or blank truckType.
   Request and Response

```
{
    "shipperId": "shipper123",
    "loadingCity": "Delhi",
    "unloadingCity": "Mumbai",
    "loadingDate": "2025-12-05T10:00:00",
    "productType": "Electronics",
    "weight": 1000,
    "weightUnit": "KG",
    "truckType": "",
    "noOfTrucks": 2
}
```



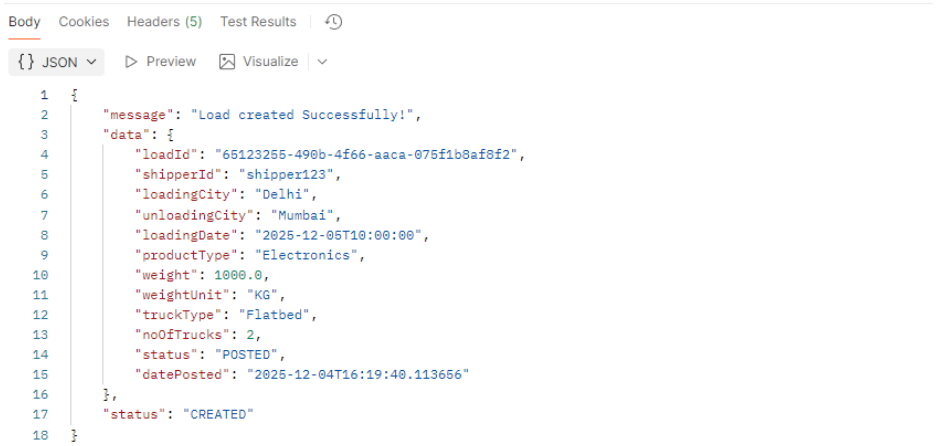
Body Cookies Headers (4) Test Results

{} JSON ▷ Preview Debug with AI

```
{
    "timestamp": "2025-12-04T17:11:28.461067400",
    "status": 400,
    "error": "Bad Request",
    "message": "truckType : truckType cannot be empty!",
    "path": "/load/"
}
```

2] Valid data: create load with valid data.

```
{
    "shipperId": "shipper123",
    "loadingCity": "Delhi",
    "unloadingCity": "Mumbai",
    "loadingDate": "2025-12-05T10:00:00",
    "productType": "Electronics",
    "weight": 1000,
    "weightUnit": "KG",
    "truckType": "Flatbed",
    "noOfTrucks": 2
}
```



Body Cookies Headers (5) Test Results

{} JSON ▷ Preview Visualize

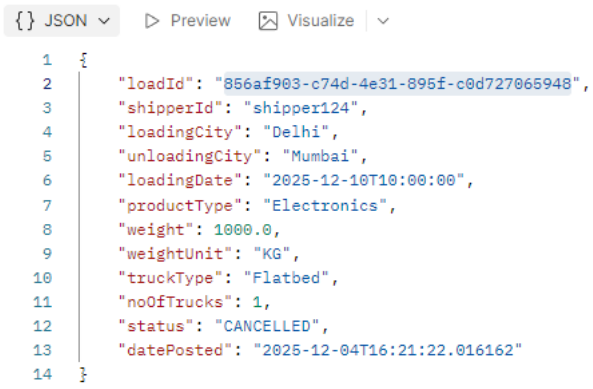
```
{
    "message": "Load created Successfully!",
    "data": {
        "loadId": "65123255-490b-4f66-aaca-075f1b8af8f2",
        "shipperId": "shipper123",
        "loadingCity": "Delhi",
        "unloadingCity": "Mumbai",
        "loadingDate": "2025-12-05T10:00:00",
        "productType": "Electronics",
        "weight": 1000.0,
        "weightUnit": "KG",
        "truckType": "Flatbed",
        "noOfTrucks": 2,
        "status": "POSTED",
        "datePosted": "2025-12-04T16:19:40.113656"
    },
    "status": "CREATED"
}
```

getLoad:

RequestMethod: GET

URL: http://localhost:8080/load/{loadId}

Response:

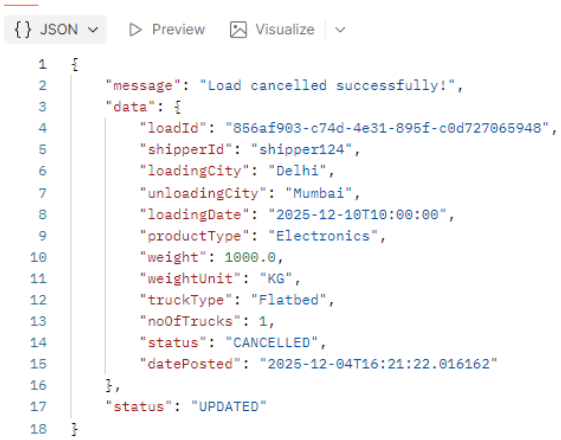


cancelLoad:

RequestMethod: PATCH

URL: http://localhost:8080/load/{loadId} /cancel

Response:

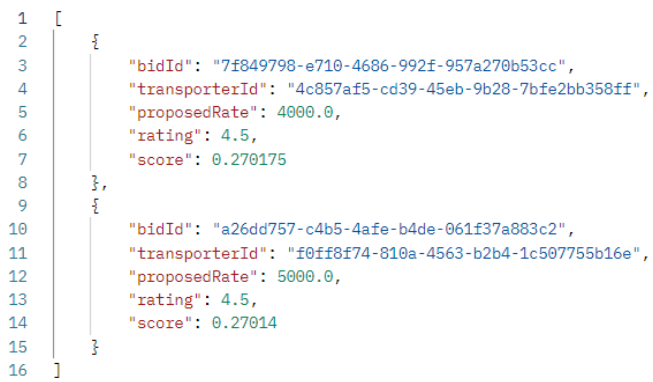


cancelLoad:

RequestMethod: GET

URL: http://localhost:8080/load/{loadId} /best-bids

Response:



If bids are not present for particular load then.

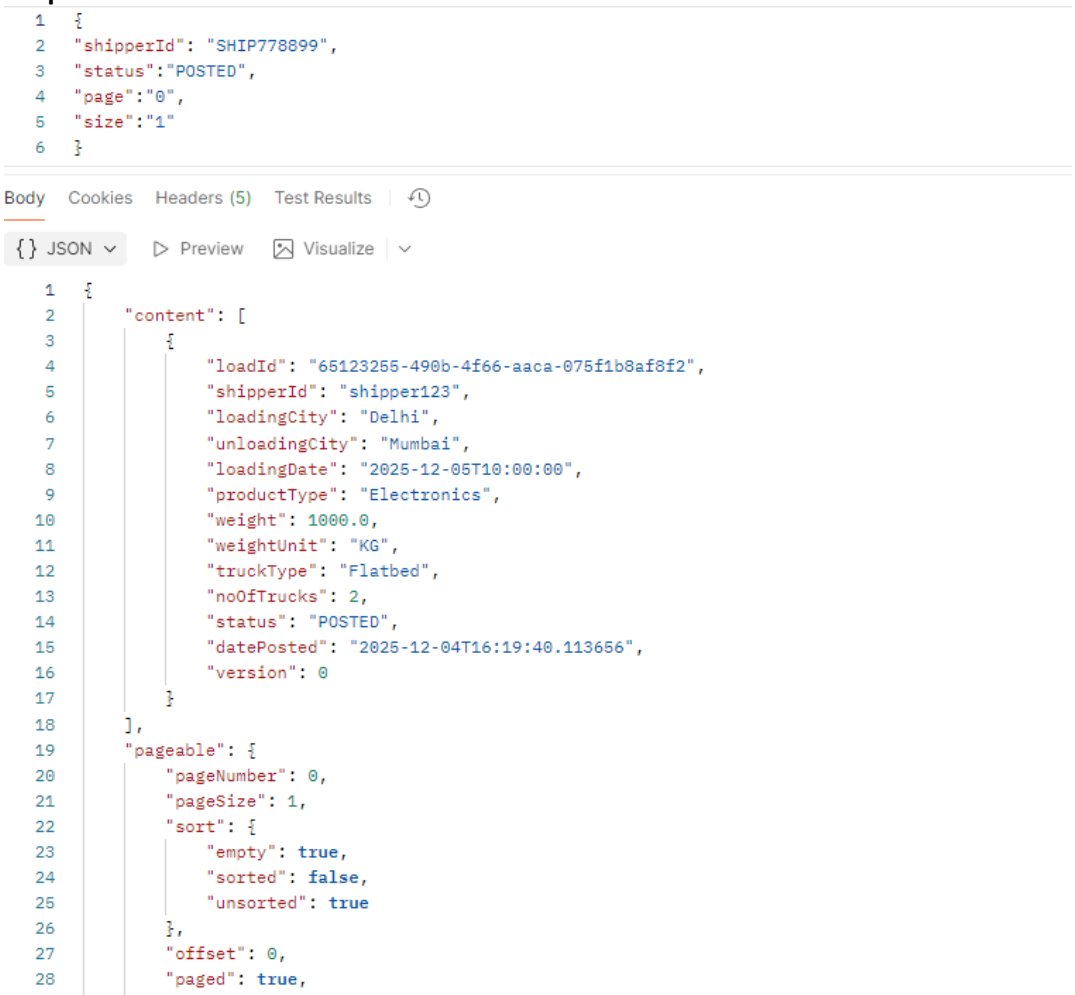
Response:

GetLoadsPaginated: Filtered on any one parameter either shipperId or status.

RequestMethod: GET

URL: http://localhost:8080/load/

Request and Response:



Bid:

CreateBid

RequestMethod: POST

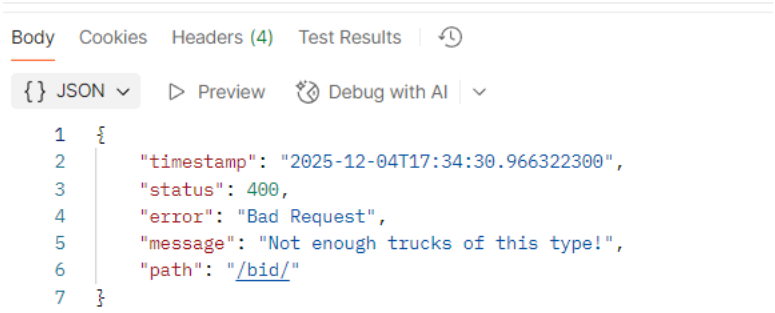
URL: http://localhost:8080/bid/

1] Invalid data:

1. InsufficientTrucksOffered: try creating bid with trucks offered more than available trucks of transporter.

Request and Response:

```
{
    "loadId": "65123255-490b-4f66-aaca-075f1b8af8f2",
    "transporterId": "3af4437c-617a-4df6-800a-99852b9cf1de",
    "proposedRate": 4000,
    "trucksOffered": 10
}
```



Body Cookies Headers (4) Test Results

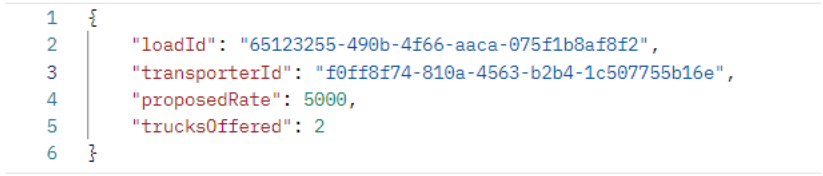
{} JSON ▷ Preview Debug with AI

```
{
    "timestamp": "2025-12-04T17:34:30.966322300",
    "status": 400,
    "error": "Bad Request",
    "message": "Not enough trucks of this type!",
    "path": "/bid/"
}
```

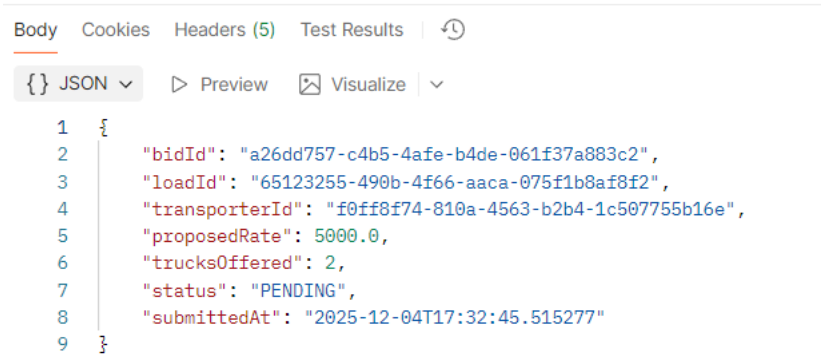
Valid data: create bid for one load

Bid1:

Request and Response:



```
{
    "loadId": "65123255-490b-4f66-aaca-075f1b8af8f2",
    "transporterId": "f0ff8f74-810a-4563-b2b4-1c507755b16e",
    "proposedRate": 5000,
    "trucksOffered": 2
}
```



Body Cookies Headers (5) Test Results

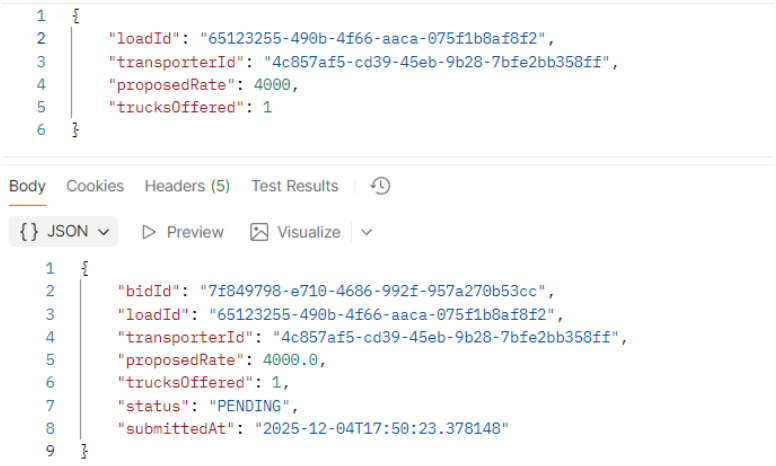
{} JSON ▷ Preview Visualize

```
{
    "bidId": "a26dd757-c4b5-4afe-b4de-061f37a883c2",
    "loadId": "65123255-490b-4f66-aaca-075f1b8af8f2",
    "transporterId": "f0ff8f74-810a-4563-b2b4-1c507755b16e",
    "proposedRate": 5000.0,
    "trucksOffered": 2,
    "status": "PENDING",
    "submittedAt": "2025-12-04T17:32:45.515277"
}
```

Bid2:

Create 2nd bid for same load

Request and Response:



```
{
    "loadId": "65123255-490b-4f66-aaca-075f1b8af8f2",
    "transporterId": "4c857af5-cd39-45eb-9b28-7bfe2bb358ff",
    "proposedRate": 4000,
    "trucksOffered": 1
}
```

Body Cookies Headers (5) Test Results

{} JSON Preview Visualize


```
{
    "bidId": "7f849798-e710-4686-992f-957a270b53cc",
    "loadId": "65123255-490b-4f66-aaca-075f1b8af8f2",
    "transporterId": "4c857af5-cd39-45eb-9b28-7bfe2bb358ff",
    "proposedRate": 4000.0,
    "trucksOffered": 1,
    "status": "PENDING",
    "submittedAt": "2025-12-04T17:50:23.378148"
}
```


getBid:

RequestMethod: GET

URL: http://localhost:8080/bid/{bidId}

Response:

{} JSON Preview Visualize


```
{
    "bidId": "a26dd757-c4b5-4afe-b4de-061f37a883c2",
    "loadId": "65123255-490b-4f66-aaca-075f1b8af8f2",
    "transporterId": "f0ff8f74-810a-4563-b2b4-1c507755b16e",
    "proposedRate": 5000.0,
    "trucksOffered": 2,
    "status": "PENDING",
    "submittedAt": "2025-12-04T17:32:45.515277"
}
```


GetFilteredBids

RequestMethod: GET

URL: http://localhost:8080/bid/

Response:

```
{
"loadId": "052cf7ac-d35a-410c-b199-45cafacc729c",
"transporterId": "c946d43f-8eae-4714-b99d-3d464182f341",
"status": "PENDING"
}
```

Body Cookies Headers (5) Test Results

{} JSON Preview Visualize


```
[
    {
        "bidId": "a26dd757-c4b5-4afe-b4de-061f37a883c2",
        "loadId": "65123255-490b-4f66-aaca-075f1b8af8f2",
        "transporterId": "f0ff8f74-810a-4563-b2b4-1c507755b16e",
        "proposedRate": 5000.0,
        "trucksOffered": 2,
        "status": "PENDING",
        "submittedAt": "2025-12-04T17:32:45.515277",
        "version": 0
    }
]
```

After checking best bids suggestions, we can choose which bid we want book and using that bidId we will book.

Booking

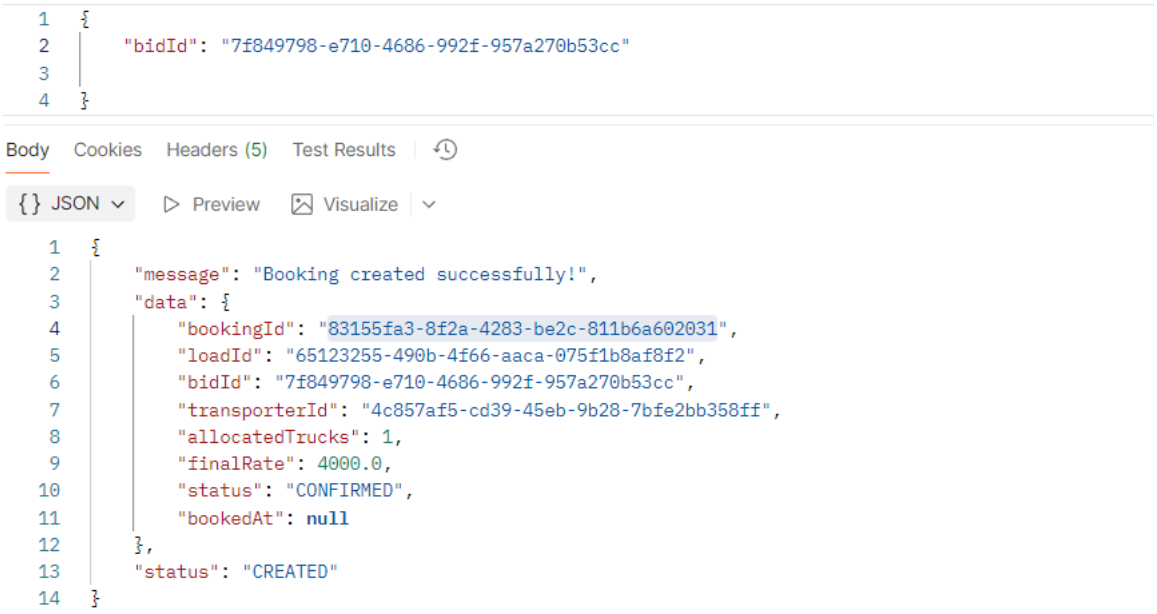
CreateBooking:

RequestMethod: POST

URL: http://localhost:8080/booking/

Request and Response

Booking1:



```
{
    "bidId": "7f849798-e710-4686-992f-957a270b53cc"

}
```

Body Cookies Headers (5) Test Results

{} JSON Preview Visualize

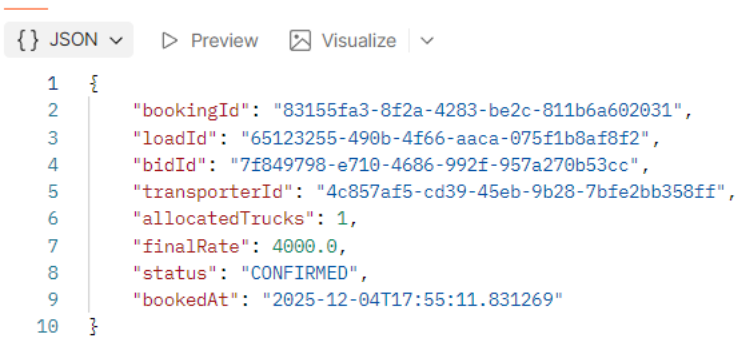

```
{
    "message": "Booking created successfully!",
    "data": {
        "bookingId": "83155fa3-8f2a-4283-be2c-811b6a602031",
        "loadId": "65123255-490b-4f66-aaca-075f1b8af8f2",
        "bidId": "7f849798-e710-4686-992f-957a270b53cc",
        "transporterId": "4c857af5-cd39-45eb-9b28-7bfe2bb358ff",
        "allocatedTrucks": 1,
        "finalRate": 4000.0,
        "status": "CONFIRMED",
        "bookedAt": null
    },
    "status": "CREATED"
}
```


getBooking:

RequestMethod: GET

URL: http://localhost:8080/booking/bookingId

Request and Response



{} JSON Preview Visualize


```
{
    "bookingId": "83155fa3-8f2a-4283-be2c-811b6a602031",
    "loadId": "65123255-490b-4f66-aaca-075f1b8af8f2",
    "bidId": "7f849798-e710-4686-992f-957a270b53cc",
    "transporterId": "4c857af5-cd39-45eb-9b28-7bfe2bb358ff",
    "allocatedTrucks": 1,
    "finalRate": 4000.0,
    "status": "CONFIRMED",
    "bookedAt": "2025-12-04T17:55:11.831269"
}
```

cancelBooking:

RequestMethod: PATCH

URL: http://localhost:8080/booking/bookingId/cancel

Request and Response