

JSON API (optional, kostenpflichtig)



Unsere GPS Fleet Software API erlaubt es Software von dritten Daten von unserem System abzurufen und Daten einzuspielen um kooperative Programme und Apps zu realisieren. Mit unserer API ermöglichen wir es anderen Entwicklern eine Synergie zwischen unserer GPS Fleet Software und deren Server oder App herzustellen. Für Anwendungsfälle in der folgenden Dokumentation werde ich unser Hosting benutzen, die API ist jedoch auch mit einem eigenen Server einsetzbar, allerdings hängt die Funktion dann von der Erreichbarkeit des Servers ab.

Die Bereitstellung und der Support für Schnittstellen erfolgt gemäß unserer Angebotskonditionen und erfordert zumeist eine Geheimhaltungsvereinbarung. Bitte wenden Sie sich bei Fragen an unseren Support! Änderungen/Weiterentwicklungen und Einstellungen sind möglich. Für einen Demo-Zugriff (Schnittstellen-Tests) wenden Sie sich bitte an unseren Support.

- Versionierung
- Wie greift man auf die API zu
- Wo findet man die verschiedenen Aufrufe und Parameter
- Grundlegende Aufrufe und Beispiele
 - Fahrzeugstammdaten abrufen
 - Aktuelle Fahrzeugpositionen abrufen
 - Historische Rohdaten (Fahrspur) abrufen
 - Aktuelle Fahrer abrufen
 - Fahrer anlegen
 - Fahrtenbuch-Einträge abfragen
 - Fahrtenbuch-Projekt-Codes anlegen (Projekt-Codes)
- Weitere Aufrufe
 - modify
 - time
 - masterdata
 - bin
 - odr
- Aufruf für den Versand von Touren

Versionierung

Datum	Version	Änderung
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>

Wie greift man auf die API zu

Um auf die API zuzugreifen wird der selbe Link verwendet, der auch zum Einloggen als normaler User in unsere Software verwendet wird. Für jeden Zugriff ist eine Verifikation notwendig, die über Username und Passwort oder einen API Schlüssel, der in der Software verfügbar ist, erfolgt. Das einfachste Anwendungsbeispiel ist die Überprüfung ob ein User zur Software zulässig ist. In diesem Fall würde der Aufruf so aussehen:

<http://gfs2.sw-management.at/GFS/api/std/checkuser?apikey=user----password>

oder

<http://gfs2.sw-management.at/GFS/api/std/checkuser?apikey=apikey>

Der APIKEY wird im Software-Zugang bei den Optionen gesetzt und kann auch verändert werden.

Die Antwort wird dann je nachdem ob der User akzeptiert wird so aussehen:

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

Wo findet man die verschiedenen Aufrufe und Parameter

Alle Möglichkeiten , Daten von unserem System abzurufen bzw zu unserem System zu schicken, findet man in einem zentralen XML Dokument, das alle Informationen enthält wie notwendige und optionale Parameter und deren Typ.

Das WADL-Dokument mit dem ganzen Überblick ist hier zu finden: <http://gfs2.sw-management.at/GFS/api/application.wadl> (oder wenn die GFS auf einem anderen Server läuft unter Serverlink/api/application.wadl).

Zum Beispiel der checkuser Aufruf, der oben zu sehen ist, würde so in unserem application.wadl aussehen:

```
<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>
```

Grundlegende Aufrufe und Beispiele

Die grundlegendsten Aufrufe befinden sich unter masterdata. Unter masterdata sind weitere Unterpunkte für verschiedene Objekte die unser Server bereitstellen kann. Zum Beispiel asset, geofence oder driver. Die grundlegendsten Aufrufe werden so aussehen:

<http://XXX.sw-management.at/GFS/api/masterdata/asset/getAssets?apikey=user----password>,

<http://XXX.sw-management.at/GFS/api/masterdata/geofence/getGeofences?apikey=user----password> or

<http://XXX.sw-management.at/GFS/api/masterdata/driver/getDrivers?apikey=user----password>

Fahrzeugstammdaten abrufen

Mit dem folgenden Aufruf wird abgefragt, welche Geräte im Kunden-Account verfügbar sind. Wichtig ist der Fahrzeugname und die **Geräte-Nummer (assetid) für weitere Abfragen**.

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

xxx ... mit richtigen Zugangsdaten ersetzen

ENTERYOURSECID ... mit der Sicherheitsnummer für den Kunden-Account ersetzen (gesetzt vom Support in den Account-Optionen).

Aktuelle Fahrzeugpositionen abrufen

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

xxx ... mit richtigen Zugangsdaten ersetzen

ENTERYOURSECID ... mit der Sicherheitsnummer für den Kunden-Account ersetzen (gesetzt vom Support in den Account-Optionen).

YYYY ... assetid (siehe Abfrage Fahrzeugstammdaten)

Historische Rohdaten (Fahrspur) abrufen

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 ... mit richtigen Zugangsdaten ersetzen

ENTERYOURSECID ... mit der Sicherheitsnummer für den Kunden-Account ersetzen (gesetzt vom Support in den Account-Optionen).

Aktuelle Fahrer abrufen

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

Fahrer anlegen

Mittels Post-Aufruf können Fahrer angelegt werden

<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"
}
```

Fahrtenbuch-Einträge abfragen

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_yyyyMMddhhmmss=20171013000000

Fahrtenbuch-Projekt-Codes anlegen (Projekt-Codes)

Mittels Post-Aufruf können Projekt-Codes angelegt werden

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

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

Weitere Aufrufe

modify

- /logbook/set_operating_hours: diesen Befehl mit den Parametern apikey, assetid und new_h_value aufzurufen setzt einen neuen Wert für die Betriebsstunden eines Fahrzeugs.
- /insertpos: diesen Befehl aufzurufen mit den Parametern apikey, assetid, lat, lng, d1, speed, ts_yyyyMMddhhmmss und optional opt_fuellevel setzt eine neue Position für ein Fahrzeug inklusive Zündung, Geschwindigkeit und der Zeit an der entsprechend Stelle und optional den Tankfüllstand.
- /insertpos_address: diesen Befehl aufzurufen mit den Parametern apikey, assetid, address, d1, speed, ts_yyyyMMddhhmmss und optional opt_fuellevel setzt eine neue Position für ein Fahrzeug an der entsprechenden Adresse.

time

- /start_time_record: diesen Befehl mit den Parametern apikey, start_ts_yyyyMMddhhmmss, personid und optional opt_assetid, opt_custid_geofenceid, opt_activityid, opt_commentstr, opt_lat, opt_lng und opt_address aufzurufen startet einen neuen Zeit Eintrag für die Person mit der angegebenen Zeit als Startzeit und fügt optional das Fahrzeug, den Geofence, die Position, oder Adresse an der die Person arbeitet, einen Kommentar und die Aktivität die die Person ausführt entsprechend der optionalen Parameter.
- /update_time_record: diesen Befehl mit den Parametern apikey, end_ts_yyyyMMddhhmmss, personid und optional opt_assetid, opt_custid_geofenceid, opt_activityid, opt_commentstr, opt_lat, opt_lng and opt_address aufzurufen schließt den letzten Zeiteintrag für die Person mit der gegebenen Zeit ab und fügt optional das benutzte Fahrzeug, den Geofence, die Position, oder die Adresse an der die Person gearbeitet hat, einen Kommentar und die Aktivität je nach den optionalen Parametern hinzu. Im Falle eines Fahrzeugs können die Anfangs- bzw Endposition/-adresse unterschiedlich sein.
- /insert_complete_time_record: diesen Befehl mit den Parametern apikey, start_ts_yyyyMMddhhmmss, end_ts_yyyyMMddhhmmss, personid und optional opt_assetid, opt_custid_geofenceid, opt_activityid, opt_commentstr, opt_lat, opt_lng und opt_address aufzurufen erzeugt einen kompletten Zeiteintrag mit der gegebenen Start- und Endzeit und wird optional das benutzte Fahrzeug, den Geofence, die Position, oder die Adresse an der die Person gearbeitet hat, einen Kommentar und die Aktivität je nach den optionalen Parametern hinzufügen.

masterdata

- /data/getLastTimeRecord: dieser Befehl mit den Parametern apikey und personid gibt den letzten Zeiteintrag dieser Person zurück.
- /data/getActivityTypes: diesen Befehl mit dem apikey aufzurufen gibt alle Aktivitäten des Accounts zurück.
- /asset/getAssets_by_assetkey: diesen Befehl mit dem apikey und dem assetkey auszuführen gibt das Fahrzeug mit dem entsprechenden Schlüssel zurück. (Der Schlüssel muss in der GPS Fleet Software gesetzt werden und kann genutzt werden um auf ein Fremdsystem zu referenzieren)

bin

- /upload: diesen Befehl aufzurufen erlaubt es eine Datei zum Server hochzuladen (in den meisten Fällen ein Bild). Wie immer wird der apikey als Parameter benötigt, optional können opt_relationID, opt_pkid, opt_category, opt_name, opt_commentinfo, opt_binid gesetzt werden um die Relation, das Objekt zu dem die Datei gehört, die Kategorie, der Name oder ein Kommentar hinzuzufügen, weiters kann eine existierende Datei über die binid überschrieben werden.

odr

- /getDynDataValues: diesen Befehl aufzurufen mit dem apikey und optional opt_limit, opt_relationID oder opt_pkid zum filtern wird die dynamischen Felder des Accounts zurückgeben die vom User in der GPS Fleet Software erstellt und konfiguriert werden können und es erlauben zusätzliche Felder für Objekte zu setzen.
- /setDynDataValue_for_pkid_defid_relationID: diesen Befehl mit den Parametern apikey, pkid, relationID, defid and one parameter of v_int, v_str, v_bigint, v_float, v_date_yyyyMMddhhmmss, v_boolean aufzurufen wird ein gegebenes Feld des gegebenen Objekts auf den gegebenen Wert ändern. Wichtig: der korrekte Parameter für den Wert muss verwendet werden um das Feld zu überschreiben.

Aufruf für den Versand von Touren

Der folgende Aufruf erstellt eine Tour mit 2 Kunden-Adressen, die an ein Fahrzeug/einen Fahrer geschickt werden. Weitere Felder (Extra-Daten) können optional hinzugefügt werden (siehe wadl).

https://uat.sw-management.at/GFS/api/email_job/email_sendJobAddress?apikey=xml1618924879-2204044689549486811-6442706128315387993&assetid=102&sendstore_also_if_unrouted_stops=true&tourname=json35&store=true&send=true&custnames_xx99xx_separated=Kunde19xx99xxKunde20&custaddresses_xx99xx_separated=8200%20Albersdorf%20209Gleisdorfxx99xx8160%20Weiz%20Schmied%20in%20der%20Weiz%20146&opt_personid=402

Liefert die Tourstops direkt an den Fahrer.

Achtung, beim Testaufruf muss der Tourname immer geändert werden!

Vertrieb 1

Tourjson34 (272) 09.03.2016 00:00

Anlegedatum: 09.03.2016 00:00

Ankunft	Geofence Name	Kommentar	Teil	Adresse	App	Start	Extradata
00:00	Kunde17		8200 Albersdorf 209Gleisdorf	Albersdorf 209, 8200 Albersdorf, Österreich, 8200 Albersdorf	LINK	00:00	Departure: 07:00 Ankunft: 07:00 Dist. [km]: 0.0
00:00	Kunde18		8160 Weiz Schmied in der Weiz 146	Siedlungsweg Schmied in der Weiz 146, 8160, Österreich, 8160	LINK	00:00	Departure: 07:30 Ankunft: 07:30 Dist. [km]: 0.027

Das ist der Footer Email Text am Ende des Auftrags

272
json34
09.03.2016 00:00
Standard Tour

Vertrieb 1

Vertrieb KH

0
0

Departure: 07:00
Ankunft: 07:30
Entfernung [km]: 0