

Geckollect.com

Mérés & adatgyűjtés

Publikus API dokumentáció

Geckollect.com

Dátum: 2022. november 22.

Verzió: v1.1

Megjegyzés

Publikus dokumentáció, korlátozás nélkül tejeszthető. Kiadásra jelölt verzió.

Tartalomjegyzék

1. Általános információk	2
1.1. Az API célja	2
1.2. A használat feltételei	2
1.3. Szerverek	2
1.4. Protokoll	2
2. Interfész	4
2.1. Összefoglalás	4
2.2. Végpontok specifikációja	5
2.2.1. /data/last	6
2.2.2. /data/list	8
2.2.3. /device/status	10
3. Támogatás	13
3.1. Kérdések	13
3.2. Jogi kikötések	13

1. fejezet

Általános információk

1.1. Az API célja

Az API célja, hogy a Geckollect.com szolgáltatás egyszerűen és egységesen integrálható legyen harmadik fél által üzemeltetett alkalmazásokba, szoftverekbe.

További távlati cél, hogy publikus API-n keresztül külső rendszer adatot szolgáltatthasson a Geckollect.com megjelenítő felületre.

1.2. A használat feltételei

A Geckollect.com API használatához szükséges:

- a rendszerrel kompatibilis adatgyűjtő (esetleg rádiós) egységre
- a webalkalmazásban elérhető API kulcsra, mely adatgyűjtőre vagy mérőre vonatkozhat
- mérési adatok esetén az API engedélyezésére az adott mérőn

A szolgáltatás használatának feltételeit, a szerződést megtalálja az Geckollect.com oldalon.

1.3. Szerverek

normál <https://api.geckollect.com>

Jelenleg sandbox környezetet nem biztosítunk.

Rosszindulatú, ésvagy egyéb hibákat tartalmazó kéréseket indoklás nélkül elutasítunk.

1.4. Protokoll

Általános

Az API elérése publikus HTTPS végpontokon keresztül lehetséges, tetszőleges klienskönyvtár alkalmazásával.

Az interakciók UTF-8 kódolt szöveg formájában történnek.

A kérések teste JSON objektumokat tartalmazhat, POST módszerrel, melyre a szerver is JSON formában fog válaszolni.

Figyelem! A válasz objektumban csakis a status alobjektum megléte garantált hiba esetén. Amennyiben az status.code nem API_OK állapotban van, egyetlen másik elem állapotát vagy meglétét se garantálja a szerver.

Kiegészítő információkat (hitelesítés, nyelv, verziókezelés) HTTP fejlécekben lehet megadni.

Kérésfejlécek

A következő fejlécek kell hogy kötelezően megjelenjenek minden az API szerver felé intézett kérdésben:

Content-Type	application/json
Version	v0

A szerver a következőket feltételezi a kliensről, de nem szükséges kitölteni:

Accept-Charset	utf-8
Accept	application/json

A szerver a kéréseket mindig az aktuális legfrissebb API leírás szerint próbálja értelmezni, ha nincs más beállítás.

Válasz fejlécek

Accept-Charset	utf-8
Accept	application/json
Content-Type	application/json
Allow	POST

Általános HTTP státuskódok

200 OK	sikeres kérés
307 Temporary Redirect	átirányítás másik szerverre
400 Bad Request	kérés elutasítva, kommunikáció szintű hiba
401 Unauthorized	kérés elutasítva, hitelesítés elégtelen a funkció eléréséhez
429 Too Many Requests	kérés elutasítva, letiltva
500 Internal Server Error	belső hiba miatt nem sikerült a kérés feldolgozása
503 Service Unavailable	nem kiszolgálható kérés

Státusz mező

Minden az API által feldolgozott kérés tartalmaz a válasz JSON objektum gyökerében egy **status** elemet. Ennek célja a kérés sikerességének, hibáinak jelzése. A *code* mező tartalmazza az API státuskódot, a *message* mező egy felhasználó elé tárható hibaüzenetet tartalmaz a nyelvi beállítás szerint, az *extra* pedig a fejlesztő számára közölhet hasznos információkat egy visszautasított kérés, vagy hibás működés okáról.

Az API státuskódjainak listája:

API_OK	0	Minden rendben.
API_BAD_REQUEST	1	Nem értelmezhető JSON.
API_NO_KEY	2	Nem sikerült kulcsmezőt találni a kérdésben.
API_BAD_AUTH	3	Sikertelen hitelesítés.
API_BAD_FORMAT	4	Hibásan bemeneti mező.
API_BAD_DBERROR	5	Adatbázis hiba.

2. fejezet

Interfész

2.1. Összefoglalás

A következő publikus végpontok elérhetők az internet felől:

[/data/last](#)

utolsó érték/változás lekérése

[/data/list](#)

adatlista lekérése

[/device/status](#)

eszköz állapotinformációi

A következő végpontokat tervezzük a felmerülő felhasználói igény szerint elérhetővé tenni:

[/tbd](#)

pont beállításainak lekérése

[/tbd](#)

pont beállításainak változtatása, (érték, egyéb korrekciók)

[/tbd](#)

eszköz párlista lekérés

[/tbd](#)

eszköz párlista szerkesztés

[/tbd](#)

vezérlőmondat lekérés

[/tbd](#)

vezérlőmondat szerkesztés

[/tbd](#)

azonnali kimenetállítás

Felmerülő további igények esetén keressenek minket a weboldalon található elérhetőségek egyikén.

2.2. Végpontok specifikációja

Az API szerver funkcionalitása az alább dokumentált módokon elérhető.

Figyelem! A válasz objektumban csakis a status alobjektum megléte garantált hiba esetén. Amennyiben az status.code nem API.OK állapotban van, egyetlen másik elem állapotát vagy meglétét se garantálja a szerver.

A klienseket fel kell készíteni a 300-as státuszkódcsalád követésére.

2.2.1. /data/last

Lekérdezhetővé teszi az adott ID-vel rendelkező kivezetés adatainak elérését a következő módokon:

- value: a legutolsó beérkezett (napos) értéket adja meg a program
- change: a legutolsó számítható változást adja meg a program

A működést a *type* mező irányítja.

Kérés:	POST /data/last
	<pre>{ "pointid": number, "type": string ("value" vagy "change") }</pre>
Tipikus válaszkódok:	200 OK 400 Bad Request
Válasz:	<pre>{ "pointid": number, "dataid": number, "collected": string, "received": string, "scheduled": boolean, "value": number, "status": { "code": number, "reason": "" } } vagy { "pointid": number, "received": string, "value": number, "data": [a két alkotó adat információobjektume, ld. példa], "status": { "code": number, "reason": "" } }</pre>

válaszmezők

- dataid: a beérkezett adat egyedi azonosítója
- collected: a begyűjtés időpontja
- received: a szerverre beérkezés időpontja
- scheduled: ha ütemezett adat volt, igaz, ellenkező esetekben hamis
- value: érték, vagy változás adat, általában tizedespontokkal rendelkező szám

példa - érték elérés

```
curl -i -H "Content-Type: application/json" \  
      -H "Version: v0" \  
      --data '{"pointid": 415, "type": "value", "key": "APIKEY"}' \  
https://api.geckollect.com/data/last
```

```
{  
  "pointid": 415,  
  "dataid": 2541705,  
  "collected": "2022-01-01 12:00:00",  
  "received": "2022-01-01 12:00:08",  
  "scheduled": true,  
  "value": 23.459999,  
  "status": {  
    "code": 0,  
    "reason": ""  
  }  
}
```

példa - változás elérés

```
curl -i -H "Content-Type: application/json" \  
      -H "Version: v0" \  
      --data '{"pointid": 415, "type": "change", "key": "APIKEY"}' \  
https://api.geckollect.com/data/last
```

```
{  
  "pointid": 415,  
  "received": "2022-01-01 12:00:08",  
  "value": -0.01,  
  "data": [  
    {  
      "dataid": 2541705,  
      "collected": "2022-01-01 12:00:00",  
      "received": "2022-01-01 12:00:08",  
      "scheduled": true,  
      "value": 23.459999  
    },  
    {  
      "dataid": 2541014,  
      "collected": "2022-01-01 11:00:00",  
      "received": "2022-01-01 11:00:09",  
      "scheduled": true,  
      "value": 23.469999  
    }  
  ],  
  "status": {  
    "code": 0,  
    "reason": ""  
  }  
}
```

megjegyzések

Ennél az API-nál különösen nagy a veszélye a kerettúllépésnek. Érdemes figyelni a kérések minimalizálására.

2.2.2. /data/list

Elérhetővé teszi a mérési pontra beérkezett adatok listáját. Felhasználási példák:

- export
- megjelenítés
- táblázatkezelő import forrás

Kérés:	POST /data/list
	<pre>{ "pointid": number, "from": string * "to": string * }</pre>
Tipikus válaszkódok:	200 OK 400 Bad Request
Válasz:	<pre>{ "pointid": number, "data": [{ "dataid": number, "collected": string, "received": string, "scheduled": boolean, "value": number, } ...], "status": { "code": number, "reason": "" } }</pre>

- from: a lista kezdőidőpontja *YYYY-MM-DD HH:II:SS* formátumban
- to: a lista kezdőidőpontja *YYYY-MM-DD HH:II:SS* formátumban

válaszmezők

- data: az egyes elemi adatok a `/data/last`-nál megszokott formátumban

példa - maximális adatmennyiség elérése

```
curl -i -H "Content-Type: application/json" \  
      -H "Version: v0" \  
      --data '{"pointid": 415, "key": "APIKEY"}' \  
https://api.geckollect.com/data/list
```

```
{  
  "pointid": 415,  
  "data": [  
    {  
      "dataid": 2541705,  
      "collected": "2022-01-01 12:00:00",  
      "received": "2022-01-01 12:00:08",  
      "scheduled": true,  
      "value": 23.459999  
    },  
    {  
      "dataid": 2541014,  
      "collected": "2022-01-01 11:00:00",  
      "received": "2022-01-01 11:00:09",  
      "scheduled": true,  
      "value": 23.469999  
    },  
    ....  
  ],  
  "status": {  
    "code": 0,  
    "reason": ""  
  }  
}
```

2.2.3. /device/status

Visszaadja a hardware eszköz utolsóként beérkezett telemetria információját. Javasolt felhasználás:

- állapotkövetés
- hibajelzések
- preventív javítások

A jelölt információkon kívül a szerver további metrikákat is nyújthat. Ezek elérhetősége nem garantált, hozzáférésük UB.

Kérés:	POST /device/status { "agyid": number, "radioid": number * }
Tipikus válaszkódok:	200 OK 400 Bad Request
Válasz:	{ "agyid": number, "radioid": number, "last_connection": string, "initial_version": string, "telemetry": { "timestamp": number, "sys_voltage": voltage, "sys_temperature": number, "uptime": number, "noisefloor": number, "commrssi": number, "dev_i2c": number, "dev_mbus": number, "sys_battery": number, "rf_packets": number, "rf_dropped": number, "rf_resends": number }, "status": { "code": number, "reason": "" } }

válaszmezők

- last_connection: az előző szerverkapcsolat indítás időpontja
- initial_version: a szoftververzió, amivel eredetileg gyártásra került
- timestamp: a telemetria blokk beérkezési ideje
- sys_voltage: rendszerfeszültség, absz. pontossága nem garantált, trendvonala érdekes
- sys_temperature: rendszerhőmérséklet, pontossága nem garantált

- uptime: legutóbbi újraindulás óta eltelt idő
- noisefloor: a rádiós zavartság szintje
- dev_i2c: a csatolt I2C eszközök száma
- dev_mbus: a csatolt MBUS eszközök száma
- sys_battery: ha rendelkezésre áll, a belső akkumulátor feszültsége
- rf_packets: az összes bekapcsolás óta küldött rádiós csomag száma
- rf_dropped: a véglegesen eldobott adatot tartalmazó csomagok aránya
- rf_resends: az újraküldések aránya
- commrssi: a rádiós kommunikáció vételi jelszintje

példa1 - adatgyűjtő állapotinformációi

```
curl -i -H "Content-Type: application/json" \
      -H "Version: v0" \
      --data '{"agyid": XXXXXXXXX, "key": "APIKEY"}' \
      https://api.geckollect.com/device/status
```

```
{
  "agyid": XXXXXXXXX,
  "radioid": 0,
  "last_connection": "2022-01-01 12:00:16.841404",
  "initial_version": "0",
  "telemetry": {
    "timestamp": "2022-01-01 12:00:04",
    "build": 700,
    "sys_voltage": 4.48,
    "sys_temperature": 16,
    "uptime": 66226,
    "reset": 8,
    "noisefloor": -120,
    "commrssi": 0,
    "dev_i2c": 0,
    "dev_mbus": 0,
    "net_availability": 99,
    "net_packettime": 3180,
    "info_wifi": 0,
    "info_gsm": 17,
    "info_control": 16,
    "sys_battery": 4.4,
    "pairs": 1
  },
  "connection": {},
  "status": {
    "code": 0,
    "reason": ""
  }
}
```

példa2 - rádió állapotinformációi

```
curl -i -H "Content-Type: application/json" \  
      -H "Version: v0" \  
      --data '{"agyid": XXXXXXXXX, "radioid": "YYYYYYYYY" "key": "APIKEY"}' \  
https://api.geckollect.com/device/status
```

```
{  
  "agyid": XXXXXXXXX,  
  "radioid": YYYYYYYYY,  
  "last_connection": "2022-01-01 12:00:00.259558",  
  "initial_version": "645",  
  "telemetry": {  
    "timestamp": "2022-01-01 11:59:52",  
    "build": 680,  
    "sys_voltage": 3.69,  
    "sys_temperature": 13,  
    "uptime": 1888552,  
    "reset": 3,  
    "noisefloor": -114,  
    "commrssi": -61,  
    "dev_i2c": 0,  
    "dev_mbus": 0,  
    "rf_packets": 532,  
    "rf_resends": 0,  
    "rf_dropped": 0,  
    "scaler": 98  
  },  
  "status": {  
    "code": 0,  
    "reason": ""  
  }  
}
```

3. fejezet

Támogatás

3.1. Kérdések

A rendszer használatáról további információkat talál az [Geckollect.com](https://geckollect.com) webalkalmazásban.

Amennyiben a problémája, fejlesztési ötlete van, keressen minket e-mailen, a info@geckollect.com címen.

3.2. Jogi kikötések

A geckollect.com és a Geckollect rendszer tulajdonosa a Plomba Kft.

Az Plomba Kft. fenntartja a tulajdonjogot, beleértve az összes szabadalmi jogot, szerzői jogot, védjegyet, kereskedelmi titkot és más szellemi tulajdonjogot az API-kra és annak változataira. A Plomba Kft.-nek exkluzív joga van minden szabadalmi és szerzői jogra. A felhasználó tudomásul veszi, hogy a Szerződésben foglaltak szerint nincs rendelkezik tulajdonjoggal az API-k fölött, kizárólag limitált felhasználási joggal, ezen Szerződés feltételei szerint.

Az API-kat és a dokumentációkat önmagában, bármiféle garancia nélkül szolgáltatjuk. A Plomba Kft. nem vállal garanciát arra, hogy az API-k és a dokumentációk megfelelőek a felhasználó számára vagy hibáktól mentesek. Emellett a Plomba Kft. nem garantál folyamatos és megszakítás nélküli hozzáférést az API-khoz és a dokumentációkhoz, és nem vállal garanciát a használatukból származó eredmények miatt.

A felhasználó nem teszi felelőssé a Plomba Kft.-t, annak ügyfeleit, beszállítóit vagy más partnereit és alkalmazottait az API-k használata miatt keletkezett károkért, követelésekért, ügyvédi díjakért.