

## Proyecto realizado en .net core 5

### Base de datos 2019

GET:

<https://localhost:44317/api/Ticket?page=1>

El parámetro page indica la página a consultar, la paginación está para 5 registros

https://localhost:44317/api/Ticket?page=1

GET https://localhost:44317/api/Ticket?page=1

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Body Cookies Headers (5) Test Results Status: 200 OK Time: 22

Pretty Raw Preview Visualize JSON

```
1 {
2   "pages": 2,
3   "records": [
4     {
5       "id": 1,
6       "usuario": "Andres",
7       "fechaCreacion": "2021-10-22T00:18:47.297",
8       "fechaActualizacion": null,
9       "estatusTicketId": 1,
10      "estatusTicket": "Abierto"
11    },
12  ]
13 }
```

<https://localhost:44317/api/Ticket/1>

Consulta para traer un único registro por medio del Id del ticket

GET https://localhost:44317/api/Ticket/1

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Body Cookies Headers (5) Test Results Status: 200 OK Time: 22

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "usuario": "Andres",
4   "fechaCreacion": "2021-10-22T00:18:47.297",
5   "fechaActualizacion": null,
6   "estatusTicketId": 1,
7   "estatusTicket": "Abierto"
8 }
```

POST:

<https://localhost:44317/api/Ticket>

Inserta registro

https://localhost:44317/api/Ticket

POST https://localhost:44317/api/Ticket

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "usuario": "Roberto",
3   "estatusTicketId": 1
4 }
```

## Ejemplo:

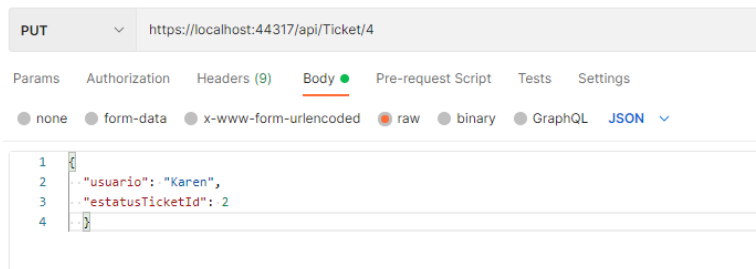
```
{
  "usuario": "Roberto",
  "estatusTicketId": 1
}
```

Fecha creación se actualiza automáticamente.

## PUT:

<https://localhost:44317/api/Ticket/4>

## Actualiza registro



## Ejemplo

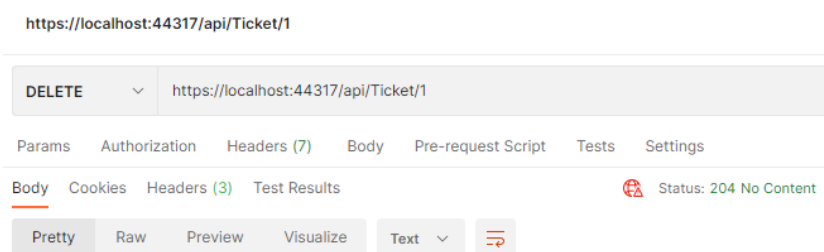
```
{
  "usuario": "Karen",
  "estatusTicketId": 2
}
```

Fecha actualización se calcula automáticamente.

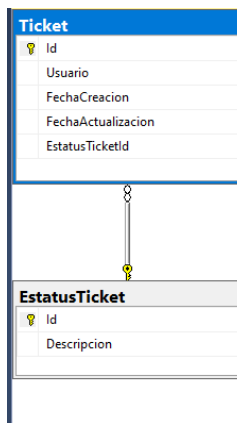
## DELETE:

<https://localhost:44317/api/Ticket/1>

## Elimina registro



Base de datos:



En la carpeta Base de datos se deja un archivo “PruebasTicket.bak” el cual es el backup de la base de datos y un archivo con nombre “script base de datos.sql” para crear la base de datos y tablas por script