

JSON API Interface (optional)



Our GPS Fleet Software has a JSON API interface in order to integrate with 3rd party software solutions (ERP, etc.). The API JSON interface works for our web-based hosting accounts as well as for local server customers (if the network allows it).

The availability of the JSON interface depends on our offers and standard conditions. Please contact your provider / support for more information and test interface data.

- Versions
- How to access the API
- How to find the different JSON calls and parameters
- Basic examples
 - Query for existing assets
 - Get the current position
 - Get historic trips
 - Get drivers
 - Create drivers
 - Get logbook entries
 - Create logbook project codes

Versions

Date	Version	Comments
06.11.19	3.8.50	<p>getGpsHistory Abfrage mit neuen Parameter für Startdatum und Enddatum</p> <p>getGpsHistory Antwort mit neuen Feldern (Temperatur, analoger Eingang)</p> <p>driver/create Aufruf für das Anlegen/Ändern von Fahrern</p> <p>building_site_codes Aufruf für das Anlegen /Ändern/Löschen von Projekt-Codes</p>

How to access the API

The API can only be used with an APIKEY that can only be set by our support.

<http://XXX.sw-management.at/GFS/api/std/checkuser?apikey=apikey>

The above command will return OK or ERROR if the apikey is correct or invalid (link must be changed).

```
{"info":"ok"} or {"error":"100"}
```

How to find the different JSON calls and parameters

Developers can access the WADL document on our hosting server (or on the local server of the customer):

<http://gfs2.sw-management.at/GFS/api/application.wadl>

The WADL document explains the calls and parameters in detail, such as the following example:

```
<resource path="/std">
  <resource path="/checkuser">
    <method id="checkuser" name="GET">
      <request>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="apikey" style="query" type="xs:string"/>
      </request>
      <response>
        <representation mediaType="application/json"/>
      </response>
    </method>
  </resource>
```

Basic examples

Query for existing assets

Use the the following http call to query which assets are in a customer account. You will need the assetIDs for other commands.

<https://XXX.sw-management.at/GFS/api/masterdata/gps/getAssetStates?apikey=ENTERYOURSECID&langid=2>

xxx ... replace the link

ENTERYOURSECID ... the APIKEY that is set for the account

Get the current position

https://XXX.sw-management.at/GFS/api/masterdata/gps/getLastpos_for_assetid?apikey=ENERYOURSECID&assetid=YYYY

xxx ... replace the link

ENTERYOURSECID ... the APIKEY that is set for the account

YYYY ... assetid

Get historic trips

https://XXX.sw-management.at/GFS/api/masterdata/gps/getGpsHistory?apikey=xmlYYYYYYYYYYYY&opt_assetid_filter=XXX&opt_datestr_from_yyyyMMddhhmmss=20191015000000&opt_datestr_to_yyyyMMddhhmmss=20191016000000

xxx ... replace the link

ENTERYOURSECID ... the APIKEY that is set for the account

Get drivers

<https://XXX.sw-management.at/GFS/api/masterdata/driver/getDrivers?apikey=ENTERYOURSECID>

Create drivers

Using a POST you can create new drivers

<https://XXX.sw-management.at/GFS/api/masterdata/driver/create?apikey=ENTERYOURSECID>

{

```
"shortname": "maxm",  
"username": "Max Maier",  
"mobile": "mobile number, not required",  
"email": "test@example.com",  
"password": "Max Maier"  
}
```

Get logbook entries

https://XXX.sw-management.at/GFS/api/masterdata/logbook/getLogbookData_for_assetid?apikey=ENTERYOURSECID&assetid=YYYY&showprivate=true&trips=true&stops=false&opt_datestr_from_yyyyMMddhhmss=20171013000000

Create logbook project codes

Using a POST you can create or delete project codes

https://XXX.sw-management.at/GFS/api/masterdata/logbook/building_site_codes?apikey=ENTERYOURSECID

```
codes: [  
  {  
    code: "B1917",  
    title: "Großmarkt",  
    address: "9999 Klöszß, Bahnhof 13",  
    from: 1571664554000,  
    to: 1572684554000,  
    people: [163,164]  
  }  
],  
toDelete: []
```