

FMSservice

Omschrijving API VeiligStallen

Versie 2.6, 15-11-2018

Wijziging t.o.v. versie 2.5 zijn met **rood** aangegeven

- Bezettingsdata doorgeven op
POST /rest/v3/citycodes/0000/locations/:locationId/sections/:sectionId/occupation
- Endpoint logberichten gewijzigd
POST
<https://remote.veiligstallen.nl/rest/v3/citycodes/:citycode/locations/:locationId/sections/:sectionId/places/:placeId/logs>

Versie 2.5, 19-7-2018

Wijziging t.o.v. versie 2.4 zijn met **rood** aangegeven

- Als de parameter *depth* niet wordt meegegeven, staat deze standaard op 3

1 Base-url

`https://remote.veiligstallen.nl/rest/v3`

2 Gebruikte datatypen

Datumtijd-properties, zoals `transactiondate`, `checkindate`, `checkoutdate`, `expirationdate` hebben altijd type `Timestamp`

Timestamp toegestane formats:

`yyyy-mm-dd hh:mm:ss`

`2017-03-18T14:45:27.000Z` (ISO 8601 time)

Property `biketypeid` (integer):

- 1: normale fiets
- 2: bromfiets/scooter
- 3: speciale fiets
- 4: elektrische fiets
- 5: motorfiets
- 6: mindervalidenfiets

Property `idtype` (integer):

- 0: barcode (sleutelhanger) (12 karakters; codes uitgegeven door CROW)
- 1: ov-chipkaart (uitgelezen nummer; juiste format op te vragen bij CROW)
- 2: cijfercode (9 cijfers; nummers uitgegeven door derden in overleg met CROW)
- 3: tijdelijk code ov-chipkaart (6 cijfers, zonder #; code gegenereerd door VeiligStallen)
- 4: tijdelijke code barcode (sleutelhanger) (code gegenereerd door VeiligStallen)

Property `typecheck` (string): `user|controle|reservation`

Property `type` (string): `in|out`

Property `paymenttypeid` (string):

- 1: klant heeft betaald
- 2: kwijtschelding, klant heeft niet betaald

Property locationtypeid (integer)

- 1: bewaakte stalling
- 2: stalling met toezicht
- 3: geautomatiseerde stalling
- 4: onbewaakte stalling
- 5: fietskluizen
- 6: buurtstalling
- 7: fietstrommel

Property messagecode (integer)

- 1: buiten gebruik
- 2: ID in gebruik voor andere plek
- 3: stallingstegoed veel te laag
- 4: stallingstegoed te laag
- 5: plek is in gebruik door iemand anders
- 6: Ongeldige ID-code
- 7: BeheerdersID

Property statuscode (integer)

- 0: vrij
- 1: bezet
- 2: abonnement
- 3: gereserveerd
- 4: buiten werking

Query-params:

citycode: 4-cijfers, doorgaans de hoofdpostcode van een gemeente

locationid: citycode_3-cijferig volgnummer: 1234_001

section: citycode_3-cijferig volgnummer_volgnummer: 1234_001_1

placeid: integer

3 Locatie informatie

3.1 Toelichting

CROW-Fietsberaad biedt alle statische informatie over de stallingen plus informatie over de actuele bezetting aan als open data. De informatie kan worden opgevraagd op gemeente- of locatieniveau.

De data zijn vrij te gebruiken, maar we stellen het zeer op prijs hiervan op de hoogte te worden gesteld. Dit kan via fietsberaad@crow.nl onder vermelding van "Gebruik Open Data VeiligStallen.nl". CROW Fietsberaad noch de gemeenten zijn aansprakelijk voor de juistheid van de gegevens.

Naast het opvragen van data kunnen stallingssystemen middels een POST data doorgeven via de FMSService (zie §3.3).

3.2 Request

Opvragen informatie over een gemeente + alle stallingslocaties

```
GET /citycodes/:citycode/
```

Response:

```
{
  citycode:"1234",
  name:"Ons Dorp",
  [
    Location,
    Location
  ]
}
```

Opvragen informatie alle stallingslocaties van een gemeente:

```
GET /citycodes/:citycode/locations/
```

Response:

```
[
  Location,
  Location,
  ...
]
```

Opvragen informatie van een bepaalde locatie

```
GET /citycodes/:citycode/locations/:locationid
```

Opvragen informatie van een specifieke sectie

```
GET /citycodes/:citycode/locations/:locationid/sections/:sectionid
```

Opvragen informatie van alle plekken van een sectie

GET /citycodes/:citycode/locations/:locationid/sections/:sectionid/places/

Depth

Om de informatiestroom te beperken kan er de parameter *depth* worden meegegeven aan GET-requests. Op deze manier is het mogelijk om bijvoorbeeld alleen de gegevens van een stalling op te vragen, zonder details over de secties en de plekken.

Er worden 3 levels onderscheiden:

1. Locaties
2. Secties
3. Plekken

GET /citycodes/:citycode/locations/?depth=1

geeft alleen de locaties, zonder secties.

GET /citycodes/:citycode/locations/? depth=2

geeft de locaties + secties.

GET /citycodes/:citycode/locations/:locationid/?depth=1

geeft alleen de locatie, zonder secties.

GET /citycodes/:citycode/locations/:locationid/?depth=2

geeft de locatie + secties.

GET /citycodes/:citycode/locations/:locationid/?depth=3

geeft de locatie + secties + plekken

GET /citycodes/:citycode/locations/:locationid/sections/:sectionid?depth=1

geeft de sectie

GET /citycodes/:citycode/locations/:locationid/sections/:sectionid?depth=2

geeft de sectie + plekken

Als de parameter *depth* niet wordt meegegeven, staat deze standaard op 3

Response:

```
[  
  Place,  
  Place,  
  ...  
]
```

Opvragen informatie over een plek

```
GET /citycodes/:citycode/locations/:locationid/sections/:sectionid/places/:placeid
```

Opvragen beperkt aantal properties

Gebruik de parameter **fields** voor het opvragen van een beperkt aantal properties:

```
GET  
/citycodes/:citycode/locations/:locationid?fields=location.capacity,location.free,location.occupied
```

Doorgeven informatie over een specifieke locatie (implementatie op aanvraag)

```
POST /citycodes/:citycode/locations/:locationid
```

Opbouw body idem aan response bij GET-request.

Response:

```
{  
  "message": "Ok",  
  "status": 1  
}
```

In geval van error

```
{  
  "message": "{reden van error}",  
  "status": 0  
}
```

Doorgeven informatie over een specifieke sectie

```
POST /citycodes/:citycode/locations/:locationid/sections/:sectionid
```

Opbouw body idem aan response bij GET-request.

Response:

```
{
  "message": "Ok",
  "status": 1
}
```

In geval van error

```
{
  "message": "{reden van error}",
  "status": 0
}
```

Doorgeven informatie over een plek

PUT /citycodes/:citycode/locations/:locationid/sections/:sectionid/places/:placeid

```
{
  "datelaststatusupdate":"2017-03-18 14:45:27",
  "name":"kluis 1",
  "statuscode":0|1|2|3|4,
  "urlwebservice": "http://www.url-stallingssysteem-voor-alert-statuswijziging.nl", (alleen POST)
}
```

Logberichten doorgeven

POST
/citycodes/:citycode/locations/:locationid/sections/:sectionid/places/:placeid/logs

Body:

```
{
  "action":"Open"|"Deur ontgrendeld"|"Deur vergrendeld"|"Code ingegeven"|"Pas
aangeboden"|"etc",
  "type":"debug|info|warning|error|critical",
  "timestamp":"2017-07-11T03:25:28",
  "description": "Vrije tekst", (alleen POST)
}
```

Response:

```
{
  "message": "Ok",
  "status": 1
}
```

Bezettingsdata doorgeven:

POST /rest/v3/citycodes/0000/locations/:locationId/sections/:sectionId/occupation

Body:

```
{
  "timestamp":"2017-08-28T13:45:00", // default = nu
  "occupation": 400,
  "checkins":1,
  "checkouts":3,
  "intervalinminutes":15, // verplicht indien checkins of checkout gegeven is
  "bikes":[
    {
      "idcode":"001234567890",
      "idtype": 1,
      "bikeid":"987654321",
      "transactiondate":"2018-06-15T14:45:27"
    },
    {...},
    {...},
    ...
  ]
}
```

3.3 Toelichting request

Alle properties worden teruggegeven bij een GET-request tenzij anders is aangegeven met “(alleen POST)”.

Tarieven zijn ingedeeld in kostenperiodes. Een stalling die de eerste 24 uur gratis en en daarna €0.50 per 24 uur kost heeft als kostenperiodes: {timespan: 24, cost: 0.00},{timespan:25, cost:0.50}. De laatste kostenperiode wordt doorlopend herhaald.

Als er geen close is gegeven dan is de stalling altijd geopend. Open heeft in dat geval waarden 0 en 0000. Er is slechts één tijdsperiode.

Het opvragen van de statische gegevens van alle stallingen duurt nog tamelijk lang. Deze gegevens wijzigen echter niet vaak. Geadviseerd wordt om de gegevens van heel Nederland daarom maximaal

eens per week op te vragen en de data lokaal op te slaan. Is een lokale opslag niet mogelijk dan wordt geadviseerd minder velden op te vragen. Dit gaat aanmerkelijk sneller dan het opvragen van alle statische gegevens van alle stallingen.

Met betrekking tot het opvragen van de actuele bezetting is het advies deze data minimaal eens per minuut op te vragen met een maximum van eens per vijf minuten.

POST-requests kunnen gebruikt worden om activiteiten bij de stalling op plaats-niveau door te geven of direct de bezetting en het aantal in- en uitchecks door te geven.

De activiteiten (bv deur geopend, pas aangeboden, deur ontgrendeld) worden binnen het FMS gelogd met als doel bij issues de exploitant snel op de hoogte te kunnen stellen. Momenteel wordt deze functionaliteit gebruikt voor E-kluizen, maar de functionaliteit kan ook worden gebruikt voor andere stallingssystemen op plek-niveau. De properties "type" en "timestamp" zijn verplicht, "action", "idtype", "idcode" en "description" zijn optioneel. "Action" is in principe een vrije tekstveld, maar het is handig met CROW af te stemmen wat in welke situatie wordt doorgegeven om zoveel mogelijk consistent te blijven tussen verschillende stallingssystemen.

In plaats van het doorgeven van alle stallingstransacties in een stalling (zie §4) op basis waarvan het FMS de actuele bezetting en aantal in- en uitchecks bepaalt kunnen stallingssystemen ook direct de actuele bezetting of het aantal in- en uitchecks in een tijdsinterval doorgeven. Voor het doorgeven van de bezetting zijn de properties "occupation" en "timestamp" verplicht. Voor het doorgeven van het aantal in- en uitchecks zijn de properties "checkins", "checkouts", "timestamp" en "intervalinminutes" verplicht. "Timestamp" is het tijdstip van het einde van de periode waarin de transacties hebben plaatsgevonden. Het minimale interval waarop de rapportages in het FMS zijn gebaseerd is 15 minuten. Kortere intervallen doorgeven is toegestaan maar weinig nuttig.

3.4 Uitbreidingsmogelijkheden

Om de webservice in de toekomst te verbeteren wordt momenteel gedacht aan:

- Het kunnen opvragen van wijzigingen vanaf tijdstip x
- Het kunnen opvragen van alle stallingen in een omtrek van x kilometer ten opzichte van locatie y
- Het kunnen opvragen van alleen stallingen van een bepaald type
- Het kunnen doorgeven van stallingseigenschappen als capaciteit, openingstijden, tarieven etc zodat deze eigenschappen ook op andere plaatsen dan in het FMS gewijzigd kunnen worden

Graag horen we uw mening over deze mogelijk verbeteringen. Voldoen de bestaande ideeën aan uw vraag? Nieuwe suggesties zijn ook van harte welkom (fietsberaad@crow.nl).

4 Transacties & betalingen

4.1 Toelichting

Stallingssystemen geven iedere in- en uitcheck door aan het FMS. Deze stallingstransacties zijn minimaal verbonden aan een sectie. Als er kosten aan een transactie zijn verbonden wordt dit ook doorgegeven. De eventuele betaling kan in hetzelfde request worden meegegeven of in een apart request. Betalingen zijn dus niet een op een verbonden met een stallingstransactie. Het kan ook een algemene opwaardering van het stallingstegoed betreffen. Betalingen zijn minimaal aan een stalling verbonden.

4.2 Request

Voor een stallingstransactie en/of betaling van een plek en/of koppeling ID's, bijvoorbeeld een fietskluis:

```
POST
/citycodes/:citycode/locations/:locationid/sections/:sectionid/places/:placeid/transac
tions
```

Voor een stallingstransactie en/of betaling van een willekeurige plek in een sectie van een stalling en/of koppeling ID's:

```
POST /citycodes/:citycode/locations/:locationid/sections/:sectionid/transactions
```

Voor een betaling in de stalling en/of koppeling ID's:

```
POST /citycodes/:citycode/locations/:locationid
```

Beveiliging: Basic-authentication

Body:

```
{
  "price":"1.00",
  "idcode":"001234567890",
  "idtype": 0|1|2,
  "bikeid":"987654321",
  "transactiondate":"2017-03-18T14:45:27",
  "typecheck":"user|controle|reservation",
  "type":"in|out",
  "clienttypeid": 1|2,
  "amountpaid": "1.00"
  "paymenttypeid":1,
}
```

Response:

```
{
  "message": "Ok",
  "status": 1
}
```

```
}
```

In geval van error

```
{  
  "message": "{reden van error}",  
  "status": 0  
}
```

4.3 Toelichting request:

Afhankelijk van het doel worden velden wel of niet meegegeven in de body:

Doorgeven stallingstransactie inclusief betaling

Verplicht: price, idcode, idtype, transactiondate, typecheck, type, amountpaid

Optioneel: bikeid, clienttypeid (default: 1), paymenttypeid (default:1)

Doorgeven stallingstransactie exclusief betaling

Verplicht: price, idcode, idtype, transactiondate, typecheck, type

Optioneel: bikeid, clienttypeid (default: 1),

Niet vullen: amountpaid, paymenttypeid

Doorgeven betaling

Verplicht: idcode, idtype, transactiondate, amountpaid, paymenttypeid

Niet vullen: price, typecheck, type, bikeid, clienttypeid

Doorgeven koppeling ID fiets en ID klant

Verplicht: idcode, idtype, bikeid, transactiondate

Niet vullen: price, amountpaid, paymenttypeid, typecheck, type, clienttypeid

Als een ID van de fiets wordt meegegeven wordt deze door het FMS gekoppeld aan het ID van de klant. Wordt geen ID fiets meegegeven dan gebruikt het FMS het klant ID dat het laatst is gebruikt bij de betreffende fiets. Is er niet eerder een koppeling gemaakt dan geldt het klant ID ook als fiets ID.

Gestalde fietsen synchroniseren met server veiligstallen

4.4 Toelichting

Er kunnen in de loop van de tijd kleine verschillen ontstaan tussen de werkelijkheid in de stalling en de stand van zaken in het FMS. Om dit recht te trekken kunnen stallingssystemen na een opschooncontrole de fietsen die zijn aangetroffen in de stalling doorgeven. Op basis hiervan vindt een correctie plaats in het FMS.

4.5 Request

POST /citycodes/:citycode/locations/:locationid/sections/:sectioned/occupation

Beveiliging: Basic-authentication

Body:

```
{
  Optioneel: "transactiondate": "2017-03-15 14:45:27", // datumtijd van de controle, indien niet
  gegeven, dan wordt de tijd van versturen aangehouden
  [
    {
      "idcode": "001234567890",
      "idtype": 0|1|2,
      "bikeid": "987654321",
      "transactiondate": "2017-03-15T14:45:27" // => datetime checkin
    },
    ...
  ]
}
```

Response:

```
{
  "message": "Ok",
  "status": 1
}
```

In geval van error

```
{
  "message": "{reden van error}",
  "status": 0
}
```

4.6 Toelichting request

Verplichte velden: idcode, idtype, bikeid, transactiondate

5 Abonnementvormen opvragen

5.1 Toelichting

Stallingsystemen kunnen per stalling opvragen welke abonnementvormen zijn toegestaan. Deze informatie kan gebruikt worden voor tekstuele toelichtingen op de scanapparatuur of voor de verkoop van abonnementen in de stalling (deze worden daarna doorgegeven aan het FMS met een referentie naar de abonnementvorm).

5.2 Request

```
GET /citycodes/:citycode/locations/:locationid/subscriptiontypes
```

Beveiliging: Basic-authentication

Response:

```
[
  {
    "subscriptiontypeid": 1,
    "subscriptiontypename": "Jaarabonnement Fiets",
    "subscriptiontypecost": "65.60",
    "biketypes": [1,2],
    "idtypes": [0,1],
    "duration": 12,
    "bikeparktypeid":"bewaakt"
  },
  ...
]
```

6 **Abonnement doorgeven** [nog niet online, implementatie op verzoek]

6.1 **Toelichting**

Abonnementen kunnen verkocht worden via VeiligStallen.nl maar ook via apps van derden of in de stalling zelf. De abonnementen worden daarna doorgegeven aan het FMS zodat het abonnement herkend wordt in alle stallingen waarvoor het abonnement geldt.

6.2 **Request**

POST /citycodes/:citycode/locations/:locationid/subscriptions

Beveiliging: Basic-authentication

Body:

```
{
  "cost": "65.60"
  "startdate": "2017-03-15T14:45:27",
  "expirationdate": "2018-03-15T14:45:27",
  "idcode": "001234567890",
  "idtype": "0|1|2",
  "subscriptiontypeid": 5
}
```

Response:

```
{
  "message": "Ok",
  "status": 1
}
```

In geval van error

```
{
  "message": "{reden van error}",
  "status": 0
}
```

6.3 **Toelichting request**

Verplichte velden: startdate, idcode, idtype, subscriptiontypeid

Optionele velden: cost (default: subscriptioncost), expirationdate (default: startdate + duration)

Het FMS gaat ervan uit dat de kosten daadwerkelijk door de klant zijn betaald aan de verkoper van het abonnement.

7 Abonnementhouders opvragen

7.1 Toelichting

Om abonnementhouders ook te kunnen herkennen in de stalling als er geen dataverbinding is tussen het stallingssysteem en het FMS kunnen stallingssystemen voor een lokale opslag een lijst opvragen van alle abonnementen die gelden in een stalling.

7.2 Request

GET /citycodes/:citycode/locations/:locationid/subscriptions

Beveiliging: Basic-authentication

Response:

```
[
  {
    "idcode":"001234567890",
    "idtype": 0|1|2|3|4,
    "subscriptiontypeid": 5,
    "expirationdate": "2018-03-19T15:45:27"
    "placeid":123
  },
  ...
]
```

7.3 Toelichting request

Abonnementen kunnen zijn gekoppeld aan een stalling of aan een specifieke plek in een stalling. De property "placeid" wordt alleen in het laatste geval meegegeven in de response.

Tijdelijk code vervangen door definitief ID

7.4 Toelichting

Als een staller bij het afsluiten van een abonnement of het maken van een reservering nog geen definitieve pas heeft genereert het FMS een tijdelijke code die in/bij de stalling kan worden omgezet naar het definitieve ID.

7.5 Request

```
POST
/citycodes/:citycode/locations/:locationid/sections/:sectionid/places/:placeid/idcodes/:idtype/:idcode
```

Of

```
POST /citycodes/:citycode/locations/:locationid/idcodes/:idtype/:idcode
```

Body:

```
{
  "idcode": "001234567890",
  "idtype": 0|1|2
}
```

Response:

```
{
  "message": "Ok",
  "status": 1
}
```

In geval van error

```
{
  "message": "{reden van error}",
  "status": 0
}
```

7.6 Toelichting request

In de URL wordt de tijdelijke code meegegeven. In het geval van een tijdelijke OV-chipkaartcode (3) wordt de code zonder “#” meegegeven.

Idtype 3 kan alleen worden gewijzigd in idtype 1 en idtype 4 alleen in idtype 0. Andere combinaties geven een error.

8 Check of ID klant plek mag gebruiken

8.1 Toelichting

Abonnementen en reserveringen kunnen gekoppeld zijn aan een specifieke plek. In dit geval kan een stalling op het moment dat de klant zijn ID aanbiedt aan het systeem aan het FMS vragen of het betreffende ID gebruikt mag worden voor een specifieke plek. Momenteel is deze functionaliteit vooral geschikt voor fietskluizen, maar uitbreiding naar andere stallingen die gebruik maken van placeID's is mogelijk.

8.2 Request

```
GET
citycodes/:citycode/locations/:locationid/sections/:sectionid/places/:placeid/idcodes/:idtype/:idcode
```

Response:

```
{
  "allowed": true|false,
  "messagecode": 1,
  "idcode": "0123456789",
  "idtype": 1
}
```

9 Servertijd opvragen

9.1 Request

GET /servertime

Response:

2017-03-15T14:45:27