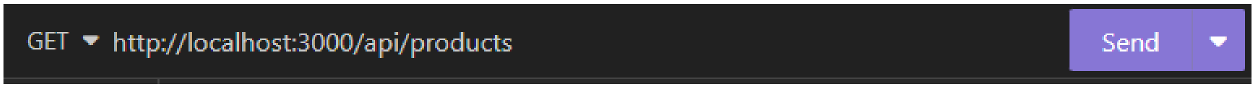
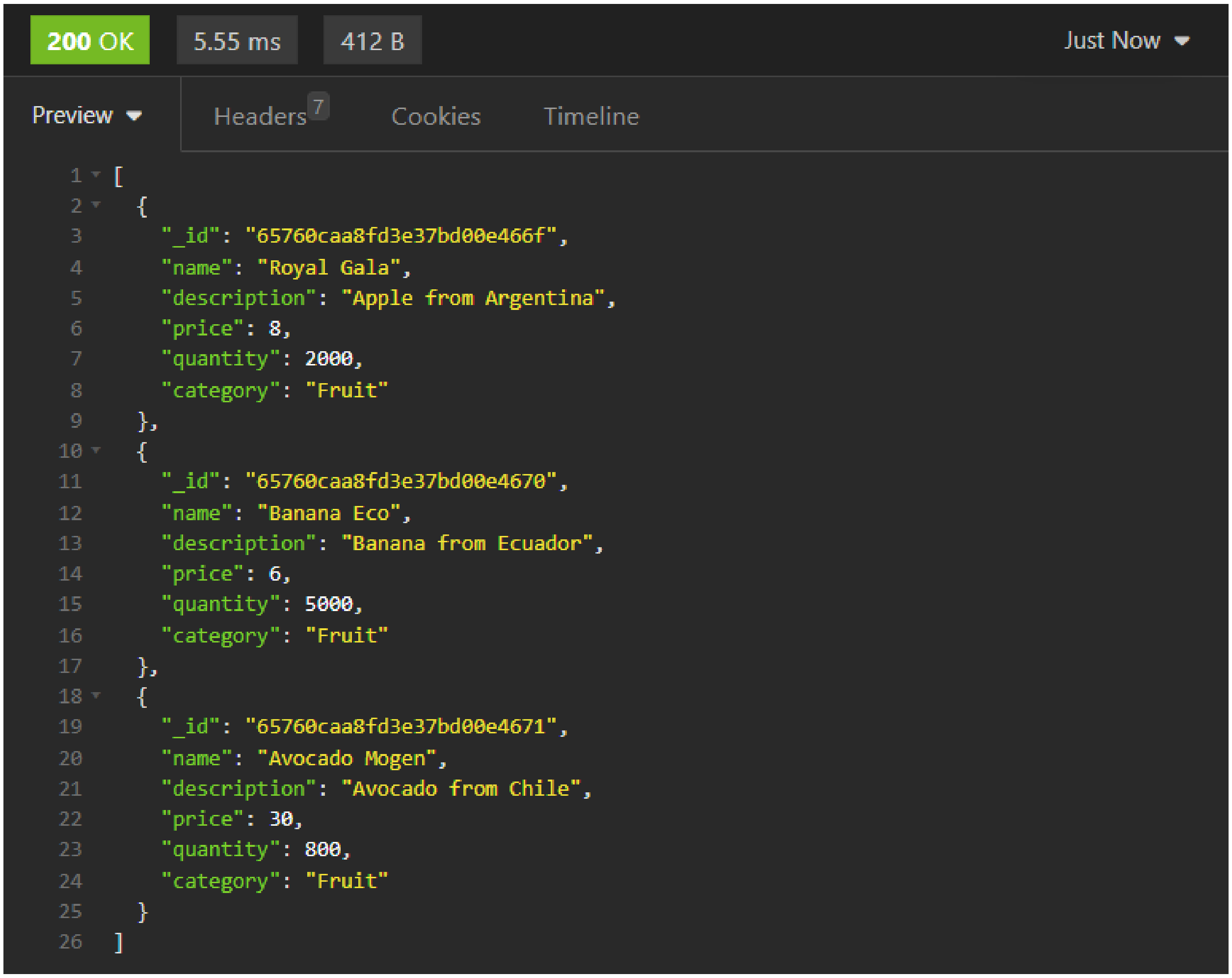
1. To retrieve all the products, make a Get REQUEST to the server.
   GET: /api/products



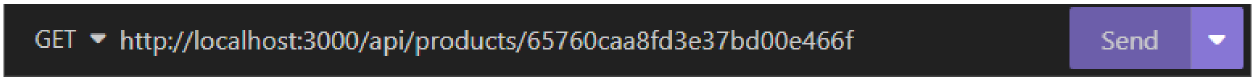
http://localhost:3000/api/products

RESPONSE: 200 OK



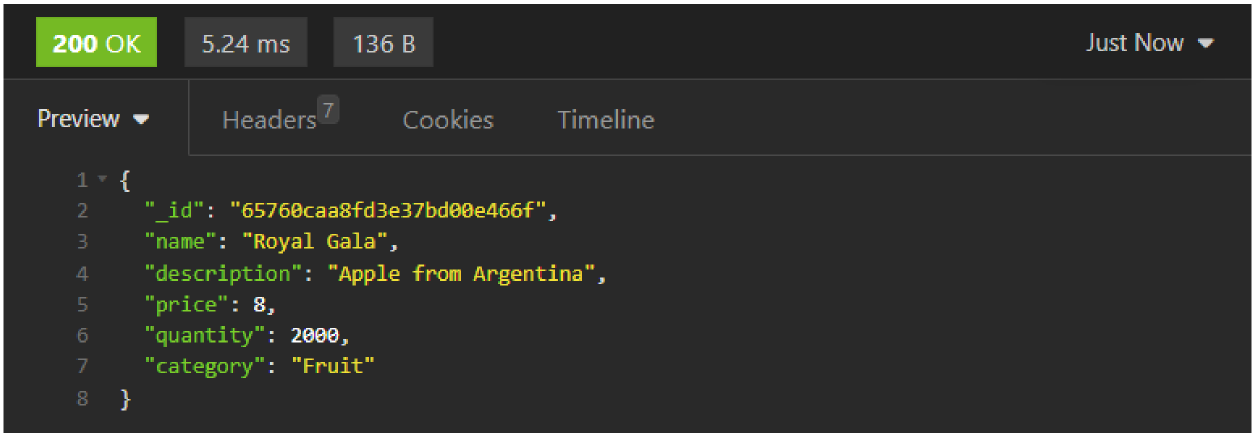
```
[
  {
    "_id": "65760caa8fd3e37bd00e466f",
    "name": "Royal Gala",
    "description": "Apple from Argentina",
    "price": 8,
    "quantity": 2000,
    "category": "Fruit"
  },
  {
    "_id": "65760caa8fd3e37bd00e4670",
    "name": "Banana Eco",
    "description": "Banana from Ecuador",
    "price": 6,
    "quantity": 5000,
    "category": "Fruit"
  },
  {
    "_id": "65760caa8fd3e37bd00e4671",
    "name": "Avocado Mogen",
    "description": "Avocado from Chile",
    "price": 30,
    "quantity": 800,
    "category": "Fruit"
  }
]
```

2. To retrieve a specific product by ID, make a GET REQUEST to the server.
   GET: /api/products/:id



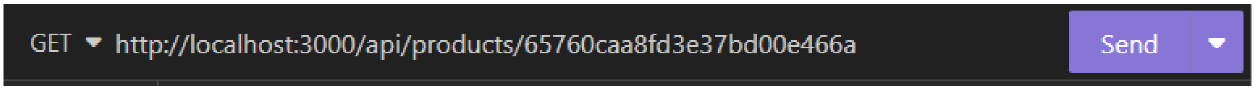
http://localhost:3000/api/products/65760caa8fd3e37bd00e466f

RESPONSE: 200 OK



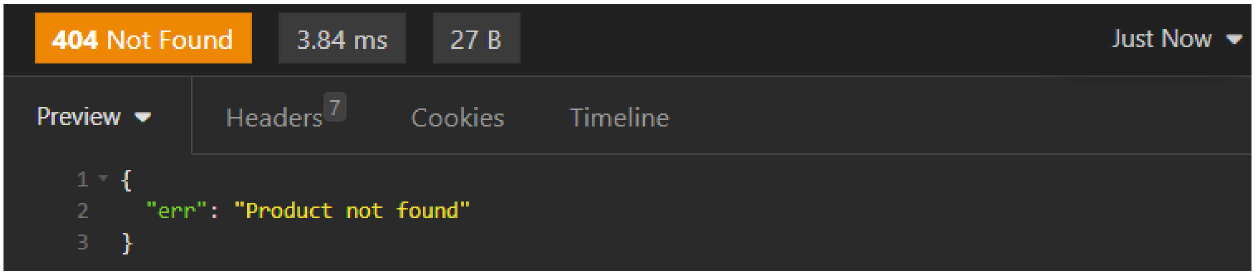
```
{
  "_id": "65760caa8fd3e37bd00e466f",
  "name": "Royal Gala",
  "description": "Apple from Argentina",
  "price": 8,
  "quantity": 2000,
  "category": "Fruit"
}
```

If the product cannot be found or the ID is incorrect when retrieving a specific product by ID:



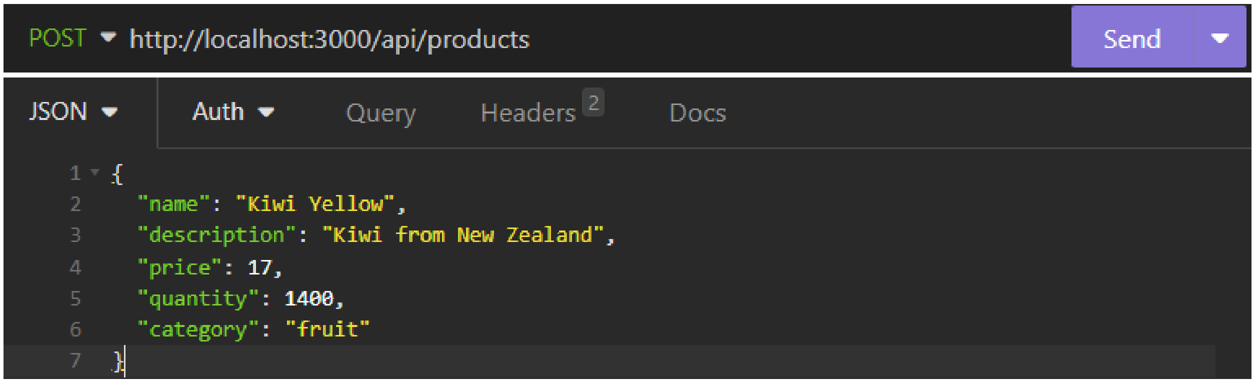
```
http://localhost:3000/api/products/65760caa8fd3e37bd00e466a
```

RESPONSE: 404 Not Found



```
{
  "err": "Product not found"
}
```

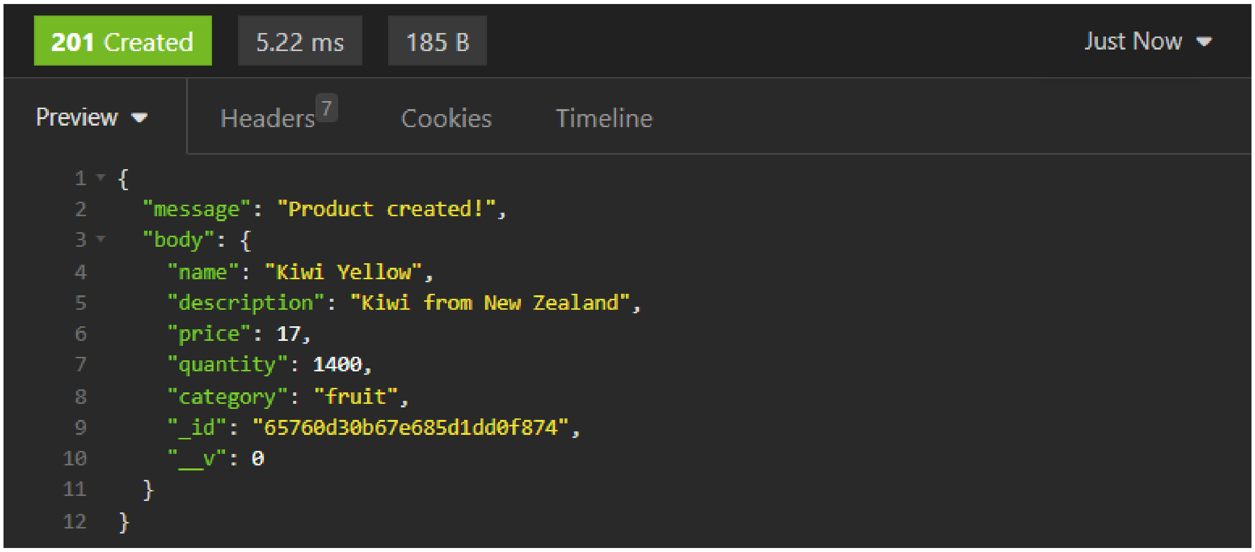
3. To create a new product, make a POST REQUEST to the server.
   POST: /api/products



```
http://localhost:3000/api/products
```

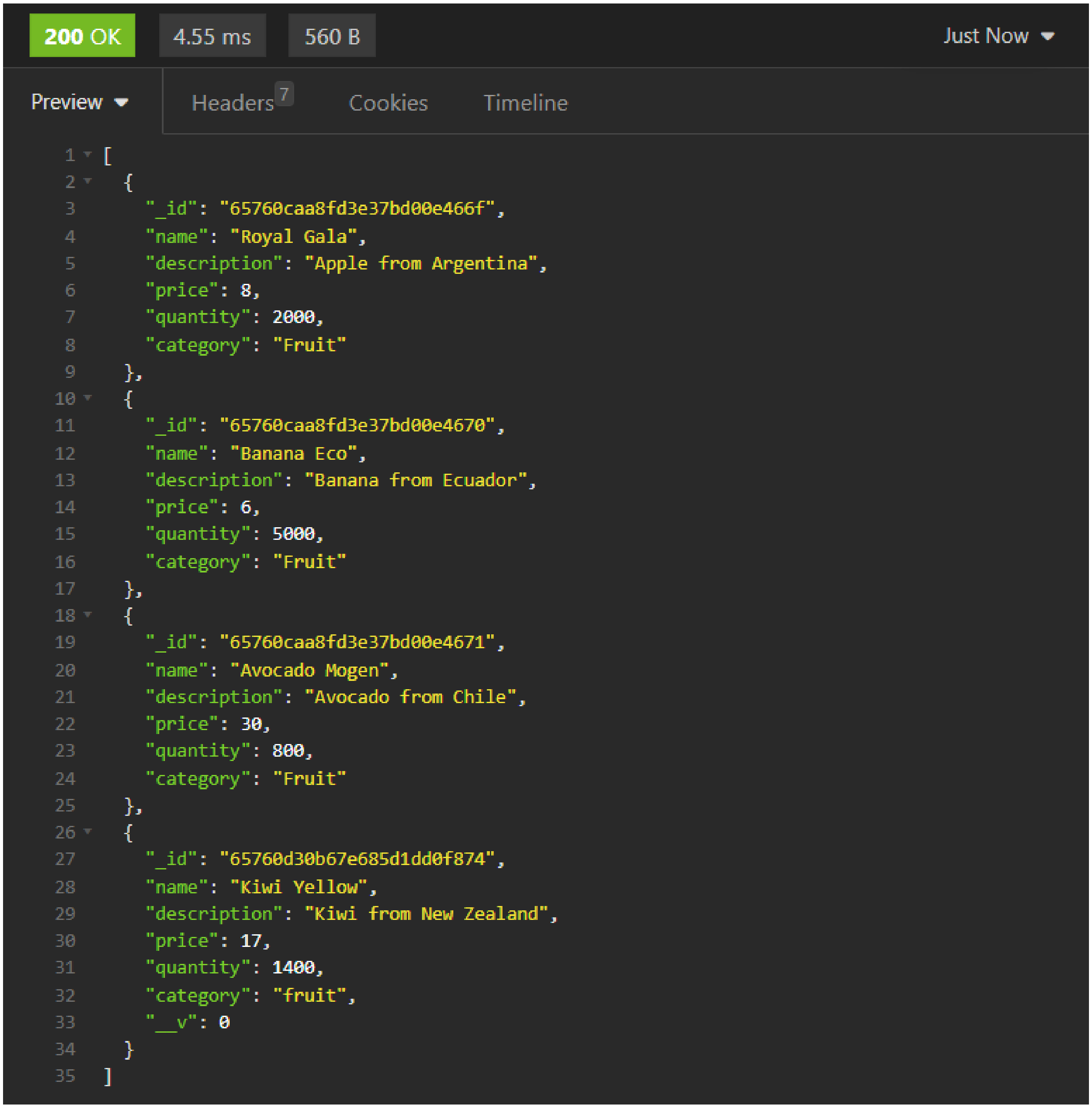
```
{
  "name": "Kiwi Yellow",
  "description": "Kiwi from New Zealand",
  "price": 17,
  "quantity": 1400,
  "category": "fruit"
}
```

RESPONSE: 201 Created



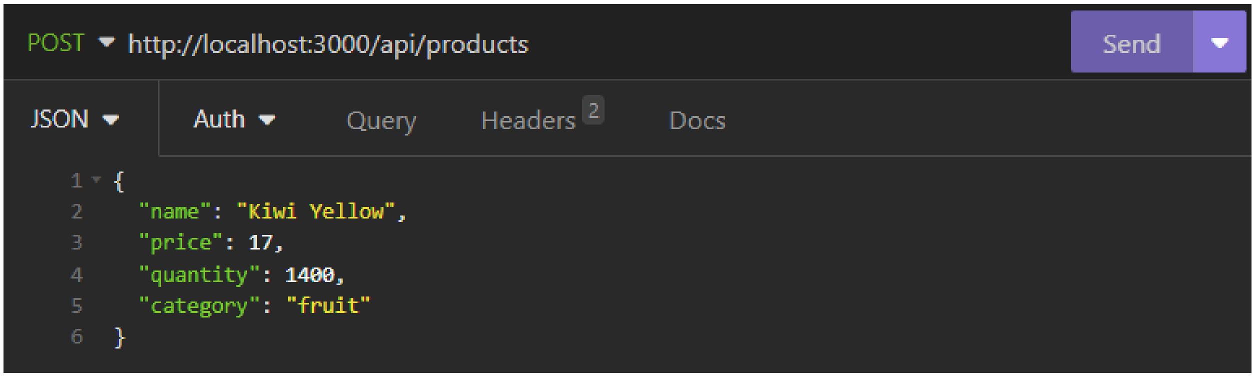
```
{
  "message": "Product created!",
  "body": {
    "name": "Kiwi Yellow",
    "description": "Kiwi from New Zealand",
    "price": 17,
    "quantity": 1400,
    "category": "fruit",
    "_id": "65760d30b67e685d1dd0f874",
    "__v": 0
  }
}
```

Doublecheck with GET all products to make sure the new product is added:



```
[
  {
    "_id": "65760caa8fd3e37bd00e466f",
    "name": "Royal Gala",
    "description": "Apple from Argentina",
    "price": 8,
    "quantity": 2000,
    "category": "Fruit"
  },
  {
    "_id": "65760caa8fd3e37bd00e4670",
    "name": "Banana Eco",
    "description": "Banana from Ecuador",
    "price": 6,
    "quantity": 5000,
    "category": "Fruit"
  },
  {
    "_id": "65760caa8fd3e37bd00e4671",
    "name": "Avocado Mogen",
    "description": "Avocado from Chile",
    "price": 30,
    "quantity": 800,
    "category": "Fruit"
  },
  {
    "_id": "65760d30b67e685d1dd0f874",
    "name": "Kiwi Yellow",
    "description": "Kiwi from New Zealand",
    "price": 17,
    "quantity": 1400,
    "category": "fruit",
    "__v": 0
  }
]
```

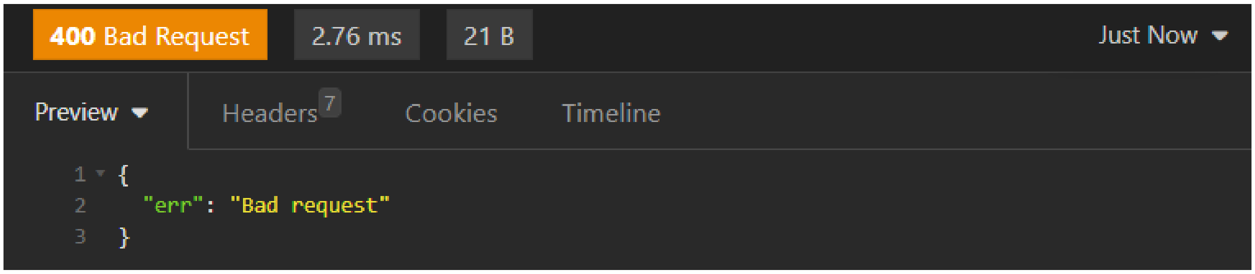
If any of the required property is missing when creating a new product with POST REQUEST:



POST http://localhost:3000/api/products

```
{
  "name": "Kiwi Yellow",
  "price": 17,
  "quantity": 1400,
  "category": "fruit"
}
```

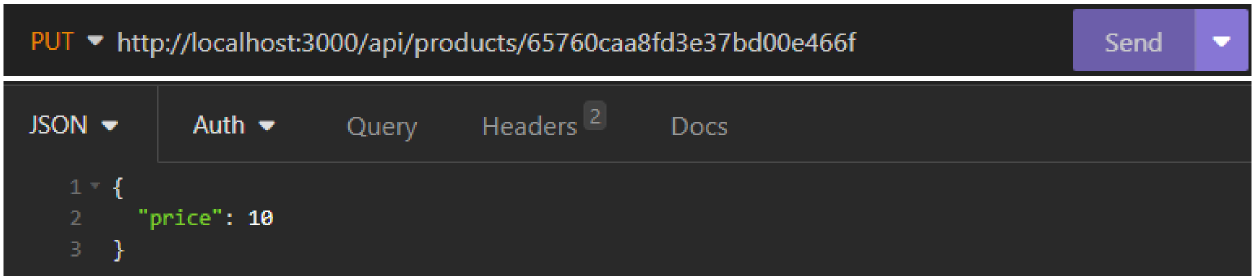
RESPONSE: 400 Bad Request



400 Bad Request 2.76 ms 21 B

```
{
  "err": "Bad request"
}
```

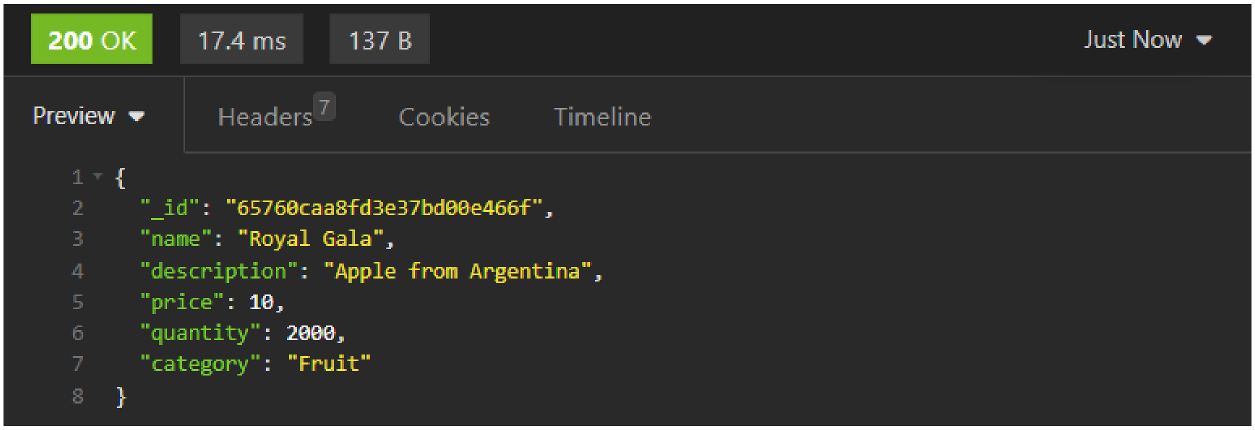
4. To update a specific product by ID, make a PUT REQUEST to the server.
   PUT: /api/products/:id



PUT http://localhost:3000/api/products/65760caa8fd3e37bd00e466f

```
{
  "price": 10
}
```

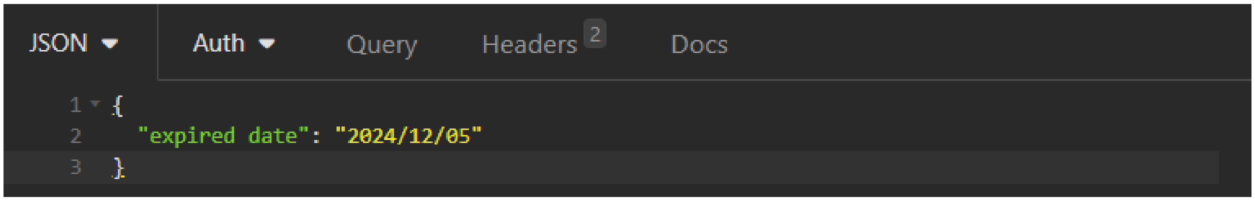
RESPONSE: 200 OK



200 OK 17.4 ms 137 B

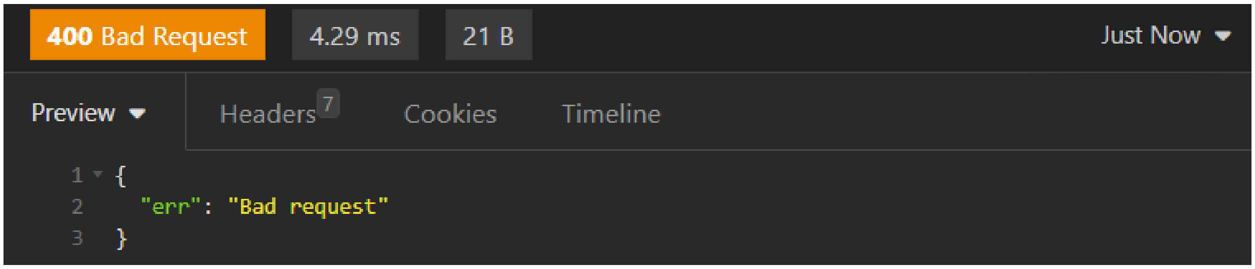
```
{
  "_id": "65760caa8fd3e37bd00e466f",
  "name": "Royal Gala",
  "description": "Apple from Argentina",
  "price": 10,
  "quantity": 2000,
  "category": "Fruit"
}
```

If update non-existed property when updating a specific product with PUT REQUEST:



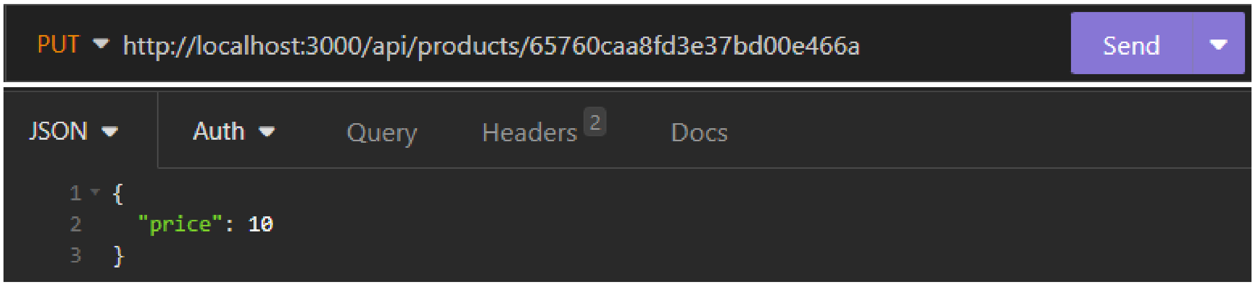
```
{
  "expired date": "2024/12/05"
}
```

RESPONSE: 400 Bad Request



```
{
  "err": "Bad request"
}
```

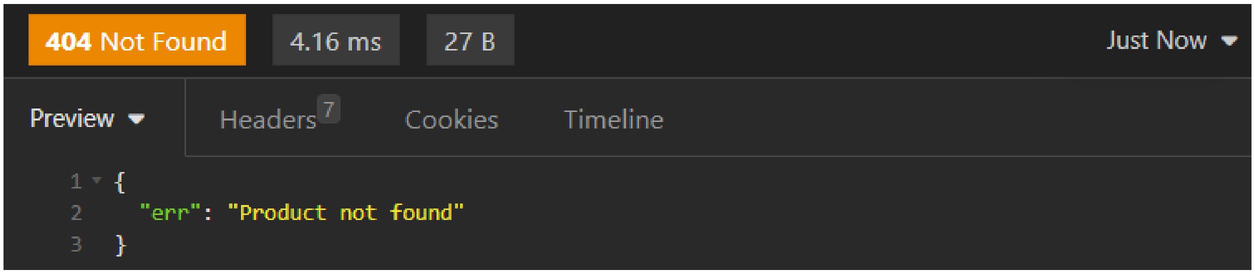
If update property to the non-existed product when updating a specific product with PUT REQUEST:



```
http://localhost:3000/api/products/65760caa8fd3e37bd00e466a
```

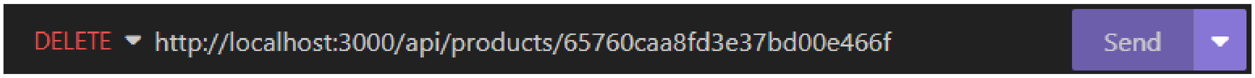
```
{
  "price": 10
}
```

RESPONSE: 404 Not Found



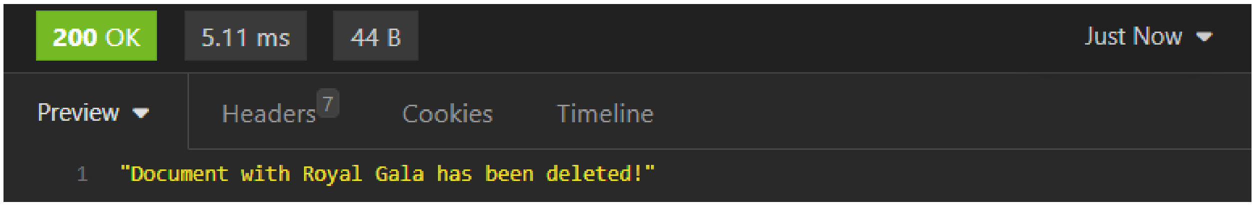
```
{
  "err": "Product not found"
}
```

5. To delete a product by ID, make a DELETE REQUEST to the server.
   DELETE: /api/products/:id



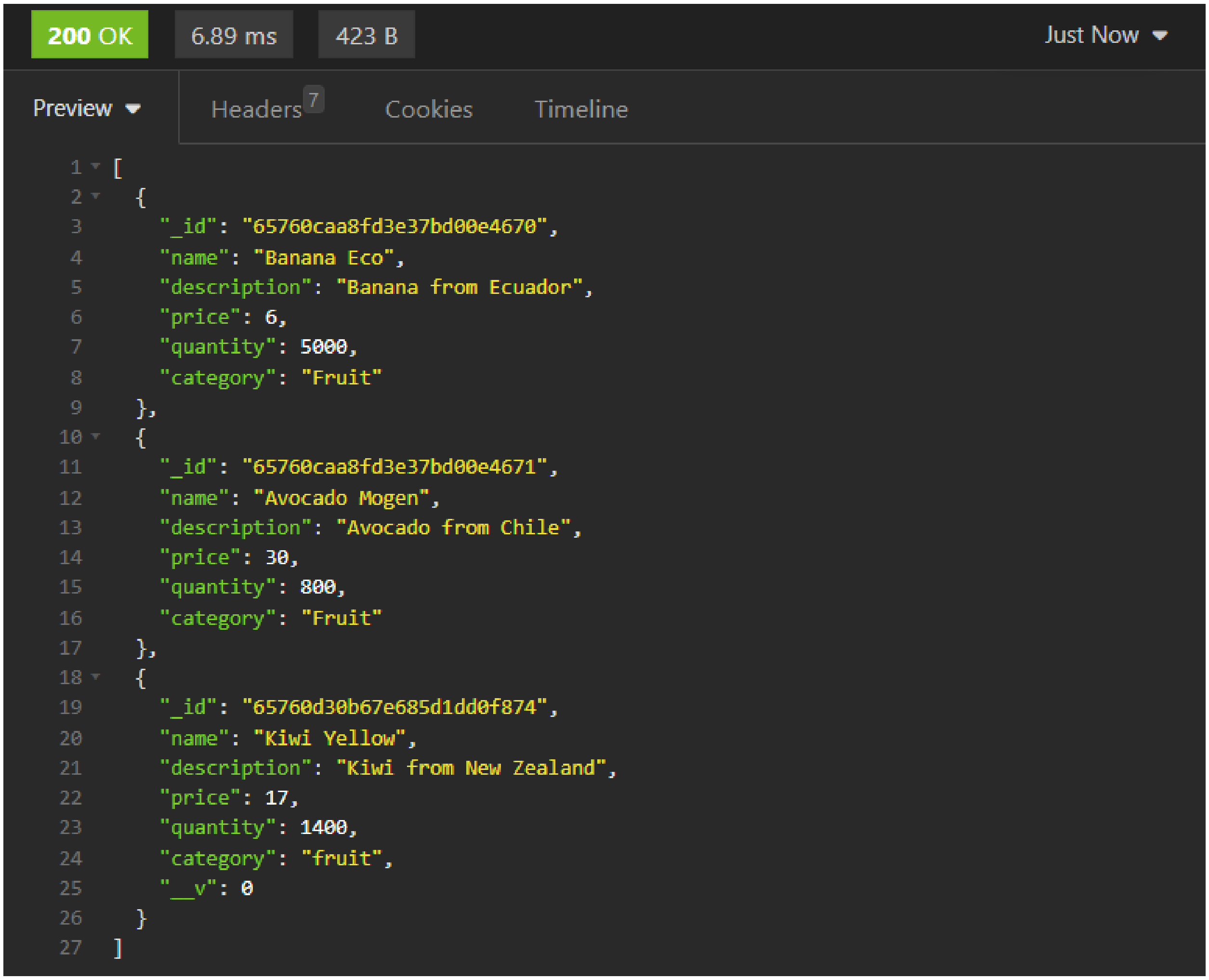
```
http://localhost:3000/api/products/65760caa8fd3e37bd00e466f
```

RESPONSE: 200 OK



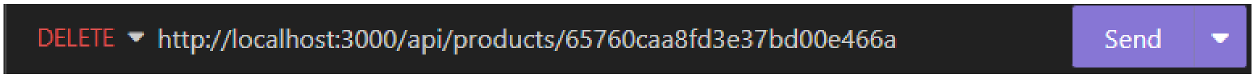
```
"Document with Royal Gala has been deleted!"
```

Doublecheck with GET all products to make sure the product is deleted:



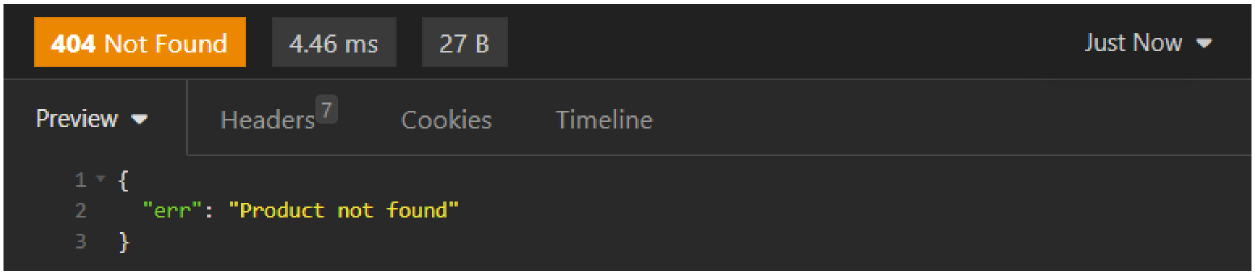
```
[
  {
    "_id": "65760caa8fd3e37bd00e4670",
    "name": "Banana Eco",
    "description": "Banana from Ecuador",
    "price": 6,
    "quantity": 5000,
    "category": "Fruit"
  },
  {
    "_id": "65760caa8fd3e37bd00e4671",
    "name": "Avocado Mogen",
    "description": "Avocado from Chile",
    "price": 30,
    "quantity": 800,
    "category": "Fruit"
  },
  {
    "_id": "65760d30b67e685d1dd0f874",
    "name": "Kiwi Yellow",
    "description": "Kiwi from New Zealand",
    "price": 17,
    "quantity": 1400,
    "category": "fruit",
    "__v": 0
  }
]
```

If the product cannot be found or the ID is incorrect when deleting a specific product by ID:



```
http://localhost:3000/api/products/65760caa8fd3e37bd00e466a
```

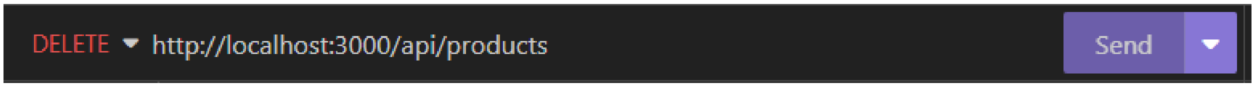
RESPONSE: 404 Not Found



```
{
  "err": "Product not found"
}
```

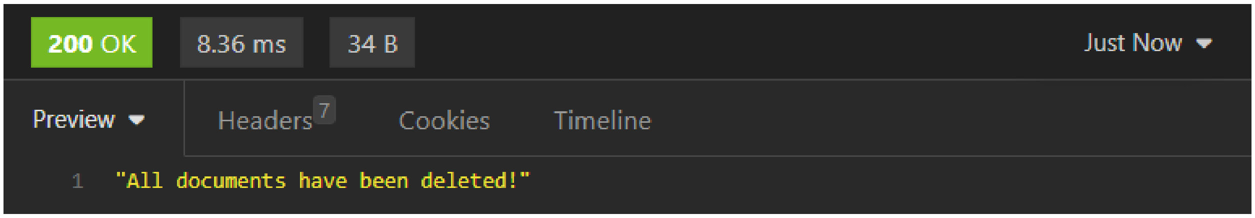
6. To delete all the products, make a DELETE REQUEST to the server.

DELETE: /api/products



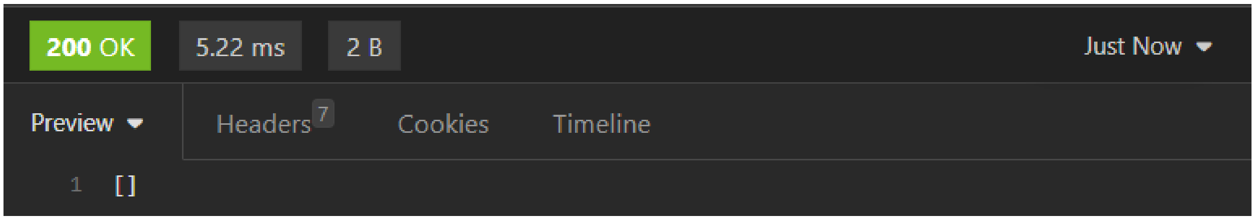
```
http://localhost:3000/api/products
```

RESPONSE: 200 OK



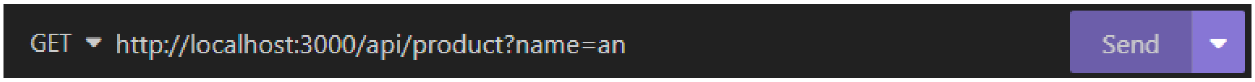
```
"All documents have been deleted!"
```

Doublecheck with GET all products to make sure all the products are deleted:



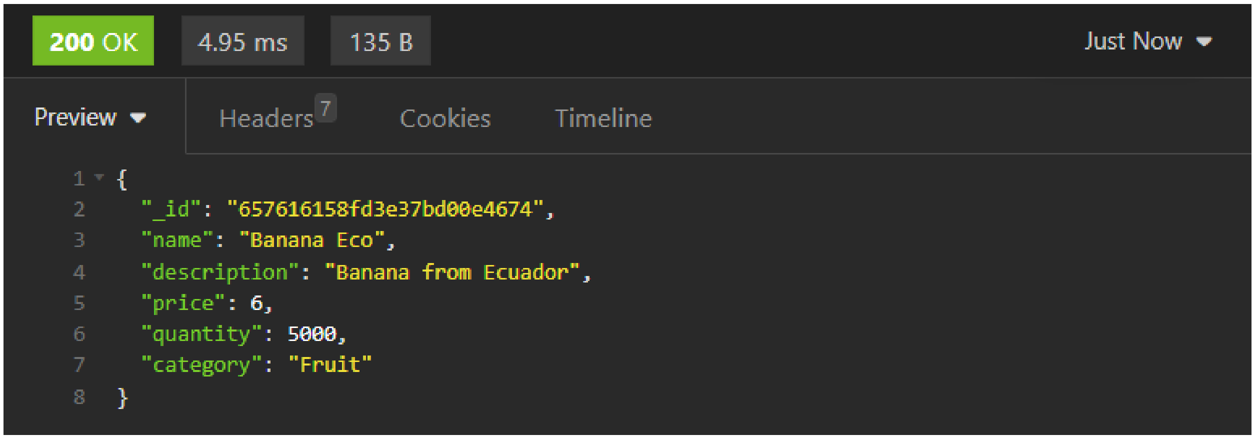
```
[]
```

7. To retrieve a product whose name contains specified keyword, make a GET REQUEST to the server.
   GET: /api/product?name=an



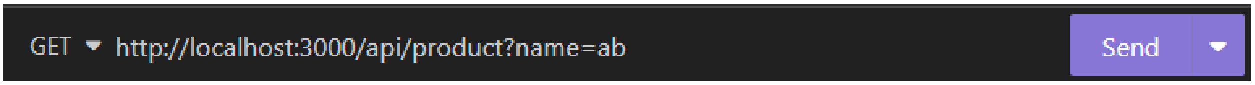
```
http://localhost:3000/api/product?name=an
```

RESPONSE: 200 OK



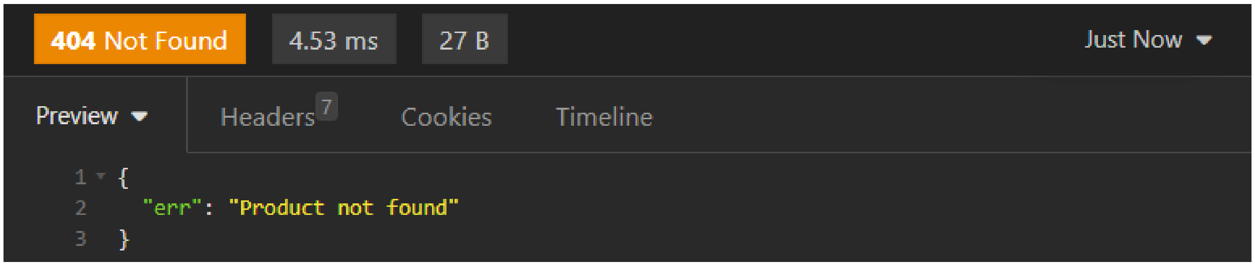
```
{
  "_id": "657616158fd3e37bd00e4674",
  "name": "Banana Eco",
  "description": "Banana from Ecuador",
  "price": 6,
  "quantity": 5000,
  "category": "Fruit"
}
```

If the product cannot be found or the name is incorrect when retrieving a specific product whose name contains specified keyword:



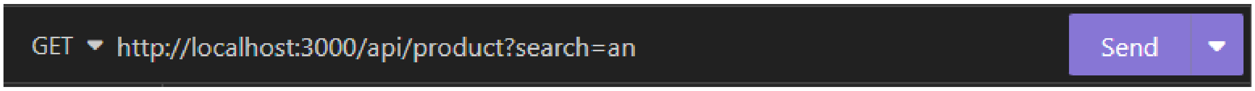
```
http://localhost:3000/api/product?name=ab
```

RESPONSE: 404 Not Found



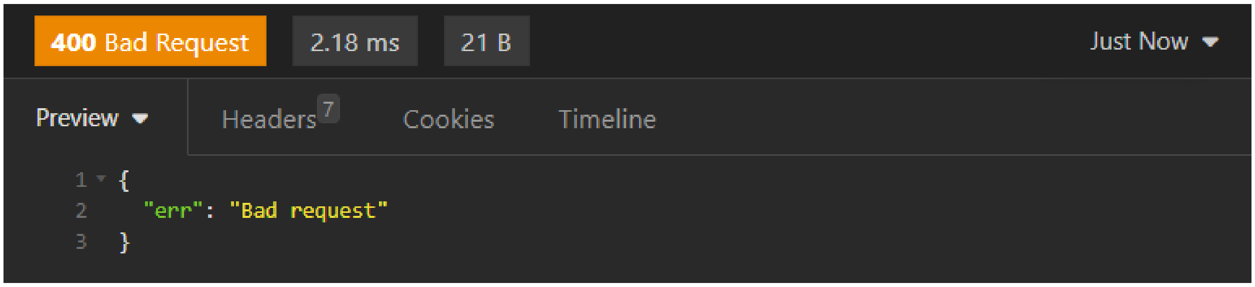
```
{
  "err": "Product not found"
}
```

If the search param is incorrect when retrieving a specific product whose name contains specified keyword:



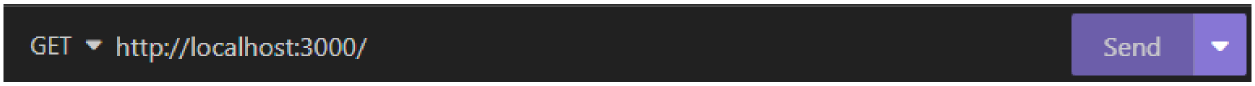
```
http://localhost:3000/api/product?search=an
```

RESPONSE: 400 Bad Request



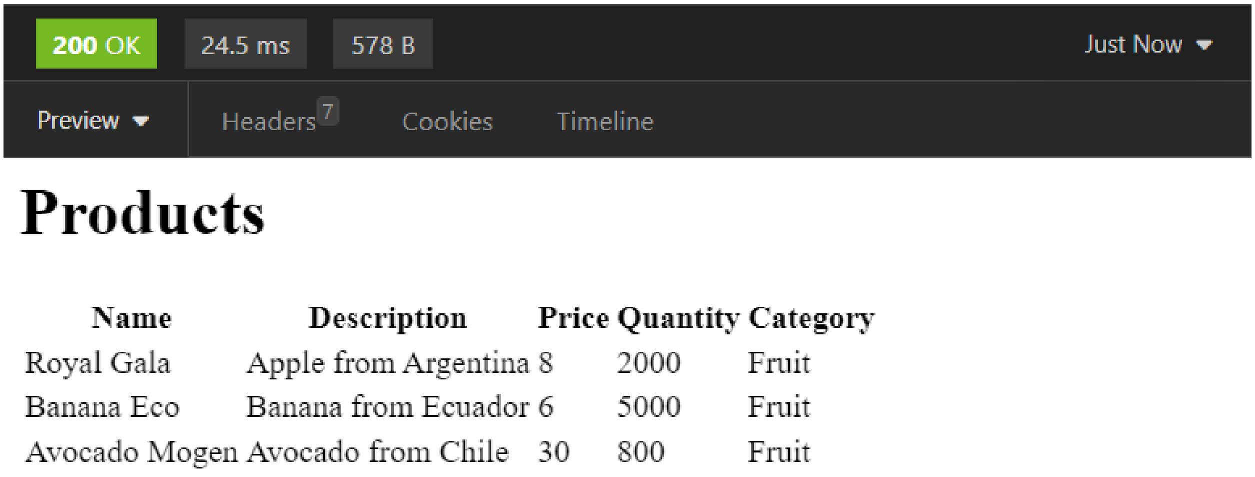
```
{
  "err": "Bad request"
}
```

8. To send back HTML with all products in a table with all information about the product, make a GET REQUEST to the server.
   GET: /



```
http://localhost:3000/
```

RESPONSE: 200 OK with HTML



Products

Name	Description	Price	Quantity	Category
Royal Gala	Apple from Argentina	8	2000	Fruit
Banana Eco	Banana from Ecuador	6	5000	Fruit
Avocado Mogen	Avocado from Chile	30	800	Fruit