

COMANDOS DE LA API REST.

Entorno:

Original: <https://bmsvieira.gitbook.io/osticket-api>

```
docker run -v  
/export/usuarios_ml4ds/sblanco/osticket-api_php8/ost_wbs:/var/www/html/ost_wbs -p  
1234:80 3bb432d70593
```

Se montan los datos del repositorio osticket:
/export/usuarios_ml4ds/sblanco/osticket-api_php8

Datos de configuración:

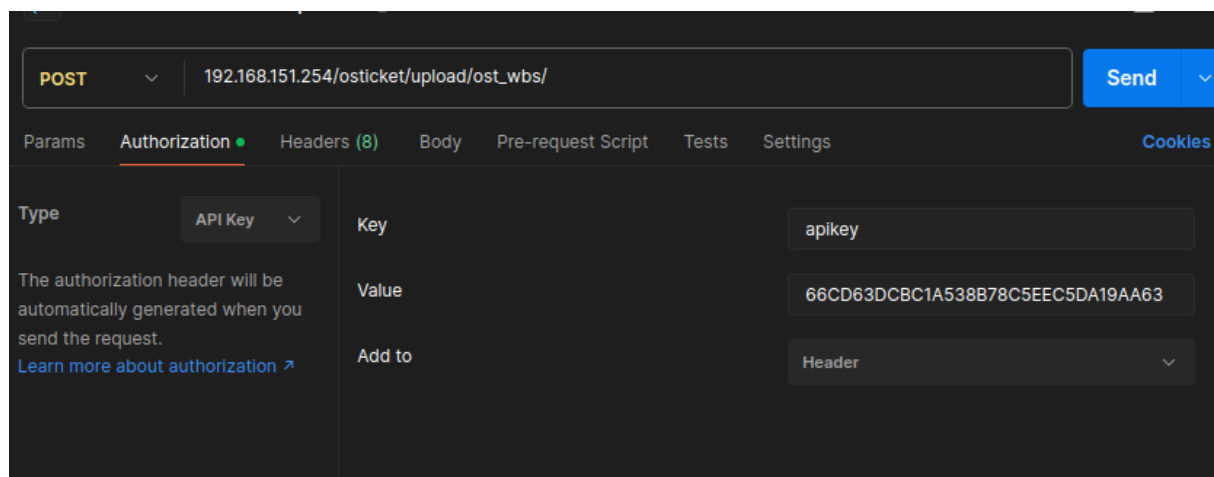
Key: [66CD63DCBC1A538B78C5EEC5DA19AA63](#)

IP: 192.168.151.254/osticket/upload/ost_wbs/

Agente: burns

Usuario: lisa, id 2
123456

Para las pruebas usaré Postman, por lo que debemos crear en Authorization la API key.



Crear incidencia:

Tipo: **POST**

Datos en el cuerpo:

```
Unset
{
  "query": "ticket",
  "condition": "add",
  "parameters": {
    "title": "Problema con la contraseña",
    "subject": "No logro entrar en el sistema",
    "user_id": 2,
    "priority_id": 2,
    "status_id": 1,
    "dept_id": 1,
    "sla_id": 1,
    "topic_id": 1
  }
}
```

La prioridad puede ser:

priority_id	priority	priority_desc	priority_color	priority_urgency	ispublic
1	low	Low	#DDFFDD	4	1
2	normal	Normal	#FFFFFF0	3	1
3	high	High	#FEE7E7	2	1
4	emergency	Emergency	#FEE7E7	1	1

Y el estado puede ser:

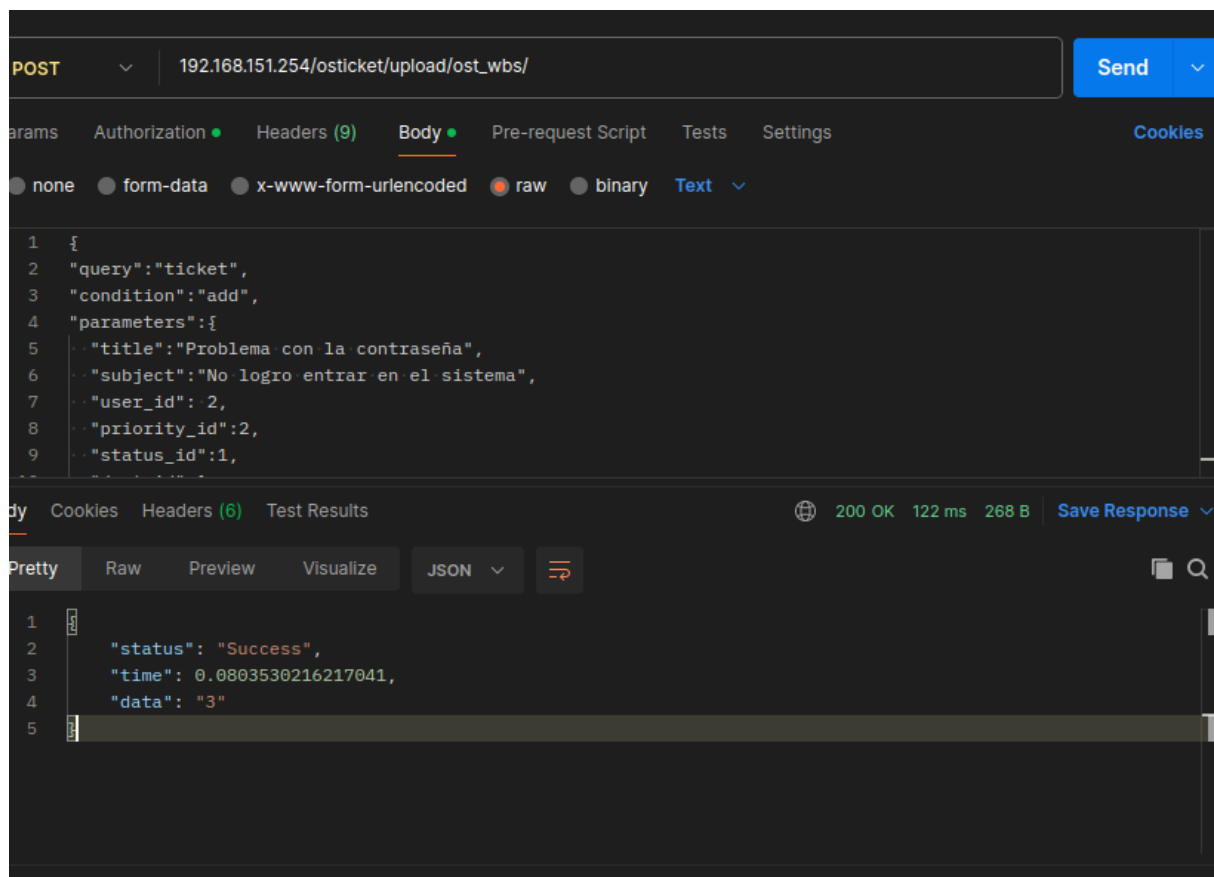
id	name	properties
1	Open	{ "description": "Open tickets." }
2	Resolved	{ "allowreopen": true, "reopenstatus": 0, "description": "Resolved tickets" }
3	Closed	{ "allowreopen": true, "reopenstatus": 0, "description": "Closed tickets. Tickets will still be accessible on client and staff panels." }
4	Archived	{ "description": "Tickets only administratively available but no longer accessible on ticket queues and client panel." }
5	Deleted	{ "description": "Tickets queued for deletion. Not accessible on ticket queues." }

```
{"query": "ticket", "condition": "add", "parameters": { "title": "Problema con la contraseña", "subject": "No logro entrar en el sistema", "user_id": 2, "priority_id": 2, "status_id": 1, "dept_id": 1, "sla_id": 1, "topic_id": 1 }}
```

Devolución:

```
Unset
{
  "status": "Success",
  "time": 0.09253191947937012,
  "data": "3425" // Ticket ID
}
```

Ejemplo en POSTMAN:



Comprobar el estado de una incidencia:

Tipo: **GET**

```
Unset
{
  "query": "ticket",
  "condition": "specific",
  "parameters": {
    "id": 3
  }
}
```

The screenshot displays a REST client interface with a GET request to the URL `192.168.151.254/osticket/upload/ost_wbs/?=`. The request body is a JSON object: `{ "query": "ticket", "condition": "specific", "parameters": { "id": 3 } }`. The response is a 200 OK status with a response time of 38 ms and a body size of 897 B. The response body is shown in a pretty-printed JSON format: `{ "status": "Success", "time": 0.0012240409851074219, "data": { "total": 1, "tickets": [{ "ticket_id": "3", "ticket_pid": null, ... }] } }`.

```
GET 192.168.151.254/osticket/upload/ost_wbs/?= Send
```

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

none form-data x-www-form-urlencoded raw binary Text

```
1 {
2   "query": "ticket",
3   "condition": "specific",
4   "parameters": {
5     "id": 3
6   }
7 }
8
```

Body Cookies Headers (6) Test Results 200 OK 38 ms 897 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "Success",
3   "time": 0.0012240409851074219,
4   "data": {
5     "total": 1,
6     "tickets": [
7       {
8         "ticket_id": "3",
9         "ticket_pid": null,
```

```

{
  "ticket_id": "3",
  "ticket_pid": null,
  "number": "API3",
  "user_id": "2",
  "user_email_id": "0",
  "status_id": "1",
  "dept_id": "1",
  "sla_id": "1",
  "topic_id": "1",
  "staff_id": "0",
  "team_id": "0",
  "email_id": "0",
  "lock_id": "0",
  "flags": "65",
  "sort": "0",
  "subject": "No logro entrar en el sistema",
  "title": "Problema con la contraseña",
  "body": "<p>No logro entrar en el sistema</p>",
  "ip_address": "0",
  "source": "API",
  "source_extra": null,
  "isoverdue": "0",
  "isanswered": "0",
  "duedate": null,
  "est_duedate": null,
  "reopened": null,
  "closed": null,
  "lastupdate": "2024-05-21 10:39:58",
  "created": "2024-05-21 10:39:58",
  "updated": "2024-05-21 10:39:58"
}

```

Si queremos usar **GET** en vez de **POST**:

● none ● form-data ● x-www-form-urlencoded ● raw ● binary Text ▾

```

1  {
2  "query": "ticket",
3  "condition": "specificP",
4  "parameters": {
5    "id": 3
6  }
7  }
8

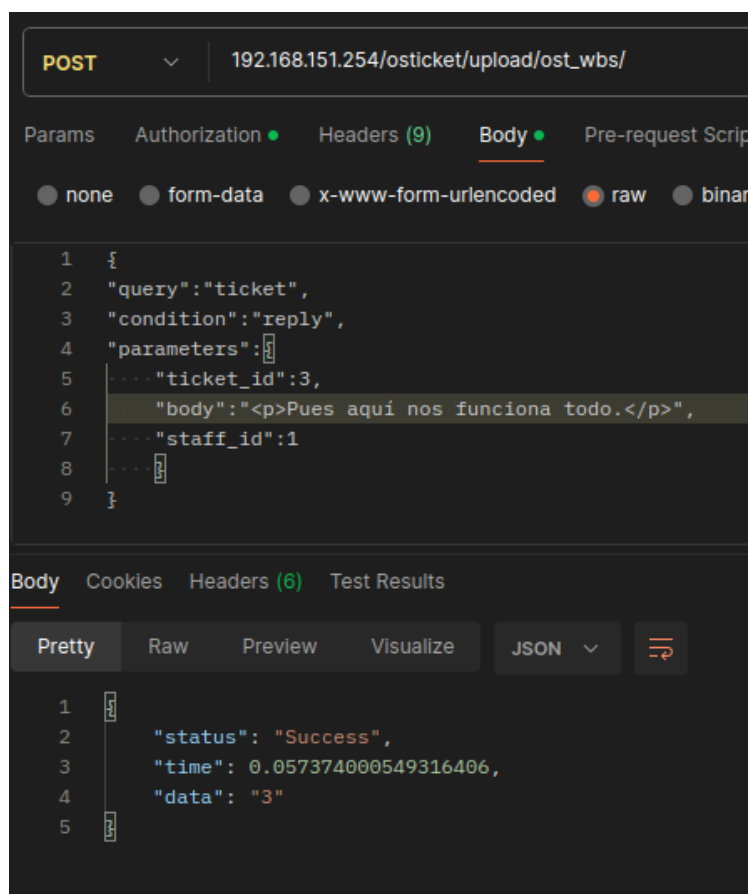
```

Contestar a una incidencia:

Tipo: **POST**

Unset

```
{
  "query": "ticket",
  "condition": "reply",
  "parameters": {
    "ticket_id": 3,
    "body": "<p>Pues aquí nos funciona todo.</p>",
    "staff_id": 1
  }
}
```



Si se contesta una incidencia cerrada, entonces se reabre en el panel.


Cerrar una incidencia:

Tipo: **POST**

Unset

```
{
  "query": "ticket",
  "condition": "close",
  "parameters": {
    "ticket_id": 3,
    "body": "<p>Ticket closed. Thank You!</p>",
    "staff_id": 1,
    "status_id": 3,
    "team_id": 1,
    "dept_id": 2,
    "topic_id": 1,
    "username": "Mollete Cósmico"
  }
}
```

En el panel el usuario lo vería reflejado en:



lisa posted 5/21/24 12:39 PM Problema con la contraseña


No logro entrar en el sistema

Saul Blanco posted 5/21/24 12:50 PM

Pues aquí nos funciona todo.

Saul Blanco posted 5/21/24 12:55 PM

Ticket closed. Thank You!

 Closed by **Mollete Cósmico** 5/21/24 12:55 PM

Listar últimas actualizaciones en el sistema:

Tipo: **POST**

Unset

```
{ "query": "ticket",  
  "condition": "lastChangesP", "parameters": {"minutes": 1200} }
```

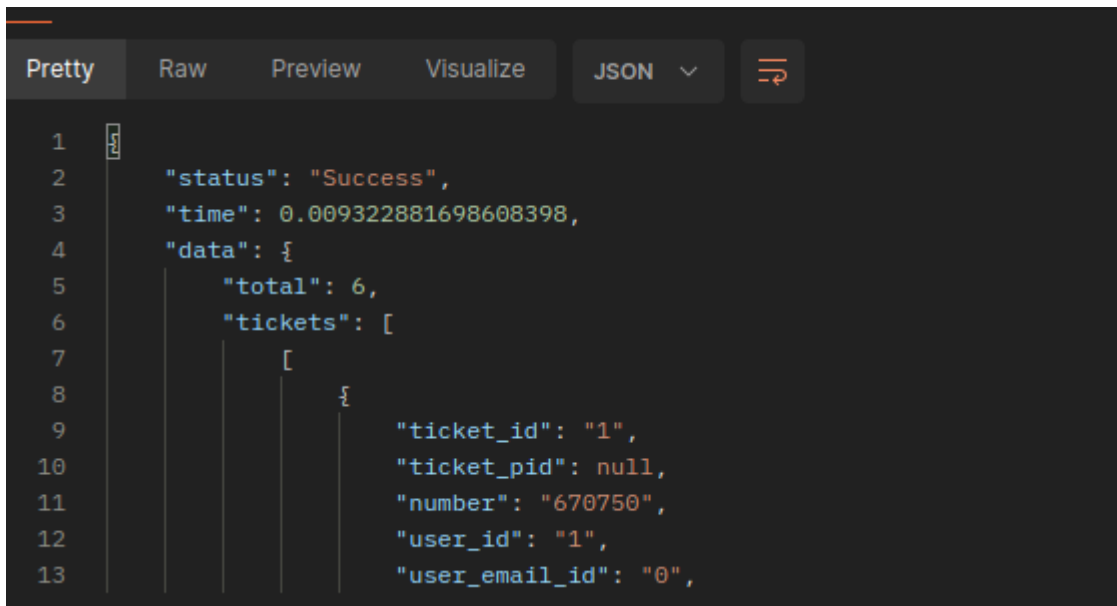
```
{  
  "status": "Success",  
  "time": 0.0012221336364746094,  
  "data": {  
    "total": 3,  
    "tickets": [  
      {  
        "ticket_id": "1",  
        "updated": "2024-05-21 09:38:57",  
        "status_id": "1",  
        "name": "Open"  
      },  
      {  
        "ticket_id": "2",  
        "updated": "2024-05-21 09:57:43",  
        "status_id": "1",  
        "name": "Open"  
      },  
      {  
        "ticket_id": "3",  
        "updated": "2024-05-21 10:55:36",  
        "status_id": "3",  
        "name": "Closed"  
      }  
    ]  
  }  
}
```

Listar tickets entre dos fechas por estado:

Tipo: **POST**

Unset

```
{
  "query": "ticket",
  "condition": "allP",
  "sort": "statusByDate",
  "parameters": {
    "status": 1,
    "start_date": "1990-01-01 00:00:00",
    "end_date": "2025-06-19 23:59:59"
  }
}
```



The screenshot shows a JSON viewer interface with tabs for 'Pretty', 'Raw', 'Preview', and 'Visualize'. The 'Pretty' tab is selected, and the JSON is displayed in a dark-themed editor. The response indicates a successful status and contains a list of tickets.

```
1  {
2    "status": "Success",
3    "time": 0.009322881698608398,
4    "data": {
5      "total": 6,
6      "tickets": [
7        [
8          {
9            "ticket_id": "1",
10           "ticket_pid": null,
11           "number": "670750",
12           "user_id": "1",
13           "user_email_id": "0",
```