**Como consumir datos de Optimus Fleet Manager Api**

El api utiliza lo que se conoce como Token-Based Authentication, lo cual permite que las credenciales de los usuarios no sean enviadas constantemente con cada petición que se realice al servidor, por lo tanto, se envía una única vez y el servidor retorna un token único con el cual se identificara al usuario que inicio sesión, este token debe ser almacenado por el cliente para poder utilizarlo cada vez que desee acceder a un recurso proporcionado por el api.

**Credenciales:**

|  |  |
| --- | --- |
| **username** | **Usuario proporcionado por el proveedor de servicios** |
| **password** | **Contraseña asignada al usuario** |
| **client\_id** | **Identificador de la aplicación con la cual se consumirán los datos del api. NOTA: este id será único para cada cliente.** |
| **client\_secret** | **Código secreto del cliente** |

Las credenciales **client\_Id** y **client\_secret** se utilizarán para acceder al api y son únicas para cada cliente, por lo tanto, no deben ser compartidas entre distintos clientes. Ejemplo: Si usted piensa consumir datos desde una aplicación de consola y desde una aplicación web usted deberá solicitar un id para cada aplicación.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Base Api Url:** http://rastreo.gps.hn:81

Esta será la dirección base mediante la cual se accederá al api

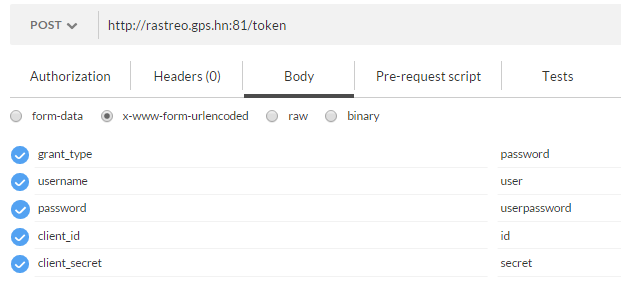
**Paso 1: Iniciar Sesión con las credenciales de usuario**

Haciendo uso de nuestras credenciales, iniciaremos sesión y solicitaremos el token al servidor.

**Url:** http://rastreo.gps.hn:81/token

**HttpMethod:** POST.

**ContentType:** application/x-www-form-urlencoded



**Ejemplo Javascript:**

****

**Respuesta del servidor:**

****

El servidor nos responderá con un objeto json el cual contiene la información que necesitamos para poder acceder a las distintas rutas del api.

El cliente tendrá que guardar el **access\_token** y el **token\_type**, ya que estos serán utilizados en todas las demás peticiones.

El cliente también tendrá que guardar el **refresh\_token**, este le ayudara a gestionar un nuevo token cuando el **access\_token** haya vencido.

**Errores:**

Si usted no envía los datos correctos el api responderá con un mensaje de error indicando cual es el posible problema por el cual usted no puede obtener el **access\_token**.

**Cliente no registrado : 400 Bad Request**

****

**Client secret invalido : 400 Bad Request**

****

**Username/Password Incorrecto : 400 Bad Request**

****

**Grant Type no soportado : 400 Bad Request**

****

**Paso 2: Solicitando datos al servidor**

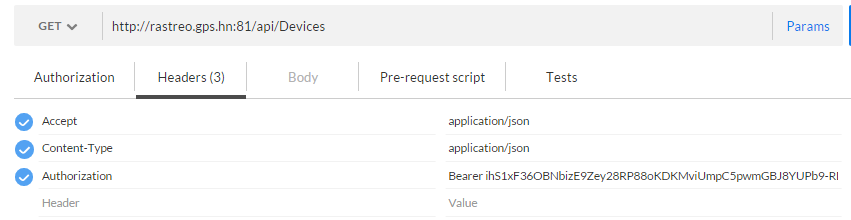
Una vez hayamos iniciado sesión y guardado el **access\_token**, **token\_type** y el **refresh\_token** correctamente, podremos comenzar a consumir datos de el servidor.

Cuando usted solicita datos al api ustede debe de proprcionar los **headers** adecuados para que la peticion sea validad, como se muestra en el ejemplo, recuerde que tiene que enviar el **token\_type y access\_token** en el header Authorization separados por un espacion

**Url:** [http://rastreo.gps.hn:81/[ApiRoute](http://rastreo.gps.hn:81/%5bApiRoute)]

Ejemplo: <http://rastreo.gps.hn:81/api/Devices>

**HttpMethod:** GET



**Ejemplo Javascript:**

****

**Respuesta del servidor: Status 200 OK**

****

El servidor nos responderá con un objeto json que contendrá la data solicitada.

**NOTA: RECUERDE QUE LOS TOKEN TIENEN UN TIEMPO DE EXPIRACION Y USTED TENDRA QUE HACER LAS GESTIONES NECESARIAS PARA PODER SOLICITAR UN NUEVO TOKEN CUANDO ESTE EXPIRE.**

**(EL TIEMPO DE EXPIRACION DEL TOKEN PUEDE SER MODIFICADO SIN PREVIO AVISO)**

**Errores:**

Cuando usted trate de acceder a una ruta específica y reciba este mensaje, es porque el token ya expiro y tendrá que gestionar un nuevo token.

**Autorización denegada: Status 401 Unauthorized**

****

**Recurso no encontrado:**  **404 Not Found**

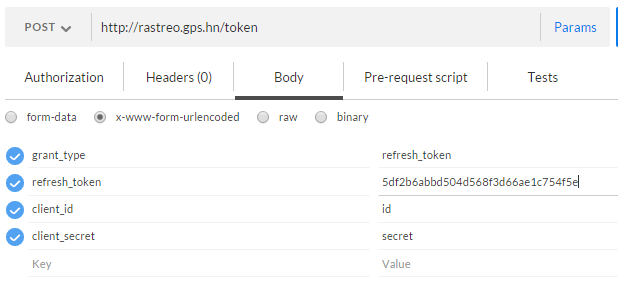
Cuando una ruta no esté disponible o ya no exista recibirá un mensaje como el siguiente

****

**calls quota exceeded: Status 429 API calls quota exceeded! maximum admitted 4 per Second.**

Esta respuesta indica que has tratado de hacer una mayor cantidad de peticiones que las permitidas en un determinado periodo de tiempo.

**Paso 3: Solicitar un nuevo Token**

Solicitamos un nuevo token, esta vez pandamos el valor “**refresh\_token**” mediante el parametro **grant\_type**, para que nuestra sesion pueda ser restaurada y nuevamente el servidor nos da una respuesta como la del **PASO 1** y almacenamos el nuevo access\_token y refresh\_token. 

**Ejemplo Javascript:**

****

**Respuesta del servidor: Status 200 OK**

****

**Rutas disponibles en el Api**

**Api/Devices**

Retorna todos los dispositivos pertenecientes a la compañía a la que pertenece el usuario que esta haciendo la petición.

**Parametros: ninguno**

**Api/Devices/LastReport/{id}**

Retorna el último reporte que el dispositivo envió al servidor

**Parametros:**

**\*Id – Id del dispositivo a buscar en el servidor Api/Devices/History/{id}?startdatetime=[dd/MM/yyyy HH:mm:ss]&enddatetime==[dd/MM/yyyy HH:mm:ss]&limit=2000&offset=0**

Retorna el historial de reportes para un dispositivo.

**Formato:**

El formato para la fecha es **dd/MM/yyyy HH:mm:ss**.

**Parametros:**

**\*Id – Id del dispositivo a buscar en el servidor**

**Startdatetime y Enddatetime –** Rango de fecha desde y hasta donde se buscaran los reportes.

Tome en cuenta que si solo envía startdatetimese tomaran todos los reportes con fecha menor a la misma.

**Limit –** por default 2000, es la cantidad de reportes que retornara el servidor por cada request,

El límite de reportes por cada request es de 2000.

**Offset –** por default 0, es el índice de la cantidad de reportes a omitir.

**Ejemplo.**

Si el dispositivo [id] tiene un historial de 7000 registros será necesario hacer varios request para poder obtenerlos todos.

1. api/Devices/History/id?startdatetime=27/11/2015 11:36:33&offset=0

Response: 2000 registros, omitio 0 registros

1. api/Devices/History/id?startdatetime=27/11/2015 11:36:33&offset=1

Response: 2000 registros, omitio 2000 registros

1. api/Devices/History/id?startdatetime=27/11/2015 11:36:33&offset=2

Response: 2000 registros, omitio 4000 registros

1. api/Devices/History/id?startdatetime=27/11/2015 11:36:33&offset=3

Response: 1000 registros, omitio 6000 registros

1. api/Devices/History/id?startdatetime=27/11/2015 11:36:33&offset=4

Response: 0 registros

**NOTA: como ya no mas registros es qui donde paramos las peticiones al servidor.**

**\*OBLIGATORIO**