

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:

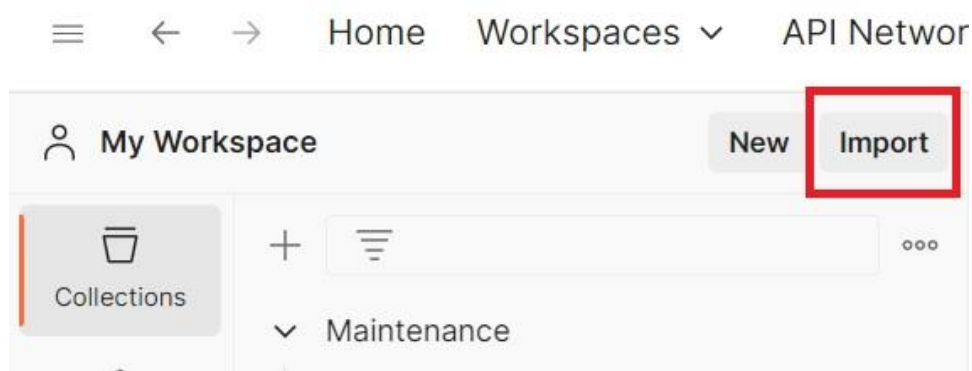
Explicar cómo integrarse con la API desarrollada por Satrack, para obtener el histórico de mantenimientos y poder calibrar su odómetro, a través del cual se podrá conectar el sistema de información propio con el aplicativo señalado.

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

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**



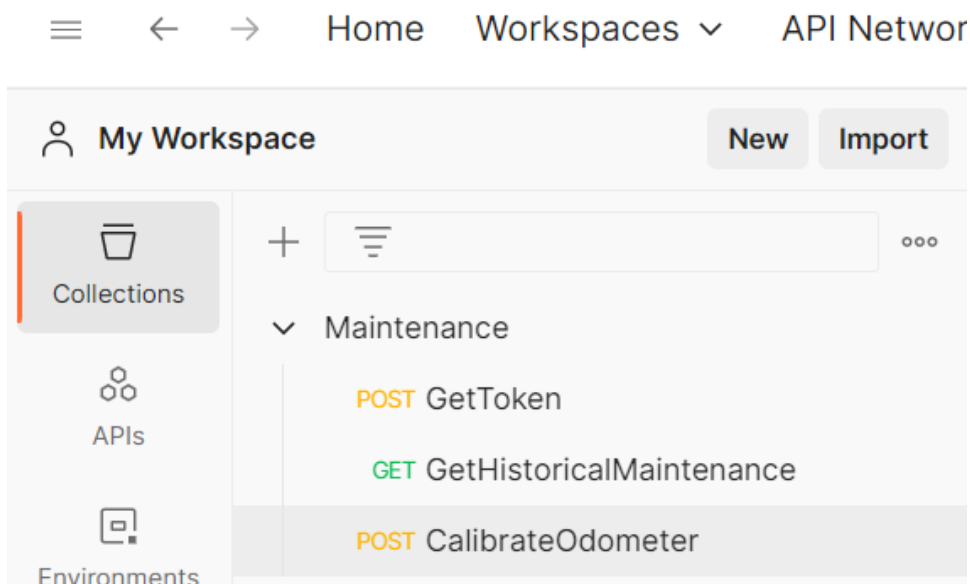
Choose Files

Seleccionar el archivo recibido por Satrack “**Maintenance-Integration.postman_collection.js**”

[Maintenance-Integration.postman_collection](#)

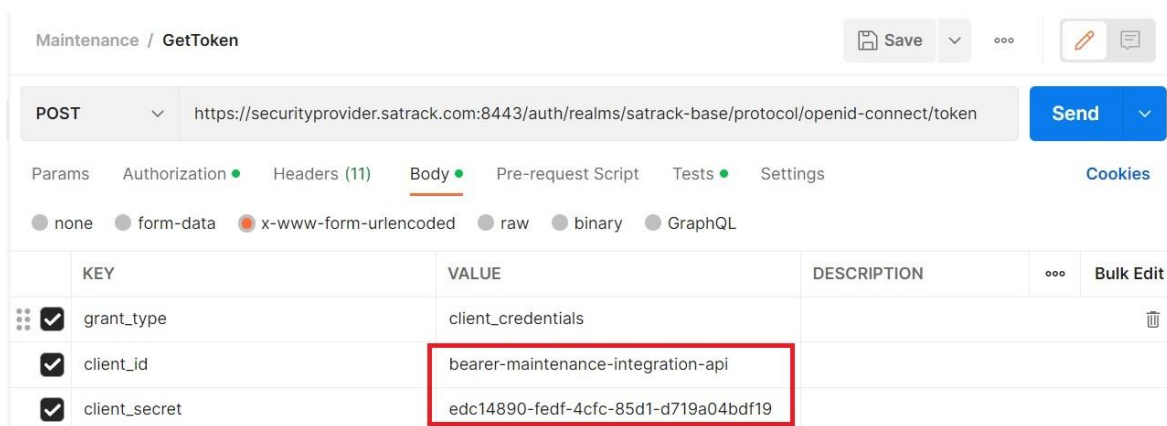
Uso de la API:

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

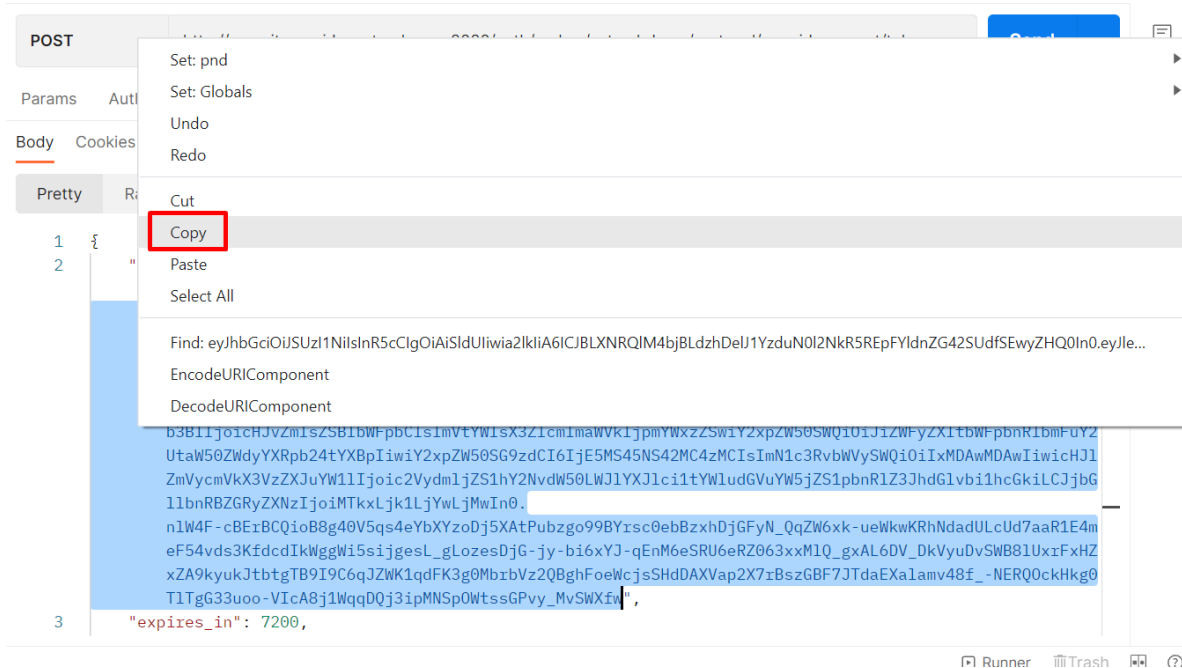


Obtener token para la autenticación:

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.

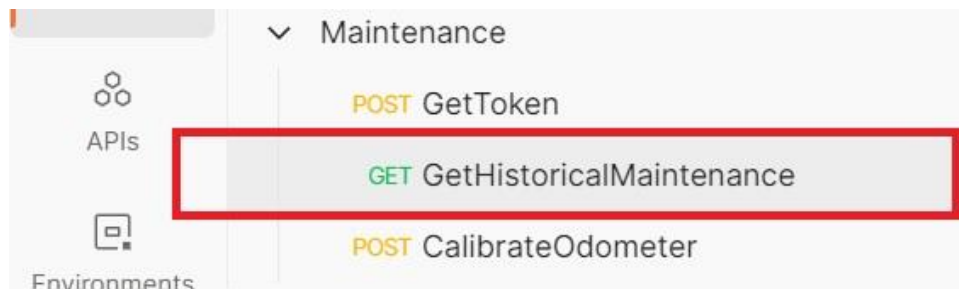


Copiar la información dándole clic derecho al texto seleccionado.



Obtener Histórico de mantenimientos:

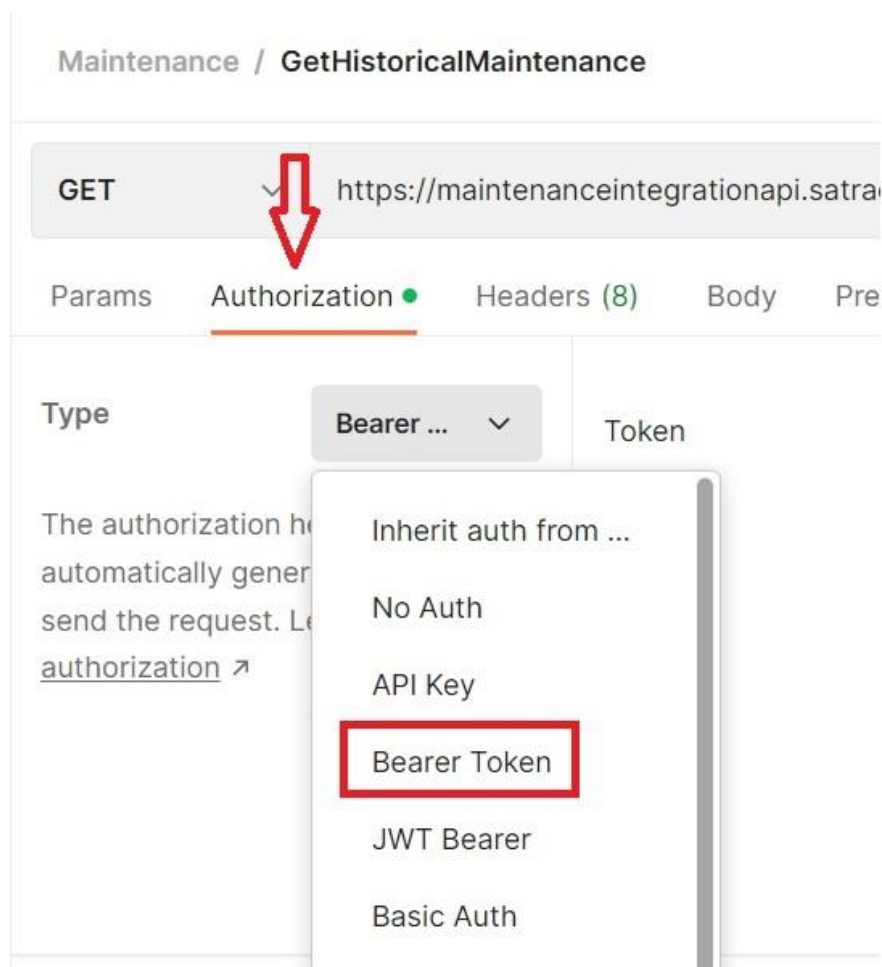
Seleccionar de la colección importada el segundo request



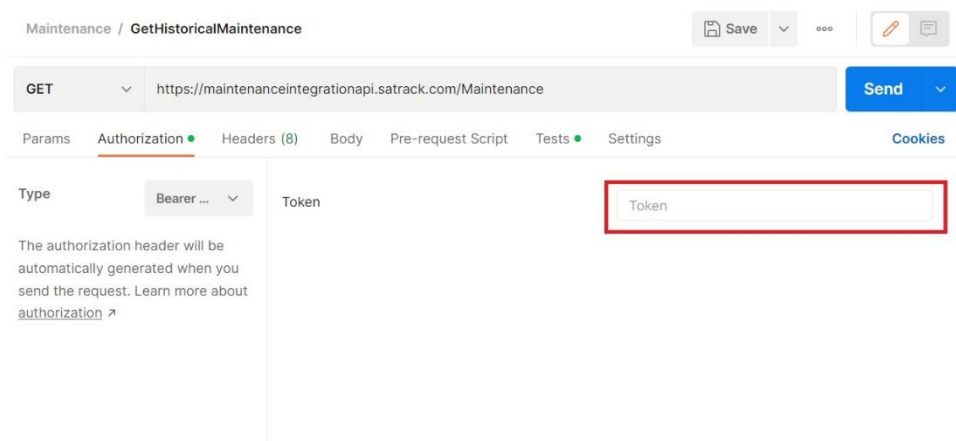
Configuración de la petición:

Autenticación:

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



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.satrack.com/Maintenance Send ▼

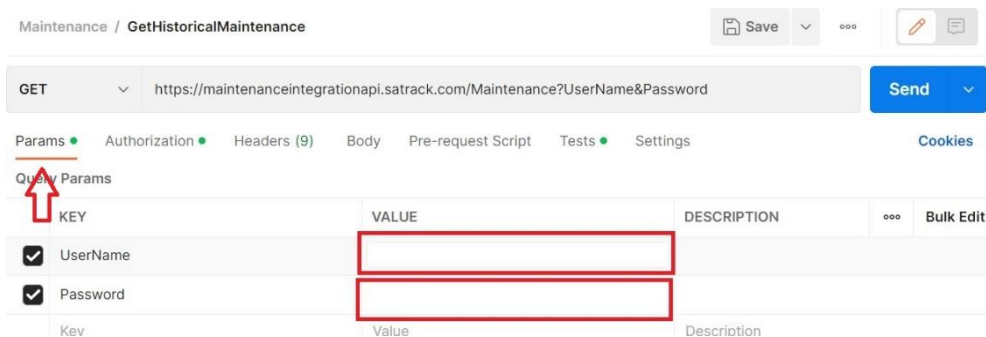
Params **Authorization** ▼ Headers (8) Body Pre-request Script Tests ▼ Settings Cookies

Type ▼ Bearer ... ▼ 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.satrack.com/Maintenance?UserName&Password Send ▼

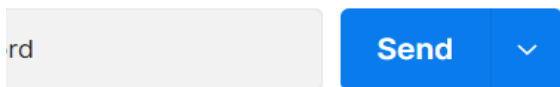
Params ▼ **Authorization** ▼ Headers (9) Body Pre-request Script Tests ▼ Settings Cookies

Query Params

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

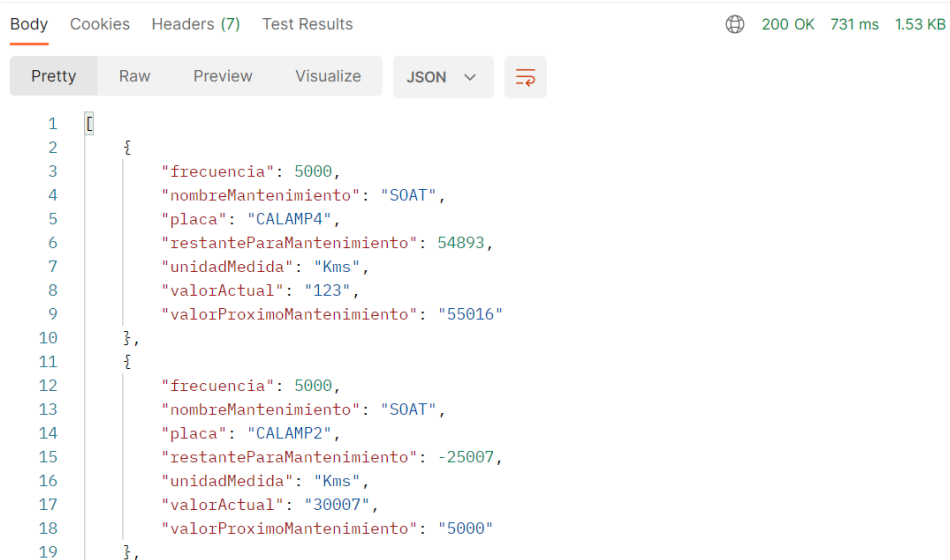
Kev Value Description

Damos clic en enviar solicitud.



rd Send ▼

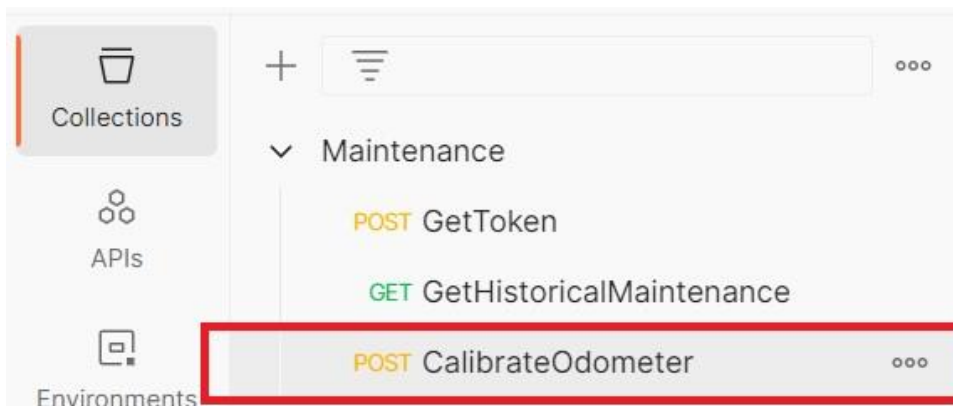
Una vez enviada la petición en la parte inferior en la etiqueta *Body* encontrará la respuesta, en caso de ser exitosa, mostrada la información con los campos: frecuencia, nombreMantenimiento, placa, restanteParaMantenimiento, unidadMedida, valorActual, valorProximoMantenimiento



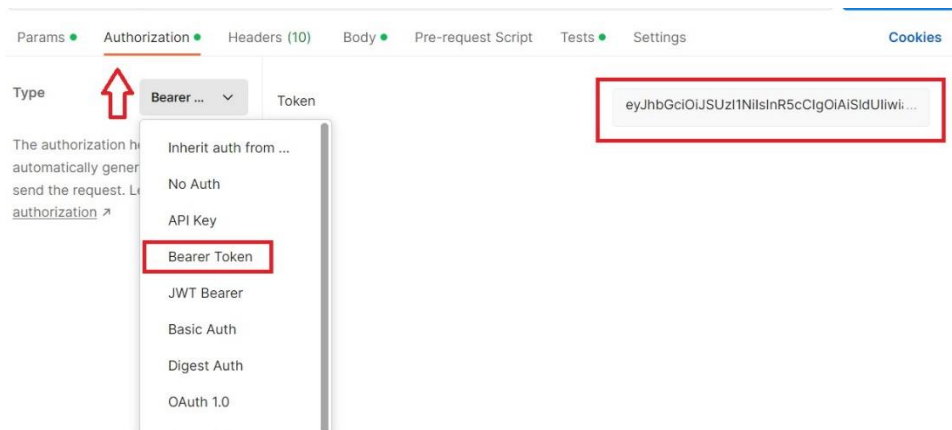
```
1 [
2   {
3     "frecuencia": 5000,
4     "nombreMantenimiento": "SOAT",
5     "placa": "CALAMP4",
6     "restanteParaMantenimiento": 54893,
7     "unidadMedida": "Kms",
8     "valorActual": "123",
9     "valorProximoMantenimiento": "55016"
10  },
11  {
12    "frecuencia": 5000,
13    "nombreMantenimiento": "SOAT",
14    "placa": "CALAMP2",
15    "restanteParaMantenimiento": -25007,
16    "unidadMedida": "Kms",
17    "valorActual": "30007",
18    "valorProximoMantenimiento": "5000"
19  },
20 ]
```

Calibración de odometro:

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



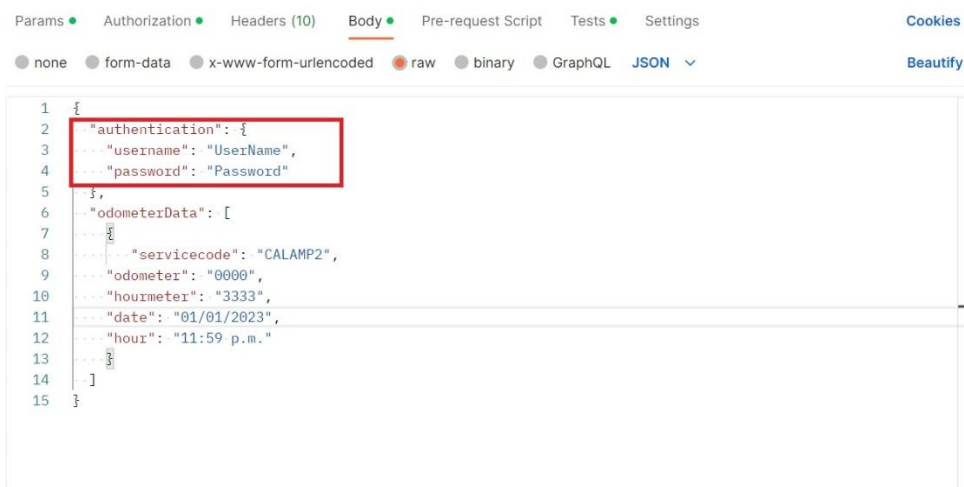
En el primer apartado “Get Token”, indicamos como obtener el token, en este apartado, mostraremos, como se configura el token para esta petición. En la pestaña “Authorization” y seleccionando en Type: “Bearer Token”, copiamos el token, previamente generado y copiado, en la casilla ubicada en la parte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



Luego, en la pestaña Body configuraremos la petición.

Configuración de la petición:

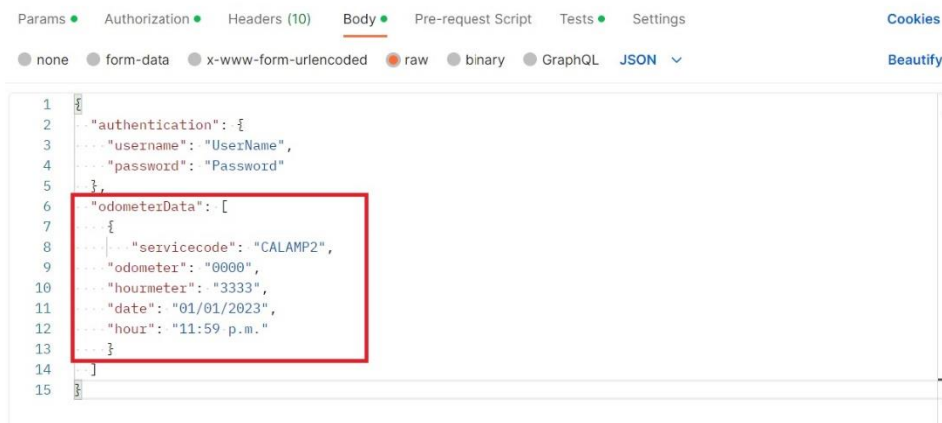
Autenticación:



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

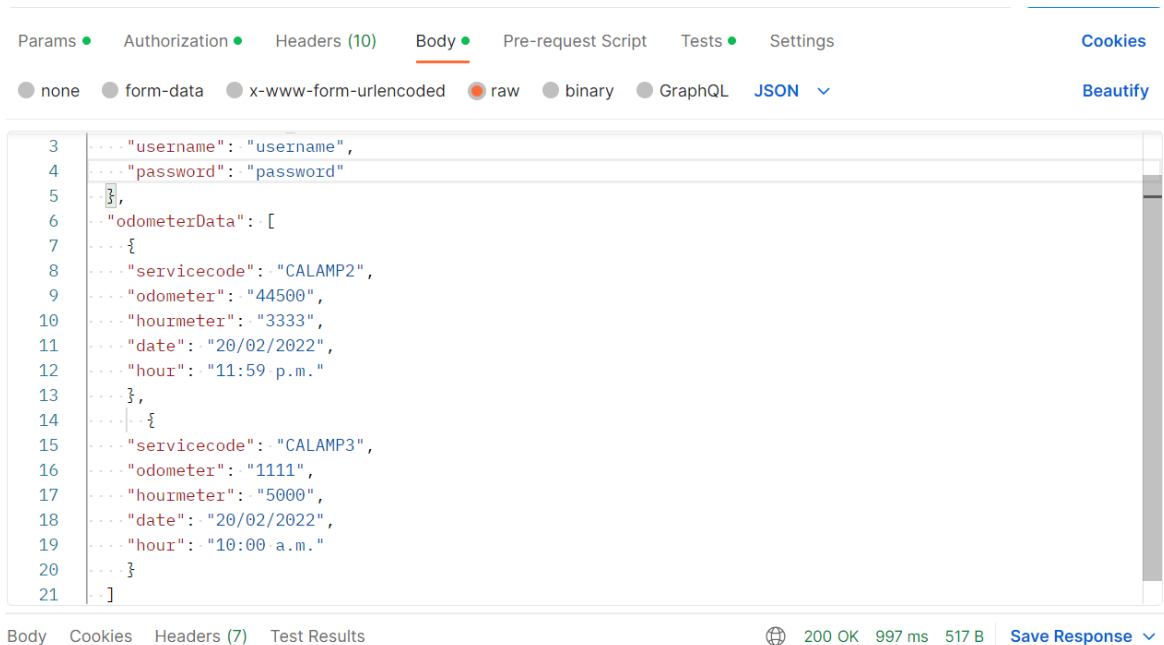
Password: en este campo colocamos la contraseña que tenemos en plataforma.

odometerData:



```
1  {
2    "authentication": {
3      "username": "UserName",
4      "password": "Password"
5    },
6    "odometerData": [
7      {
8        "servicecode": "CALAMP2",
9        "odometer": "0000",
10       "houmeter": "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  {
4    "username": "username",
5    "password": "password"
6  },
7  "odometerData": [
8    {
9      "servicecode": "CALAMP2",
10     "odometer": "44500",
11     "houmeter": "3333",
12     "date": "20/02/2022",
13     "hour": "11:59 p.m."
14   },
15   {
16     "servicecode": "CALAMP3",
17     "odometer": "1111",
18     "houmeter": "5000",
19     "date": "20/02/2022",
20     "hour": "10:00 a.m."
21   }
22 ]
```

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.

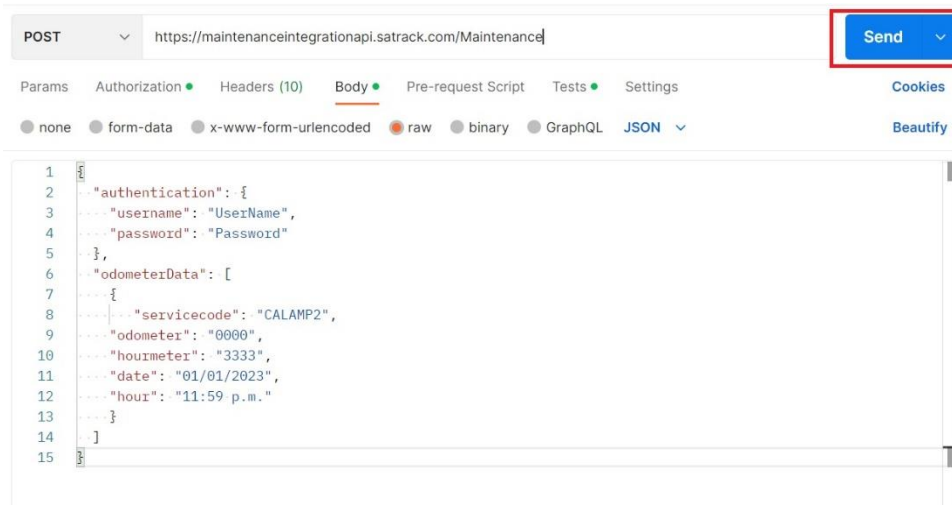
odometer: en este campo colocamos el valor del odometro

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.

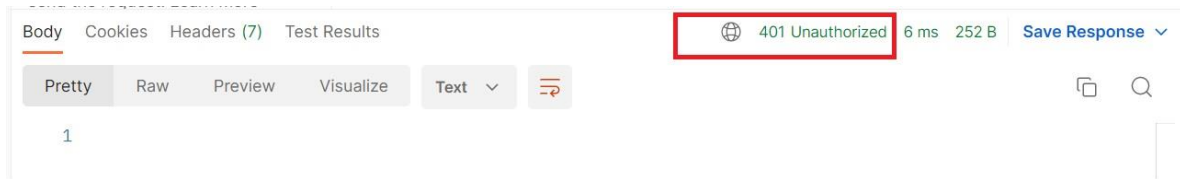
Enviamos la petición dando clic en “Send”.



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



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



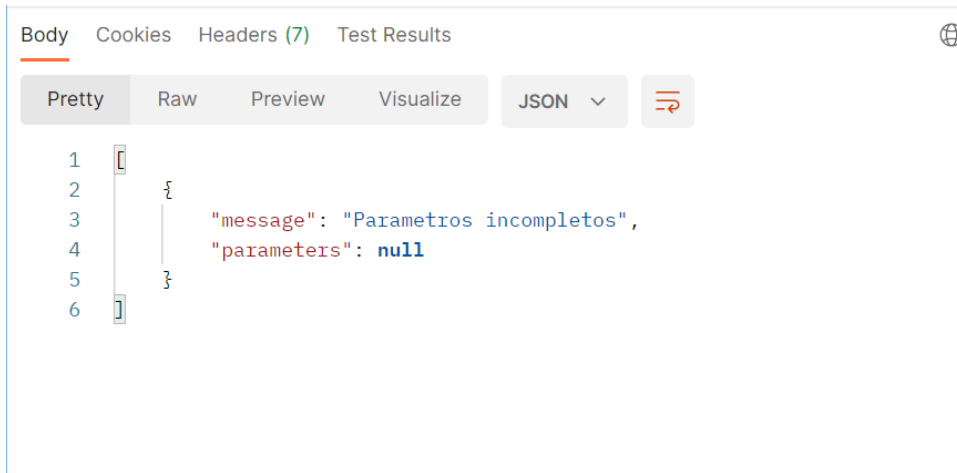
Si ingresamos información con usuarios no validos



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



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



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

Limites:

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.
```