

## Contenido

Dirigido A:.....	2
Requisitos:.....	2
Objetivo:.....	2
Importar colección de Postman: .....	3
Uso de la API: .....	4
Obtener token para la autenticación:.....	5
Obtener Histórico de mantenimientos: .....	7
Configuración de la petición: .....	8
Autenticación: .....	8
Ingresar Parametros:.....	9
Calibración de odometro: .....	10
Configuración de la petición: .....	11
Autenticación: .....	11
odometerData:.....	12
Limites: .....	15

## Documentación de la API de Integración

### Mantenimiento

#### Dirigido A:

Administradores del departamento de sistemas y desarrolladores de software.

#### Requisitos:

1. Usuario con acceso al servicio de mantenimiento.
2. *Client\_id* y *secret\_id*, los cuales son suministrados por Satrack para el consumo de la API de Integración.

Para realizar el desarrollo de esta documentación se usará el software Postman, a través del cual se estará probando la API y se generará la configuración.

#### Objetivo:

Explicar cómo integrarse con la API desarrollada por Satrack, para obtener el histórico de mantenimientos y poder calibrar su odómetro, a través del cual se podrá conectar el sistema de información propio con el aplicativo señalado.

Una vez finalizada la documentación, el usuario estará en capacidad de consultar su historial de mantenimientos y poder hacer la calibración de sus odómetros.

---

**Colombia:** Línea nacional **01 8000 51 33 34** • Teléfono **(4) 604 54 54** • ventas@satrack.com

**Ecuador:** Línea nacional **1 800 95 77 00** • Teléfono **(2) 395 77 00** • ventasecuador@satrack.com

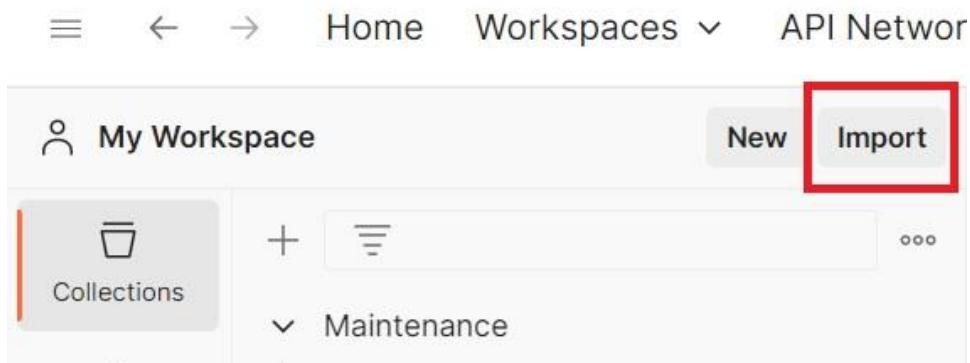
**Panamá:** Teléfono **270 31 33** • ventas@fastrack.com

**satrack.com**

Este manual está enfocado en familiarizarlo con el uso de la API de integración, la forma y herramientas usadas son para una interacción manual y fácil uso. Sin embargo, esta misma API en la práctica se usará de forma automática a través del mecanismo y estrategias de integración que su organización determine.

### Importar colección de Postman:

Darle clic al botón importar colección de Postman.



Emergerá una ventana de importación, darle clic en el botón Choose Files..

## Import

File   Folder   Link   Raw text   Code repository [New](#)   API Gateway [New](#)



Choose Files

Seleccionar el archivo recibido por Satrack “**Maintenance-Integration.postman\_collection.js**”

[Maintenance-Integration.postman\\_collection](#)

## Uso de la API:

Una vez importado aparecerá las 3 solicitudes HTTP en la colección importada.



# satrack

≡ ← → Home Workspaces ▾ API Network

My Workspace

New Import

Collections

+

☰

...

Maintenance

POST GetToken

GET GetHistoricalMaintenance

POST CalibrateOdometer

## Obtener token para la autenticación:

Aquí le damos clic al primero: “GetToken” en la parte derecha nos cargará la petición, en ella seleccionamos la pestaña Body, luego en client\_id y client\_secret ingresamos los valores que le suministra Satrack para el consumo de la API.

Maintenance / GetToken

Save ⌂ ⌄ ⌅ ⌆

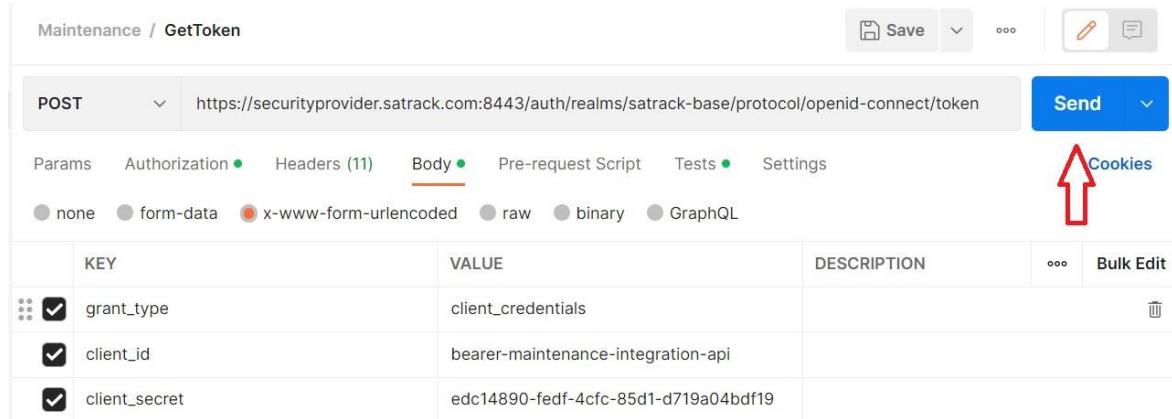
POST https://securityprovider.sattrack.com:8443/auth/realms/sattrack-base/protocol/openid-connect/token **Send**

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings Cookies

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

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	grant_type	client_credentials		
<input checked="" type="checkbox"/>	client_id	bearer-maintenance-integration-api		
<input checked="" type="checkbox"/>	client_secret	edc14890-fedf-4fcf-85d1-d719a04bd19		

Luego le damos clic a enviar solicitud.



Maintenance / GetToken

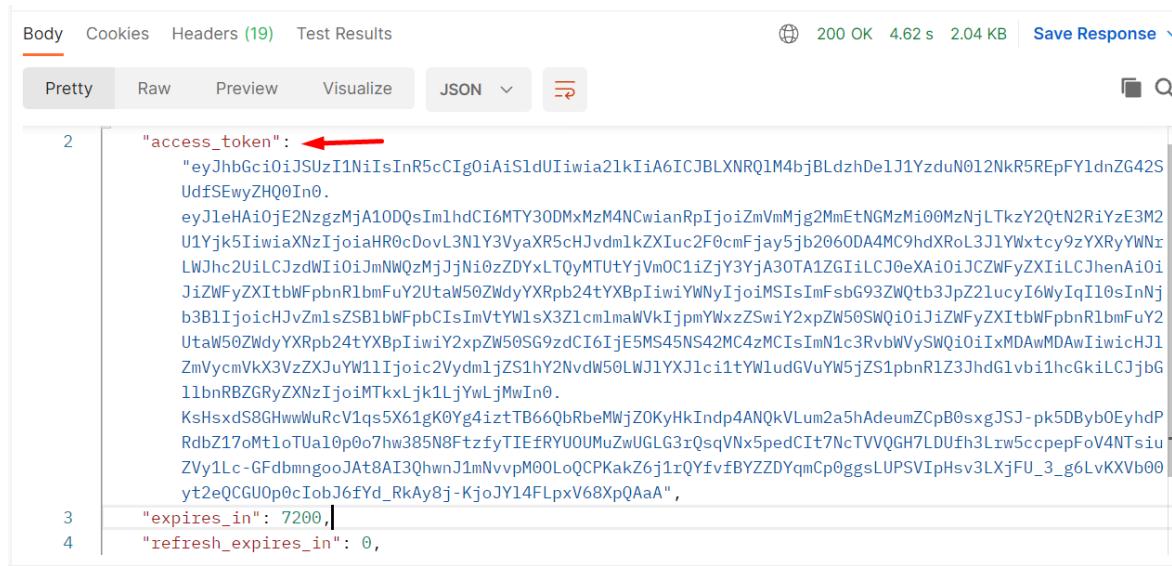
POST https://securityprovider.sattrack.com:8443/auth/realms/sattrack-base/protocol/openid-connect/token

Params Authorization Headers (11) Body Pre-request Script Tests Settings

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

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> grant_type	client_credentials			
<input checked="" type="checkbox"/> client_id	bearer-maintenance-integration-api			
<input checked="" type="checkbox"/> client_secret	edc14890-fedf-4fcf-85d1-d719a04bdf19			

En la respuesta del API como se muestra a continuación seleccionar el *access\_token*



Body Cookies Headers (19) Test Results

Pretty Raw Preview Visualize JSON 

2 "access\_token":   
 "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUiawiaw1kIiA6ICJBLXNRQ1M4bjBLdzhDeIj1YzduN012NkR5REpFY1dnZG42S  
 UdfSEwyZH0In0.  
 eyJleHAiOjE2NzgzMjA10DQsImhdCI6MTY30DMxMzM4NCwianRpIjoiZmVmMjg2MmEtNGMzMj00MzNjLTkzY2QtN2RiYzE3M2  
 U1Yjk5IiwiXNzIjoiaHR0cDovL3N1Y3VyaXR5cHJvdm1kZXIuc2F0cmFjay5jb2060DA4MC9hdXRoL3J1YWxtcy9zYXRyYWNi  
 LWJhc2UiLCJzdWIi0iJmNWQzMjJjNi0zDYxLTQyMTUtYjVmOC1iZjY3YjA30TA1ZGiilCJ0eXAi0iJCZWFyZXIiLCJhenAi0i  
 JjZWfyZXItbWFpbnR1bmFuY2UtaW50ZWdyYXRpb24tYXBPiwiYWNyIjoiMSIsImFsbG93ZWQtb3JpZ2lucyI6WylqIl0sInNj  
 b3B1IiiochJvZmlsZSB1bWFpbCIImVtYWlsX31cm1maWVkJpmYWxzZSwiY2xpZW50SWQi0iJizWFyZXItbWFpbnR1bmFuY2  
 Utaw50ZWdyYXRpb24tYXBPiwiY2xpZW50SG9zdCI6IjE5MS45NS42MC4zMCIsImN1c3RvbWVySWQi0iIxMDAwMDAwIiwichJ1  
 ZmVycmVX3VzZXJuYW1ljoic2VydmljZS1hY2NvdW50LWJ1YXJlcit1YWludGVuYW5jZS1pbnR1Z3JhdGlvbi1hcGkiLCJjbG  
 11bnRBZGRyZXNzIjoiMTkxLjk1LjYwLjMwIn0.  
 KshsxdS8GHwwWuRcV1qs5X61gK0Yg4iztTB66QbRbeMWjZOKyHkIndp4ANQkVLum2a5hAdeumZCpB0sxgJSJ-pk5DByb0EyhdP  
 Rdbz170Mt10Tu10p0o7hw385N8FtzfyTIEfRYU0UmuZwUGLG3r0sqVNx5pedC1t7nCTVVQGH7LDUfh3Lrw5ccpepFoV4NTsiu  
 ZVY1Lc-GFdbmgooJAt8AI3QhwnJ1mNvpM00LoQCPKakZ6j1rQYfvfBYZZDYqmCp0ggSLUPSVIpHsv3LXjFU\_3\_g6LvKXVb00  
 yt2eQCGU0p0cIobJ6fYd\_RkAy8j-KjoJY14FLpxV68XpQAaA",  
 3 "expires\_in": 7200,  
 4 "refresh\_expires\_in": 0,



Copiar la información dándole clic derecho al texto seleccionado.

POST

Params Aut Body Cookies

Pretty Raw

Set: pnd  
Set: Globals  
Undo  
Redo

Cut  
**Copy**  
Paste  
Select All

Find: eyJhbGciOiJSUzI1NlslnR5cClgOiAiSldeUiwia2lkIA6ICjBLXNRQIM4bjBLdzhDelJ1YzduN0l2NkR5REpFYldnZG42SudfSEwyZH0In0eyJle...  
EncodeURIComponent  
DecodeURIComponent  
b3B11joicHJVZm1sZSB1bWFpbC1sImVtYWlsX3Z1cm1maWVkJpmYWxzZSw1YzpZW50SWQ10iJ1ZWfyzX1tbWFpbmR1bmFuY2Utaw502WdyYXRpb24tYXBpIiwiY2xpZw50SG9zdC16IcjE5MS45NS42MC4zMCIeImN1c3RvbWVySwQ10iIxDawMDAwLiwich3lZmVycmVx3VzXJuYw11Ijoic2VydmljZS1hY2NvdW50LWJ1YXJlc1i1tYw1udGVuYW5jZS1pbnR1Z3JhdGlvbi1hcGkiLCJjbG11bnRBZGRyZXNzIjoimTlxLjk1LjYwLjMwIn0.  
n1W4F-cBEiBCQioB8g40V5qs4eYbXYzdj5XAtPubzgo99BYrsc0ebBzxhDjGFyN\_QqZW6xk-ueWkwKRhNdadULcUd7aaR1E4m  
eF54vds3KfdcdIkWggWi5sijgesl\_gLozesDjG-jy-bi6xYJ-qEnM6eSRU6eRZ063xxM1Q\_gxAL6DV\_DkVuDvSWB81UxiFxHZ  
xZA9kyukJbtgTB9I9C6qJZWK1qdFK3g0MbrbVz2QBgHfoeWcjsSHdDAXVap2X7rBszGBF7J7daExalamv48f\_-NERQ0ckhkg0  
T1TgG33uu0-VIca8j1WqqDQj3ipMNSp0WtssGPvy\_MvSWXfu",  
"expires\_in": 7200,

Runner Trash

## Obtener Histórico de mantenimientos:

Seleccionar de la colección importada el segundo request

APIs	Maintenance
○○	<b>POST</b> GetToken
	<b>GET</b> GetHistoricalMaintenance
□	<b>POST</b> CalibrateOdometer

## Configuración de la petición:

Autenticación:

En la pestaña Authorization y en el Type seleccionar: Bearer Token como se muestra en la siguiente imagen:

Maintenance / GetHistoricalMaintenance

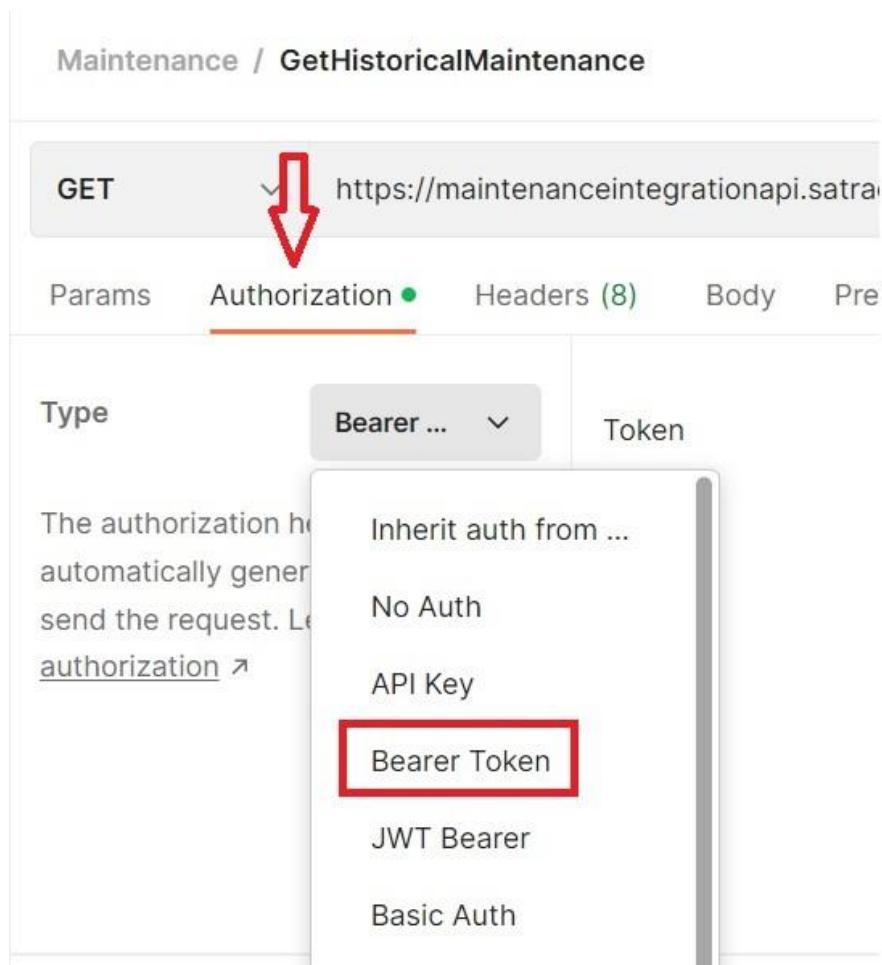
GET https://maintenanceintegrationapi.sattrack.com

Params    **Authorization** ●    Headers (8)    Body    Pre

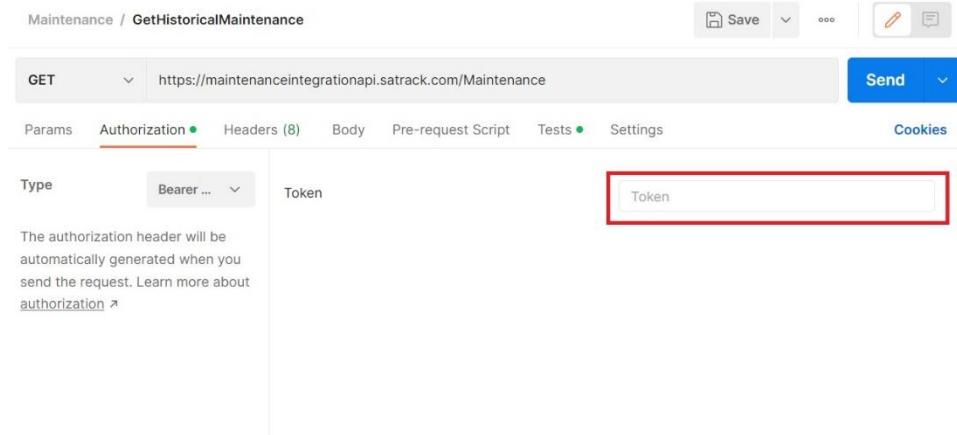
Type    **Bearer ...**    Token

The authorization header will be automatically generated and sent with the request. Learn more about [authorization](#).

Inherit auth from ...  
No Auth  
API Key  
**Bearer Token**  
JWT Bearer  
Basic Auth



Pegamos el token, previamente copiado en la siguiente casilla que se muestra en la imagen. (No hay que hacer ninguna acción, simplemente lo copiamos ahí).



Maintenance / GetHistoricalMaintenance

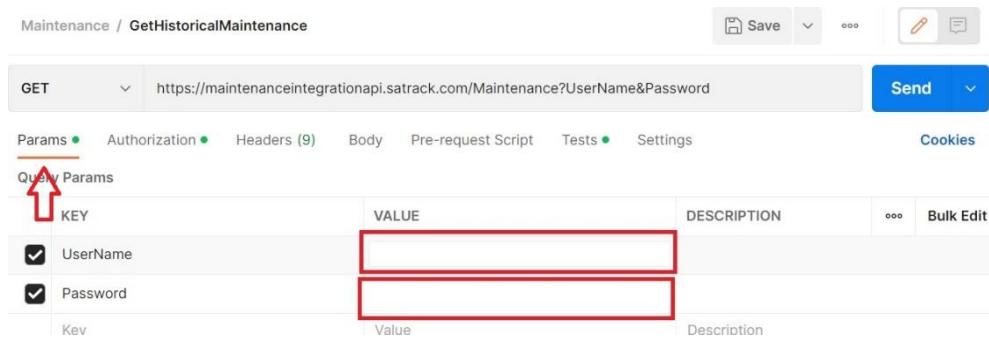
GET https://maintenanceintegrationapi.sattrack.com/Maintenance

Authorization Type: Bearer ... Value: Token

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

Ingresar Parametros:

En la pestaña Params, ingresamos el valor para la llave UserName y Password correspondientes.



Maintenance / GetHistoricalMaintenance

GET https://maintenanceintegrationapi.sattrack.com/Maintenance?UserName&Password

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

KEY	VALUE	DESCRIPTION	Bulk Edit
<input checked="" type="checkbox"/> UserName			
<input checked="" type="checkbox"/> Password			

Damos clic en enviar solicitud.



Una vez enviada la petición en la parte inferior en la etiqueta *Body* encontrará la respuesta, en caso de ser exitosa, mostrada la información con los campos: frecuencia, nombreMantenimiento, placa, restanteParaMantenimiento, unidadMedida, valorActual, valorProximoMantenimiento

Body Cookies Headers (7) Test Results

200 OK 731 ms 1.53 KB

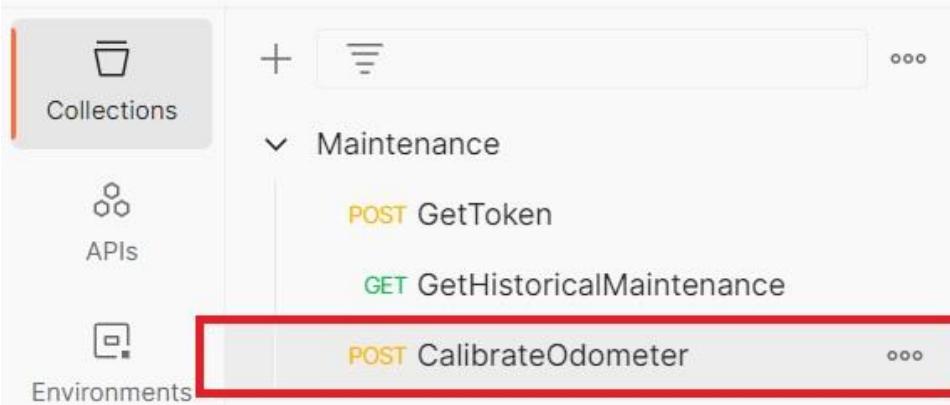
Pretty Raw Preview Visualize JSON ↗

```

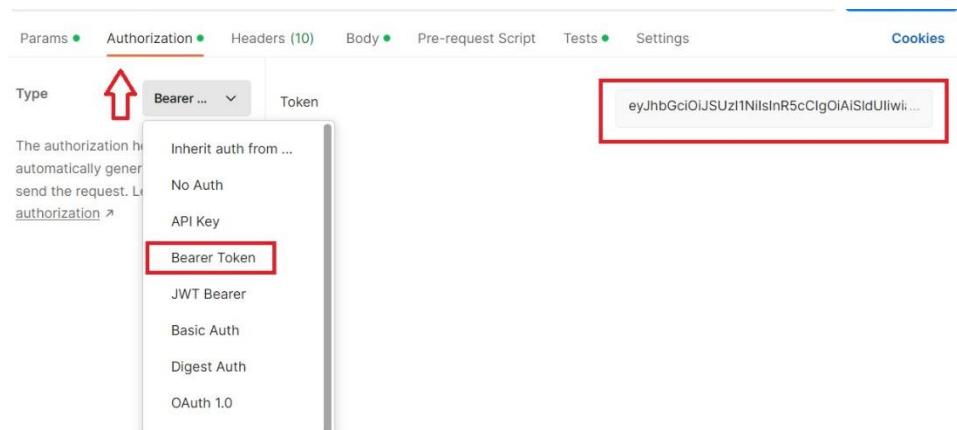
1 [
2   {
3     "frecuencia": 5000,
4     "nombreMantenimiento": "SOAT",
5     "placa": "CALAMP4",
6     "restanteParaMantenimiento": 54893,
7     "unidadMedida": "Kms",
8     "valorActual": "123",
9     "valorProximoMantenimiento": "55016"
10    },
11    {
12      "frecuencia": 5000,
13      "nombreMantenimiento": "SOAT",
14      "placa": "CALAMP2",
15      "restanteParaMantenimiento": -25007,
16      "unidadMedida": "Kms",
17      "valorActual": "30007",
18      "valorProximoMantenimiento": "5000"
19    }
]
```

## Calibración de odómetro:

Tercera petición, aquí podemos calibrar los odómetros. Seleccionamos CalibrateOdometer en el panel de la collection



En el primer apartado “Get Token”, indicamos como obtener el token, en este apartado, mostraremos, como se configura el token para esta petición. En la pestaña “Authorization” y seleccionando en Type: “Bearer Token”, copiamos el token, previamente generado y copiado, en la casilla ubicada en la parte superior derecha.

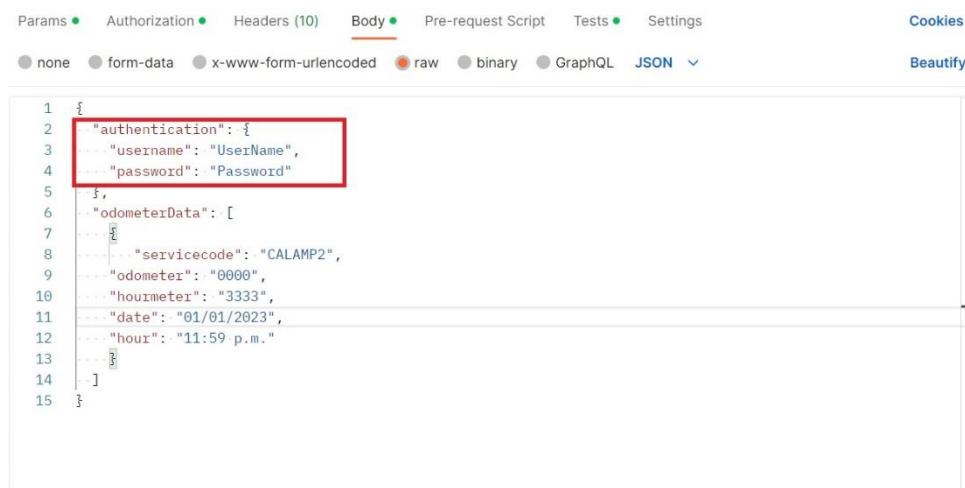


The screenshot shows the Postman interface with the Authorization tab selected. In the dropdown menu under 'Type', the 'Bearer Token' option is highlighted with a red box. To the right, a text input field contains a long JWT token, which is also highlighted with a red box.

Luego, en la pestaña Body configuraremos la petición.

## Configuración de la petición:

### Autenticación:



The screenshot shows the Postman interface with the Body tab selected. The request body is defined as a JSON object with two main fields: 'authentication' and 'odometerData'. The 'authentication' field contains 'username' and 'password' fields. The 'odometerData' field is an array containing a single object with 'servicecode', 'odometer', 'hourmeter', 'date', and 'hour' properties. The entire JSON structure is highlighted with a red box.

```

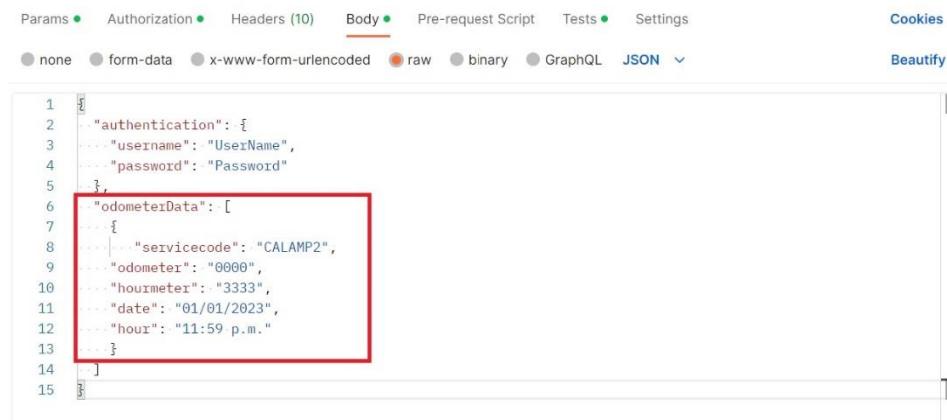
1 {
2   "authentication": {
3     "username": "UserName",
4     "password": "Password"
5   },
6   "odometerData": [
7     {
8       "servicecode": "CALAMP2",
9       "odometer": "0000",
10      "hourmeter": "3333",
11      "date": "01/01/2023",
12      "hour": "11:59 p.m."
13    }
14  ]
15 }

```

**username:** en este campo colocamos el usuario de la plataforma.

**Password:** en este campo colocamos la contraseña que tenemos en plataforma.

### odometerData:

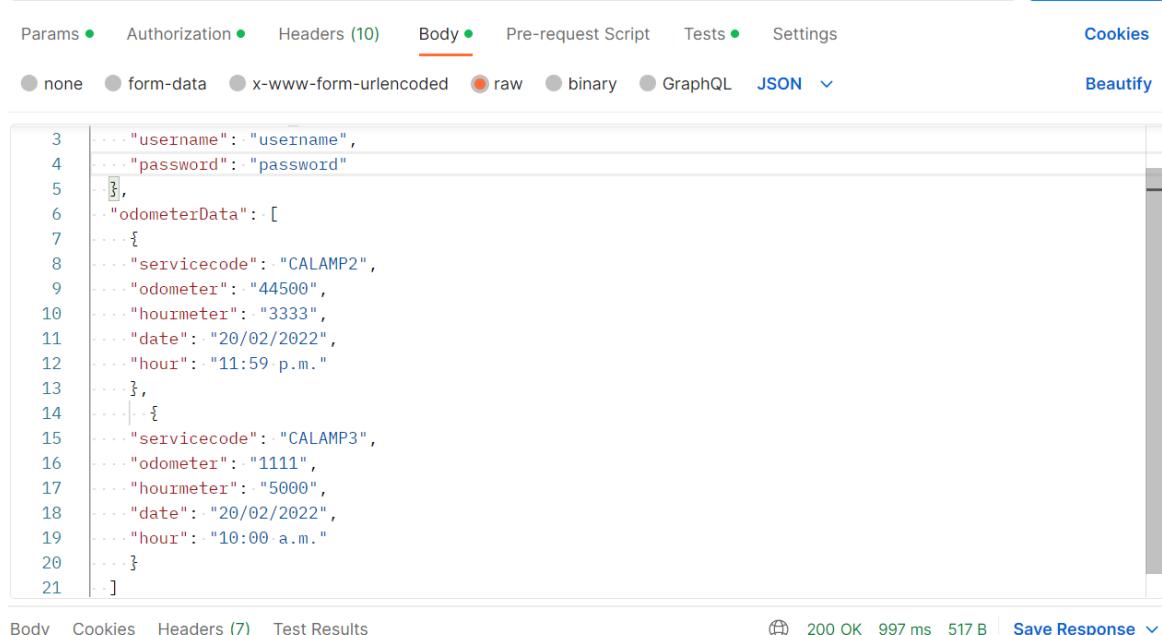


```

1
2   "authentication": {
3     "username": "UserName",
4     "password": "Password"
5   },
6   "odometerData": [
7     {
8       "servicecode": "CALAMP2",
9       "odometer": "0000",
10      "hourmeter": "3333",
11      "date": "01/01/2023",
12      "hour": "11:59 p.m."
13    }
14  ]
15

```

También podemos calibrar 2 o más placas, de esta manera:



```

3   "username": "username",
4   "password": "password"
5 },
6   "odometerData": [
7     {
8       "servicecode": "CALAMP2",
9       "odometer": "44500",
10      "hourmeter": "3333",
11      "date": "20/02/2022",
12      "hour": "11:59 p.m."
13    },
14    {
15       "servicecode": "CALAMP3",
16       "odometer": "1111",
17       "hourmeter": "5000",
18       "date": "20/02/2022",
19       "hour": "10:00 a.m."
20     }
21   ]

```

Body Cookies Headers (7) Test Results  200 OK 997 ms 517 B Save Response ▾



**servicode:** en este campo colocamos la placa del vehiculo a calibrar.

**odometer:** en este campo colocamos el valor del odometro

**hourmeter:** en este campo colocamos el valor del horometro.

**date:** en este campo colocamos la fecha que queremos registrar

**hour:** en este campo colocamos la hora.

Enviamos la petición dando clic en “Send”.

```
POST https://maintenanceintegrationapi.sattrack.com/Maintenance
Body (JSON)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

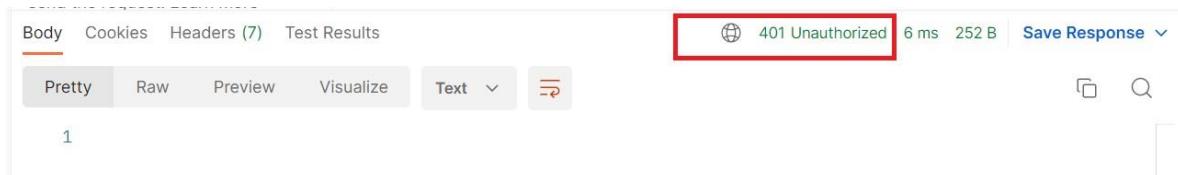
Una vez enviada la petición, y la información es correcta. En la parte inferior encontraremos en el Body

```
Body (Pretty)
1
2
3
4
5
6
```

Body Cookies Headers (7) Test Results 200 OK 607 ms 387 B Save Response

```
1
2
3
4
5
6
```

Si ingresamos el token invalido o no ingresamos el token, nos mostrara el siguiente mensaje de error.



A screenshot of the Postman application interface. At the top, there are tabs for 'Body', 'Cookies', 'Headers (7)', and 'Test Results'. On the right, there's a status bar showing '401 Unauthorized' with a red border around it, '6 ms', '252 B', and 'Save Response'. Below the tabs, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'Text' with a dropdown arrow. The main body area contains the number '1'.

Si ingresamos información con usuarios no validos



A screenshot of the Postman application interface. At the top, there are tabs for 'Body', 'Cookies', 'Headers (7)', and 'Test Results'. On the right, there's a status bar showing '200 OK' with a green border around it, '1099 ms', '312 B', and 'Save Response'. Below the tabs, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON' with a dropdown arrow. The main body area shows a JSON response with lines numbered 1 through 6:

```
1 [ {  
2   "message": "Usuario o clave inválidos",  
3   "parameters": "sand"  
4 } ]
```

Si ingresamos la información y faltan datos en la [authentication](#), nos mostrara este mensaje:



A screenshot of the Postman application interface. At the top, there are tabs for 'Body', 'Cookies', 'Headers (7)', and 'Test Results'. On the right, there's a status bar showing '200 OK' with a green border around it, '197 ms', '319 B', and 'Save Response'. Below the tabs, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON' with a dropdown arrow. The main body area shows a JSON response with lines numbered 1 through 6:

```
1 [ {  
2   "message": "Faltan datos para la autenticación",  
3   "parameters": null  
4 } ]
```



Si ingresamos la información y faltan datos en el `odometerData`, nos mostrara este mensaje:

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1  []
2  {
3      "message": "Parametros incompletos",
4      "parameters": null
5  }
6  []
```

### Límites:

La api tiene un posee una restricción de límite de peticiones de dos solicitudes cada 5 segundos, Con el fin de evitar saturar el web services de manteniendo.

Y el servicio mostrara el siguiente mensaje:

1 API calls quota exceeded! maximum admitted 2 per 5s.