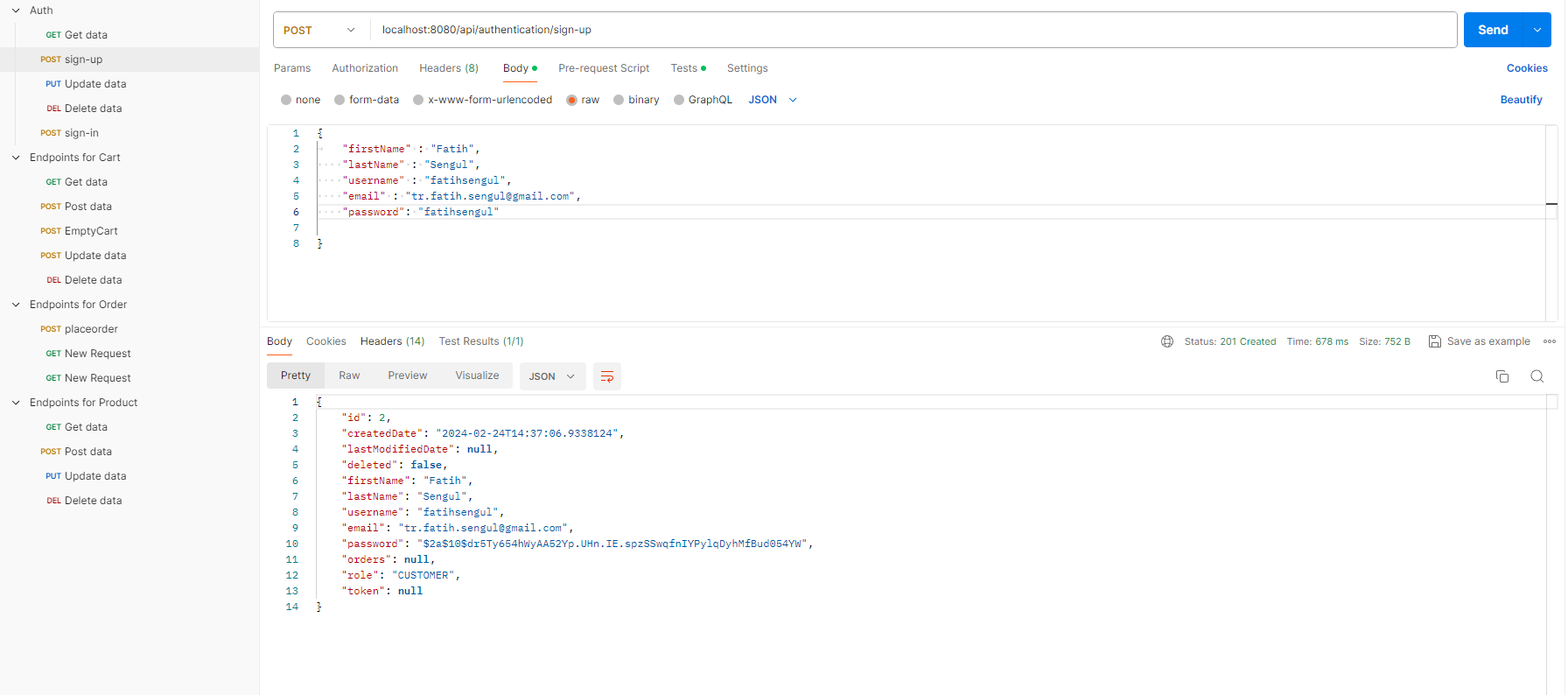
**Projenin genel yapısı şu şekilde oluşturulmuştur:**

* config
* controller
* exception
* mapper
* model
  + common
  + entity
  + enums
  + request
  + response
* repository
* security
  + jwt
* service
* util

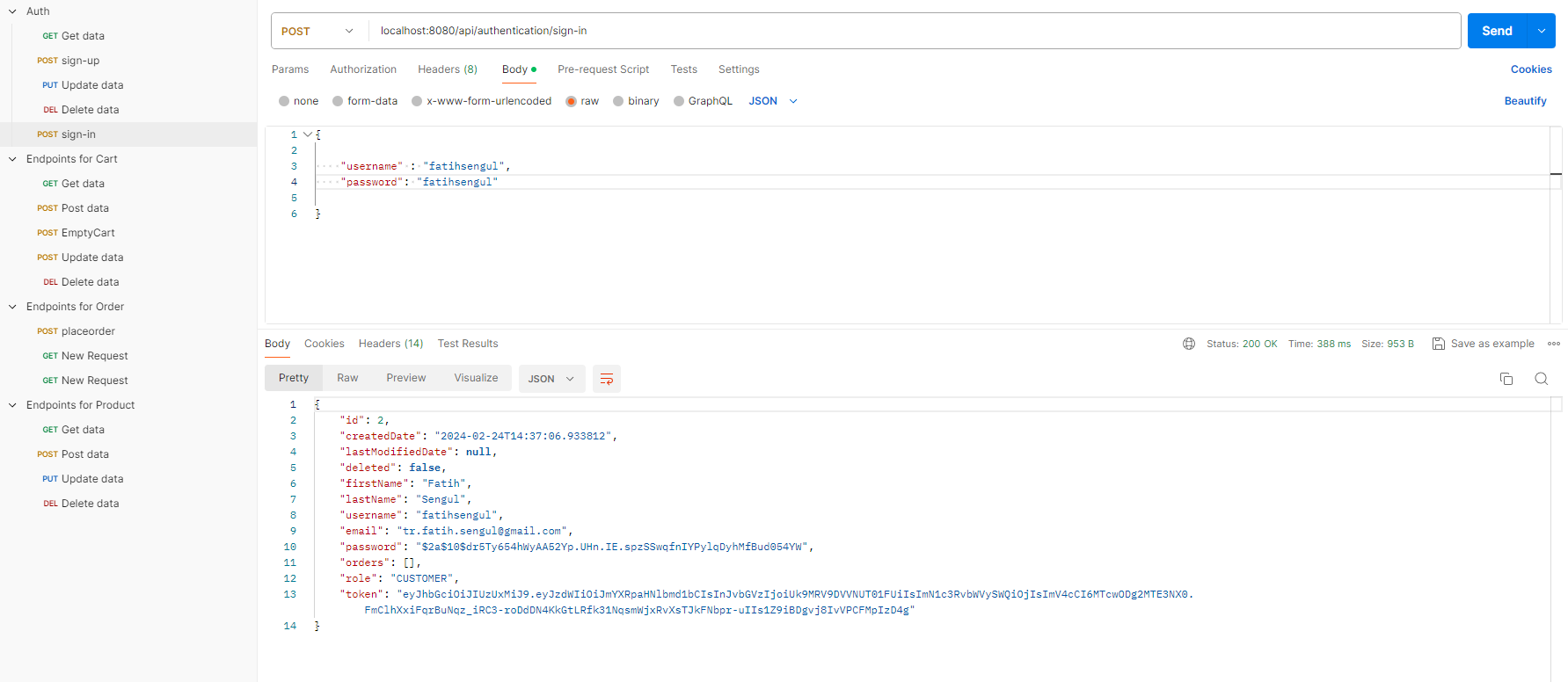
**Kullanılan Teknolojiler**

* Spring Boot framework
* Spring Data JPA
* Spring Security
* PostgreSQL veritabanı
* Lombok
* MapStruct
* JWT

Endpointler ve Servisler:

1. **Sign Up (POST /sign-up)**  
     
   localhost:8080/api/authentication/sign-up  
     
   
2. **Sign In (POST /sign-in):**

localhost:8080/api/authentication/sign-in

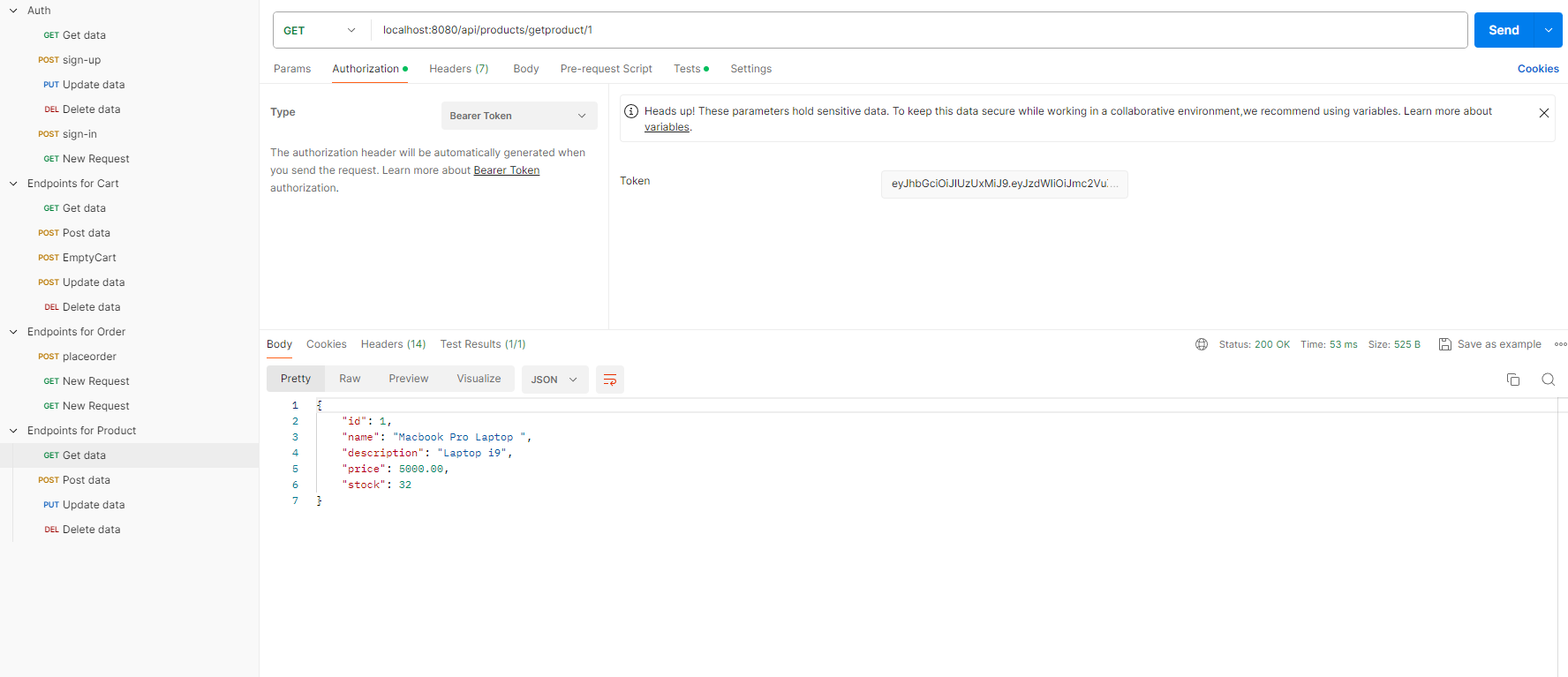


Ayrıca kullanıcıyı Admin yapabilmek için bir servis oluşturulmuştur. Bir ADMIN rolü başka bir kullanıcıyı ADMIN rolü ile yetkilendirebilmesi sağlanmıştır.

localhost:8080/api/internal/make-admin/2

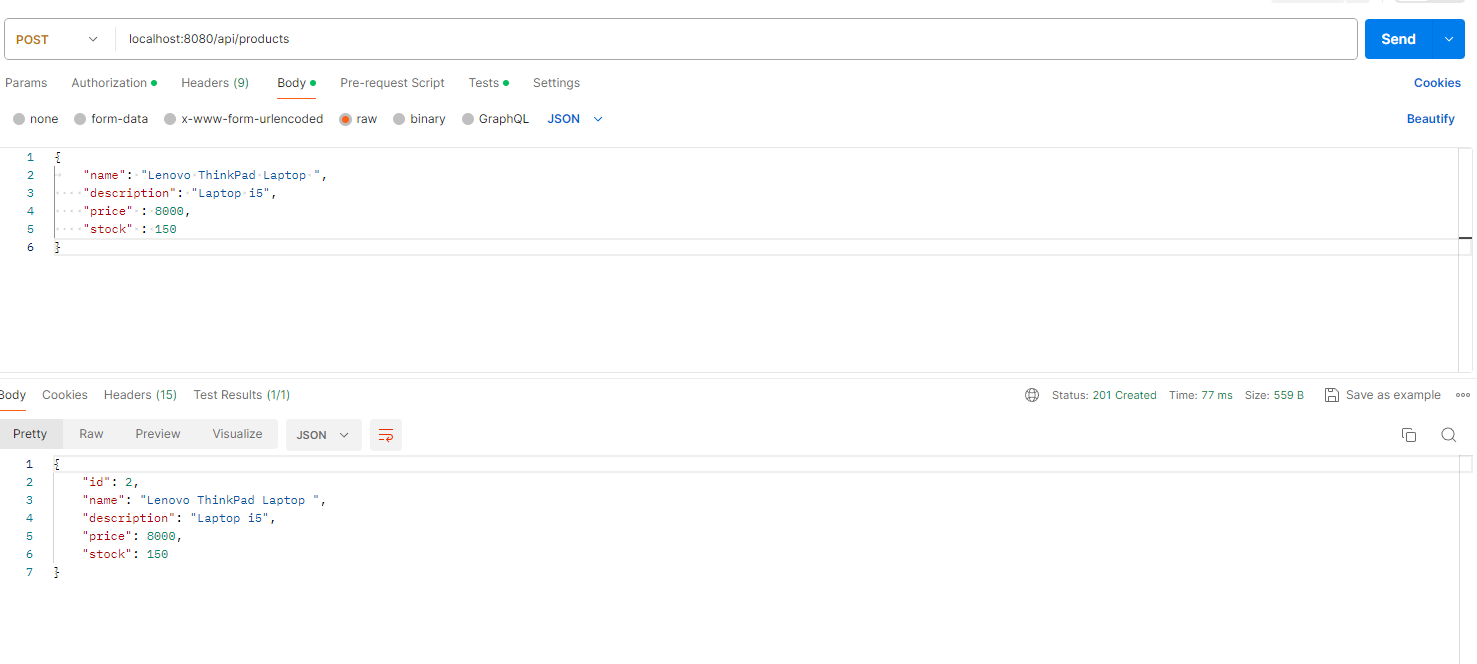
1. **GetProduct (GET /getproduct):**

/api/authentication/getproduct/{productId}



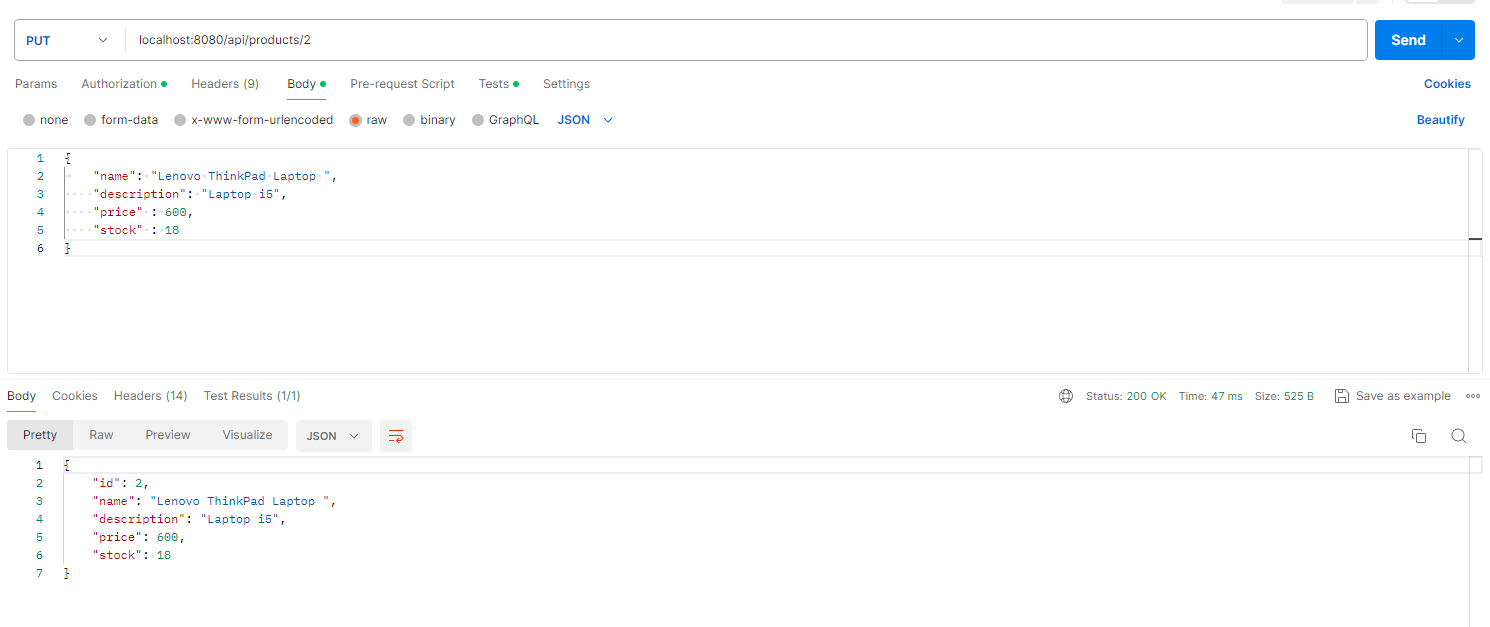
1. **CreateProduct (POST/products)**

localhost:8080/api/products

****

1. **UpdateProduct(PUT/products)**

localhost:8080/api/products

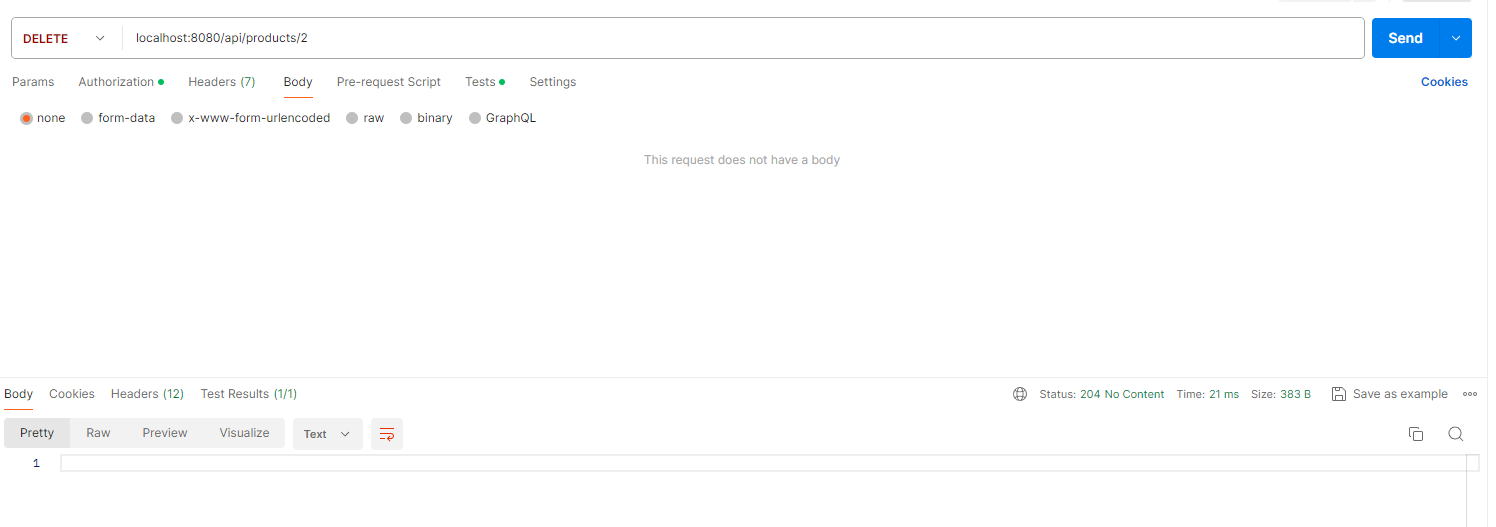
****

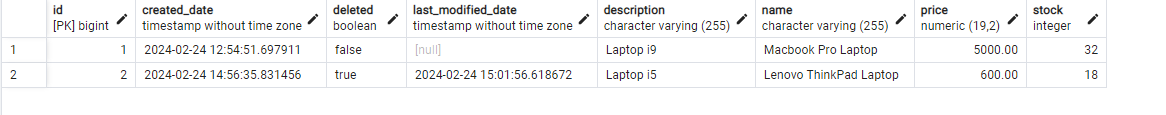
1. **DeleteProduct(DELETE/products)**

localhost:8080/api/products/2

Proje içerisinde Delete işlemi doğrudan db’den kaldırmak yerine deleted column’u true çekilir.

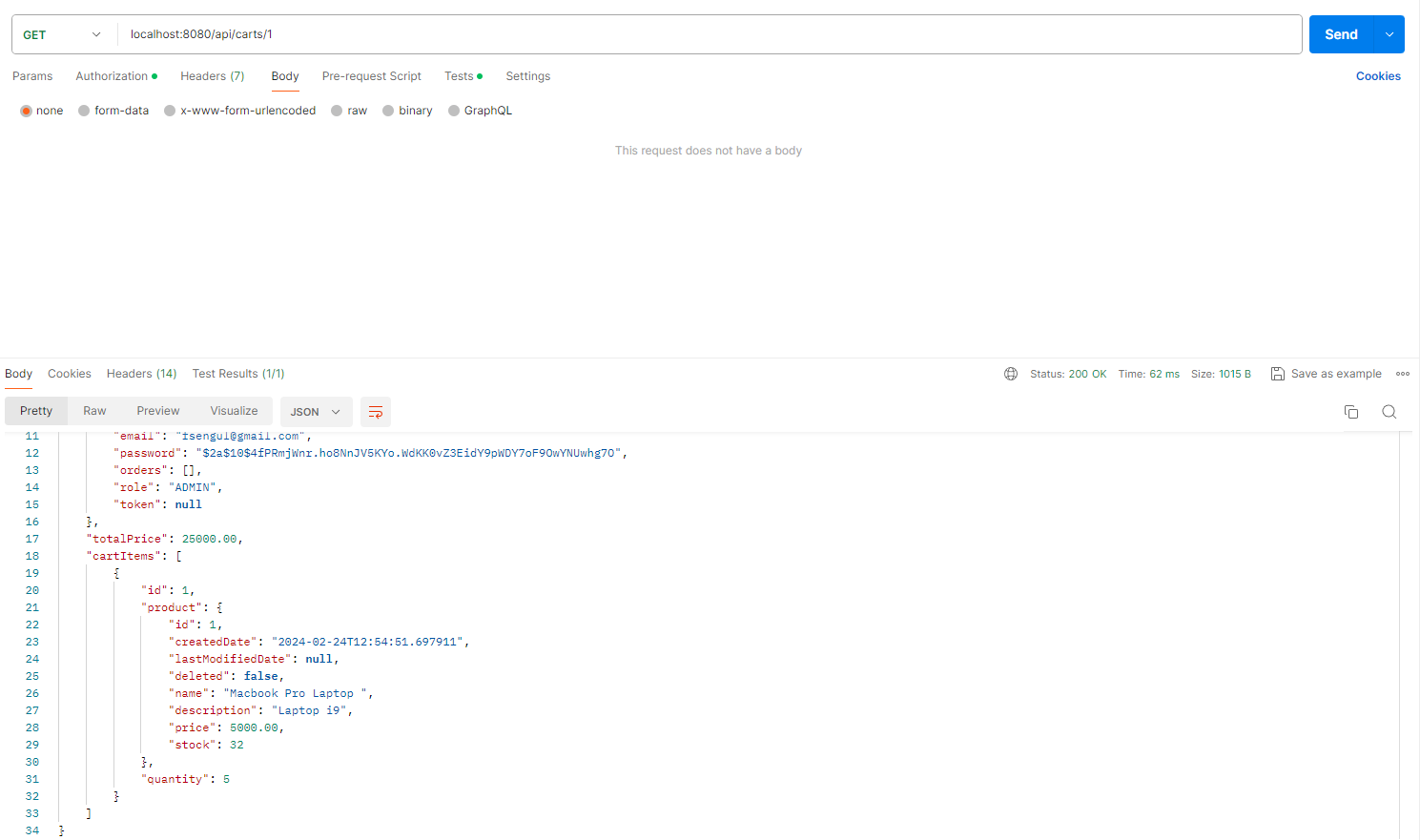
localhost:8080/api/products/2

****

****

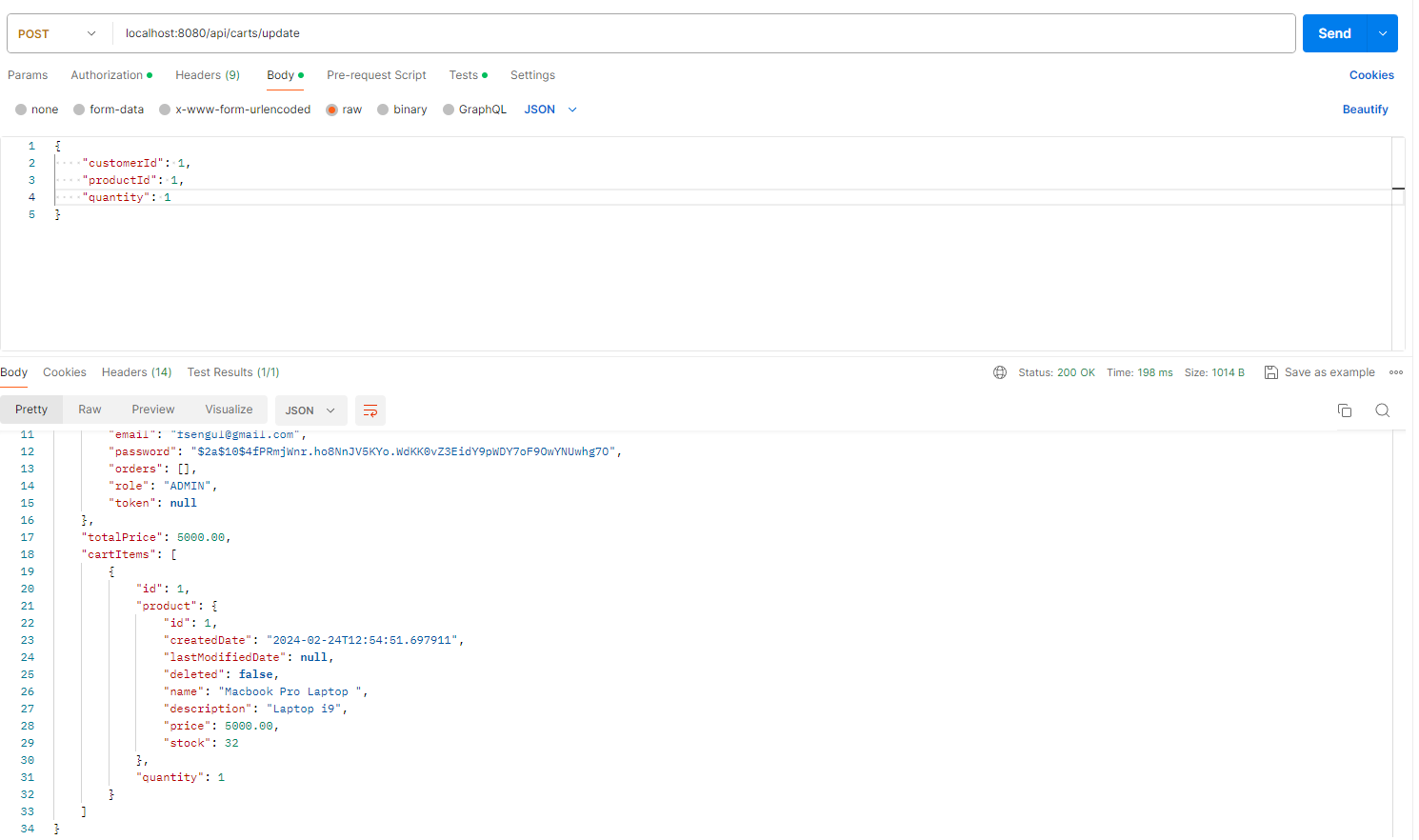
1. **GetCart(GET/{customerId})**

localhost:8080/api/carts/2



1. **UpdateCart(POST/update)**

localhost:8080/api/carts/update

****

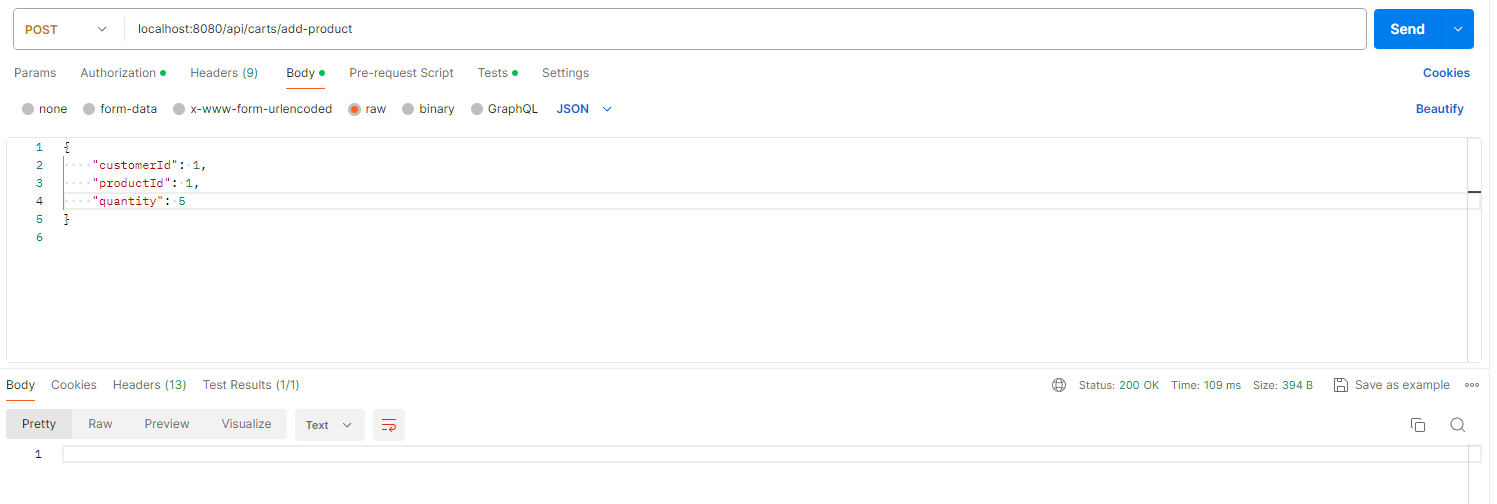
Her cart’a ekleme, çıkarma, miktar arttırma yapıldığında Total Price alanında güncellendiği görülmektedir.

1. **EmptyCart(POST/empty)**

localhost:8080/api/carts/empty

1. **AddProductToCart(POST/add-product)**

localhost:8080/api/carts/add-product

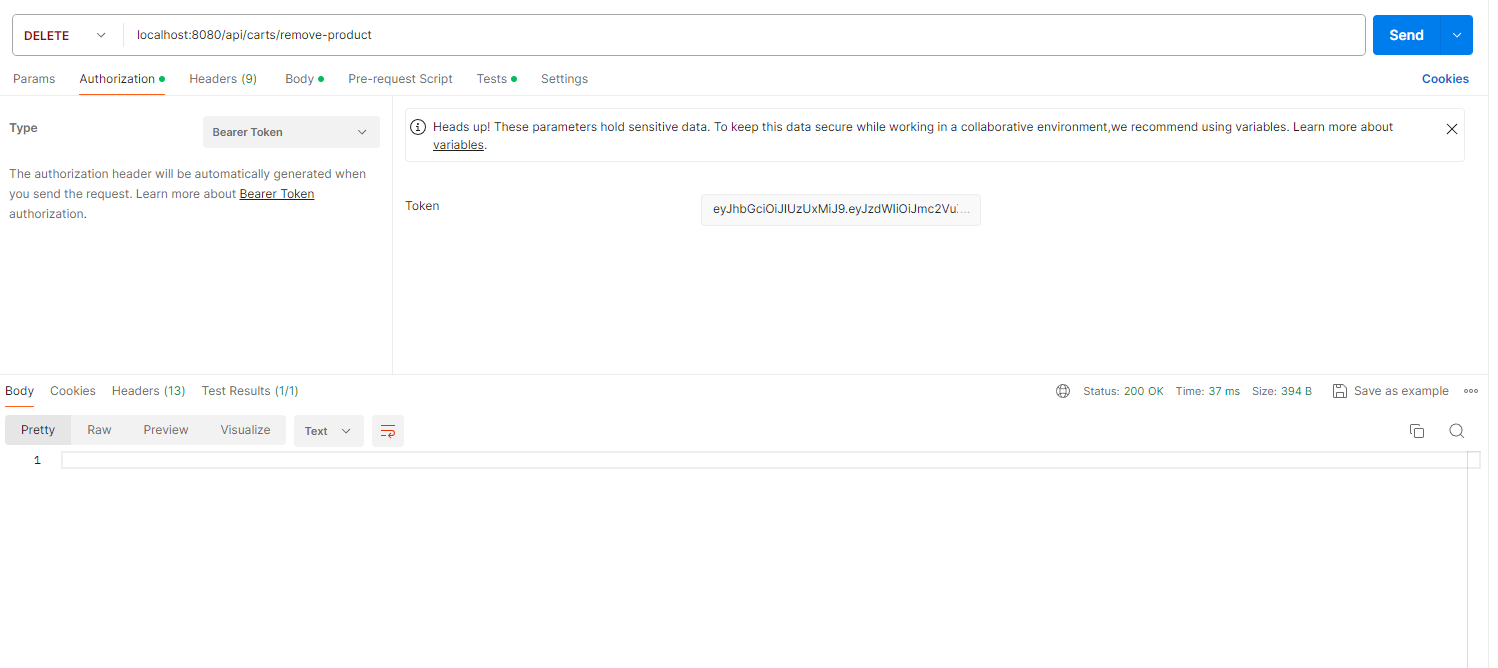


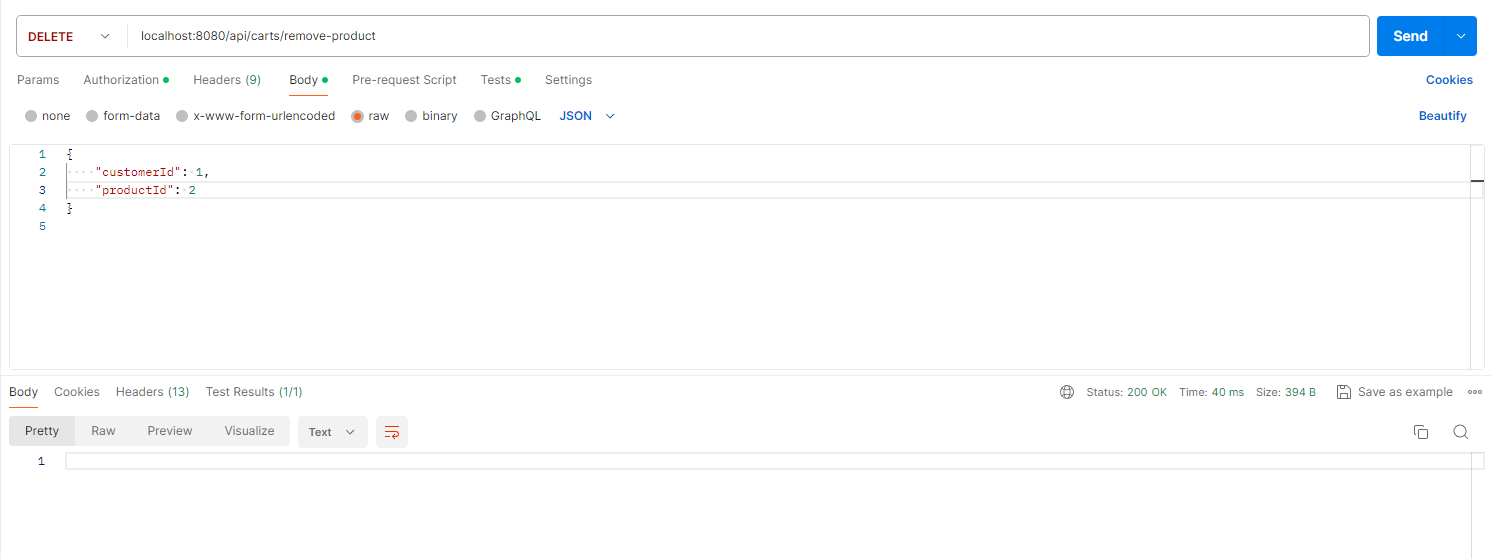
Aynı üründen tekrardan sepete eklenmek istenirse aynı üründe olup olmadığı kontrol edilerek o ürünün eklenen miktar kadar toplam ürün miktarı arttırılmıştır. Product ile CartItem arasında OneToOne ilişkisi kurulmuştur.



1. **RemoveProductToCart(DELETE/remove-product)**

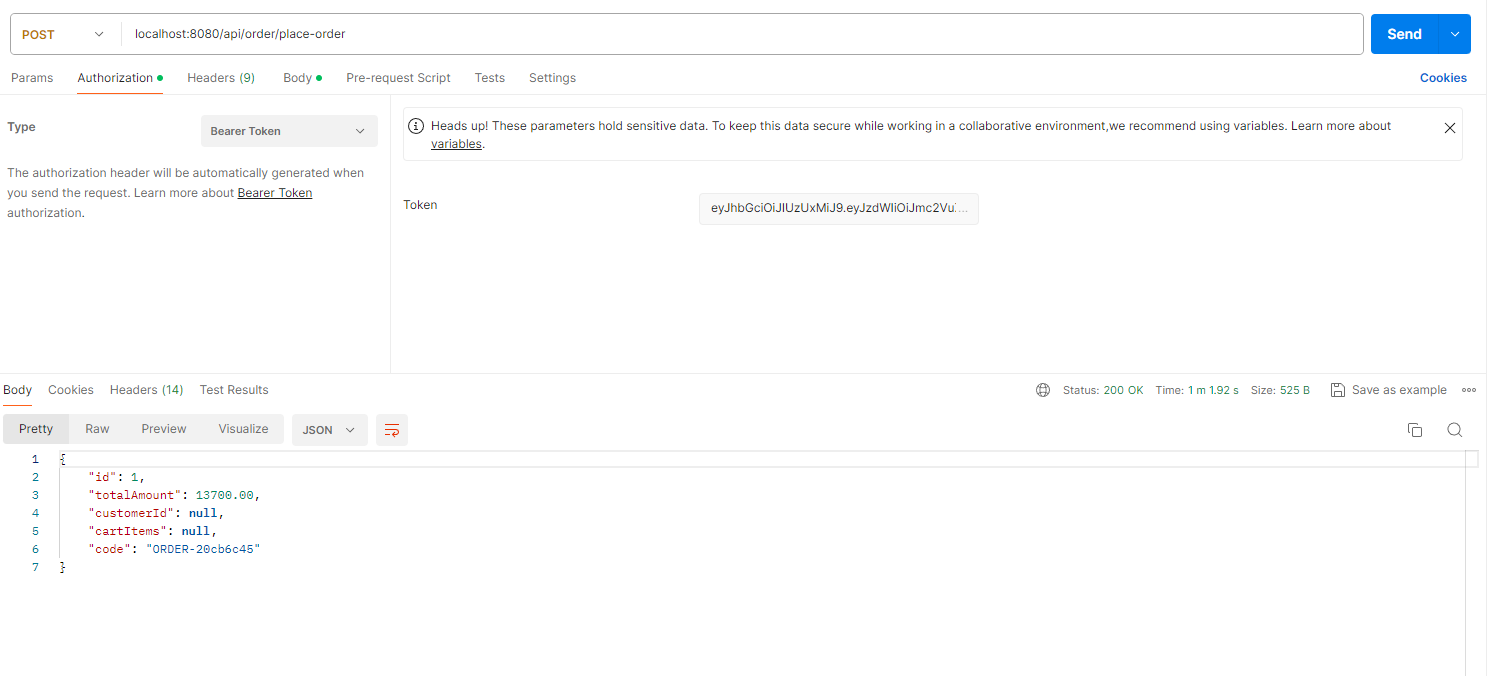
localhost:8080/api/carts/remove-product





1. **PlaceOrder(POST/place-order)**

localhost:8080/api/order/place-order

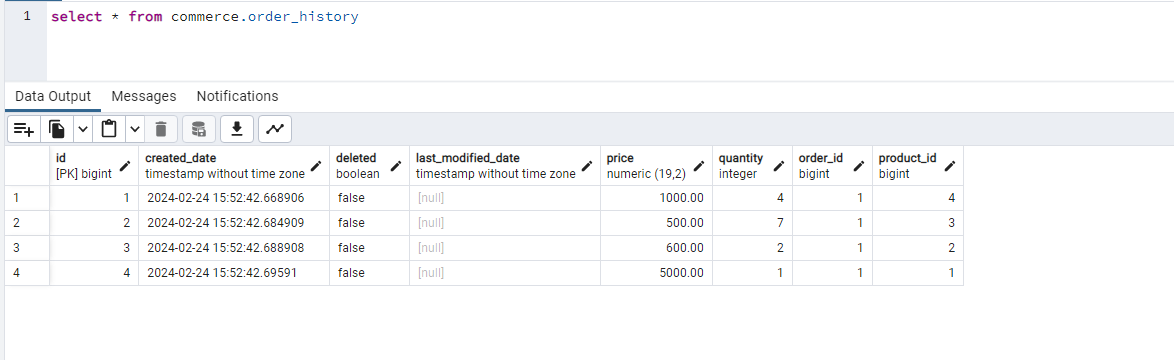


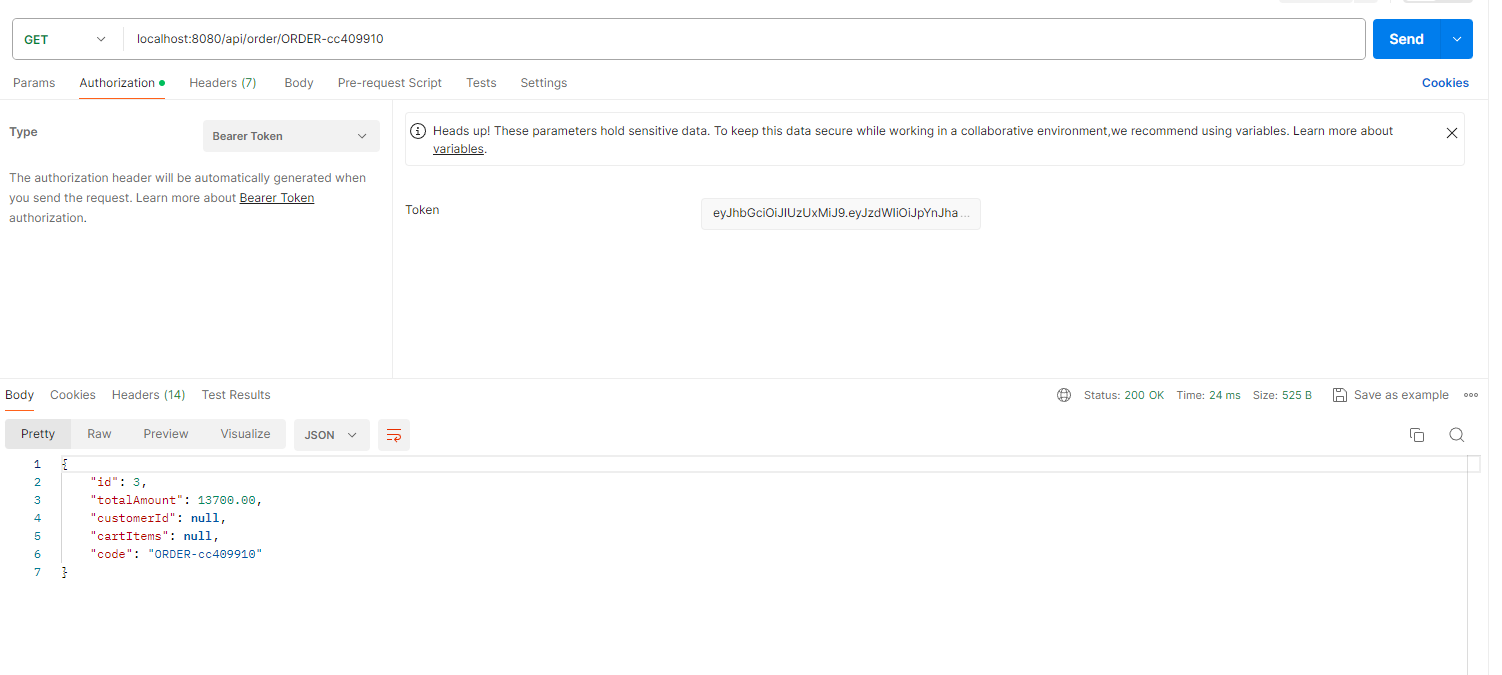
Place Order işlemi tamamlandıktan sonra cart’ın boşatılması sağlanmıştır.

1. **GetOrderForCode(GET/order{ORDER-})**

localhost:8080/api/order/ORDER-f8e60e23

OrderHistory Tablosu oluşturularak geçmişe dönük siparişlerin ve sipraiş sırasındaki fiyatların tutulması sağlanmıştır. Ayrıca bir sipariş code oluşturularak bunların sorgulanması sağlanmıştır. Order Code oluşturma için UUID kütüphanesi kullanılmıştır.





1. **GetAllOrdersForCustomer**

localhost:8080/api/order/customer/3

