

# FMService

## Omschrijving API VeiligStallen

Versie 3.0.4, 8 oktober 2020

## Inhoudsopgave

<b>1</b>	<b>Basisgegevens</b>	<b>4</b>
1.1	Base-url	4
1.2	Gebruikte datatypen	4
<b>2</b>	<b>Rechten</b>	<b>6</b>
<b>3</b>	<b>Locatie-informatie</b>	<b>6</b>
3.1	Toelichting	6
3.2	Opvragen locaties	6
3.1	Dieper doorvragen: de parameter <i>Depth</i>	8
3.2	Snijden in de gegeven informatie: de parameter <i>Fields</i>	9
3.3	Doorgeven informatie over een specifieke locatie (implementatie op aanvraag)	9
3.4	Toelichting request	12
3.5	Belangrijke noten	12
3.6	Uitbreidingsmogelijkheden	13
<b>4</b>	<b>Transacties &amp; betalingen</b>	<b>14</b>
4.1	Request	14
4.2	Toelichting request	15
4.3	Toelichting	16
4.4	Synchronisatie gestalde fietsen	16
4.5	Toelichting request	16
4.6	Volledige transacties doorgeven	16
<b>5</b>	<b>Abonnementsvormen opvragen</b>	<b>17</b>
5.1	Toelichting	17
5.2	Request	17
<b>6</b>	<b>Abonnementen</b>	<b>18</b>
6.1	Toelichting	18
6.2	Request	18
6.3	Toelichting request	18
6.4	Tijdelijk code vervangen door definitief ID	19

6.5 Check of ID klant plek mag gebruiken	20
6.6 Nieuwe abonnement	20
6.7 Update bestaand abonnement	21
<b>7 Extra functies</b>	<b>22</b>
7.1 Server time	22
7.2 Fietstypen	22
7.3 Paymenttypes	23
7.4 Gebruikte fietsen	23
7.5 Stallingstegoeden opvragen	24

## Inleiding

Het Fietsparkeer Management Systeem (FMS) van VeiligStallen verzamelt en verstrekt data over fietsparkeren. Centraal in dit dataverkeer is de FMS-Service. Deze REST-API kan data ontvangen van bijvoorbeeld fietsenstallingen en ook data verstrekken aan externe webapplicaties of voor analysedoeleinden.

Dit document beschrijft de diverse endpoints die de FMS-Service ter beschikking stelt.

Een groot deel van de data is open, d.w.z. voor iedereen toegankelijk. Voor privacy- en bedrijfsgevoelige data geldt dat uiteraard niet. Deze is uitsluitend beschikbaar voor de rechthebbenden. Denk hierbij aan data van kluizengebruikers, houders van abonnementen voor bewaakte stallingen en stallingstegoeden.

Daarnaast zijn uiteraard ook de endpoints om data naar het FMS te sturen, beschermd.

Het hoofdstuk 'Rechten' (hoofdstuk 2) gaat in meer detail in op de rechtenstructuur.

## 1 Basisgegevens

### 1.1 Base-url

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

### 1.2 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

## 2 Rechten

Gebruikers van de afgeschermd functies van de FMS-Service worden daartoe door CROW-Fietsberaad gemachtigd. Zij ontvangen daarbij een username en een wachtwoord, alsmede een per gemeente en per stalling ingestelde rol.

Voor de open data zijn geen credentials vereist.

Username en wachtwoord dienen gebruikt te worden om zich bij de API middels **basic-http-authentication** te identificeren.

De rol bepaalt voor welke functies de gebruiker is gemachtigd. Er bestaan drie rollen:

1. **Dataleverancier#1**: mag data insturen en lezen over stallingesevent (het in- en uitchecken van gebruikers in bewaakte stallingen)
2. **Dataleverancier#2**: mag bezettingsdata insturen
3. **Operator**: beschikt over alle beschermde lees- en schrijfrechten

Bij iedere onder beschreven functie staat vermeld, met welke rol het recht geeft tot het gebruik ervan. Indien dit niet het geval is, is voor deze functie geen authenticatie vereist.

## 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](mailto: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 §3.4).

### 3.2 Opvragen locaties

**Opvragen informatie over een gemeente + alle stallingslocaties**

GET /citycodes/:citycode/

Response:

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

Bijvoorbeeld:

<https://remote.veiligstallen.nl/rest/v3/citycodes/3500>

Geeft informatie over de gemeente Utrecht + alle stallingslocaties in deze gemeente

**Opvragen informatie alle stallingslocaties van een gemeente:**

GET /citycodes/:citycode/locations/

Response:

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

Bijvoorbeeld:

<https://remote.veiligstallen.nl/rest/v3/citycodes/3500/locations>

Geeft informatie over alle stallingslocaties in de gemeente Utrecht

**Opvragen informatie van een bepaalde locatie**

GET /citycodes/:citycode/locations/:locationid

Bijvoorbeeld:

[https://remote.veiligstallen.nl/rest/v3/citycodes/3500/locations/3500\\_004](https://remote.veiligstallen.nl/rest/v3/citycodes/3500/locations/3500_004)

Geeft informatie over de locatie met id 3500\_004 (Utrecht Stadhuis)

**Opvragen informatie van een specifieke sectie**

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

Bijvoorbeeld:

[https://remote.veiligstallen.nl/rest/v3/citycodes/5700/locations/5700\\_004/sections/5700\\_004\\_1](https://remote.veiligstallen.nl/rest/v3/citycodes/5700/locations/5700_004/sections/5700_004_1)

Geeft informatie over sectie #1 van de locatie met id 5700\_004 (Stationsstalling, Helmond)

## Opvragen informatie van alle plekken van een sectie

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

Bijvoorbeeld:

[https://remote.veiligstallen.nl/rest/v3/citycodes/3708/locations/3708\\_003/sections/3708\\_003\\_1/places](https://remote.veiligstallen.nl/rest/v3/citycodes/3708/locations/3708_003/sections/3708_003_1/places)

Geeft informatie de fietskluizen van sectie #1 – dit is overigens de enige sectie van deze locatie - van de locatie met id 3708\_003 (Fietskluizen Jordanlaan, Zeist)

### 3.1 Dieper doorvragen: de parameter *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.

Let op: te diep doorvragen vanaf een te hoog niveau (bijvoorbeeld alle gemeentes tot op plekniveau uitvragen) gaat ten koste van de performance!

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

### 3.2 Snijden in de gegeven informatie: de parameter *Fields*

Gebruik de parameter **fields** voor het opvragen van een beperkt aantal property's:

GET /citycodes/:citycode/locations/:locationid?fields=location.occupation

Geeft bijvoorbeeld enkel de bezettingsdata van een stalling

Gebruik van de parameter Fields verhoogt de performance van de requests!

### 3.3 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
of
POST /citycodes/:citycode/locations/:locationid/sections/:sectionid/places/:placeid
{
  "statuscode": 0|1|2|3|4,
  "transactiondate": "2017-07-11T03:25:28",
  "name": "kluis 1",
  "urlwebservice": "http://www.url-stallingssysteem-voor-alert-statuswijziging.nl"
}
```

Beveiliging: Basic-http-authentication

Rechthebbenden: operator

Alle velden zijn optioneel

Response:

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

#### Kluisacties doorgeven

Ter ondersteuning van de kluizen kan een kluis een bepaalde actie doorgeven die de kluis zojuist heeft uitgevoerd. De kluis kan bijvoorbeeld doorgeven als de deur is geopend.

Aan de hand van de actieid's wordt de status van de kluizen door het fms gemonitord. Het fms stuurt een e-mail aan de exploitant als de acties van de kluis niet corresponderen met de doorgegeven statussen en transacties.

Voor meer info over fietskluizen binnen het FMS, zie document *E-kluizen/Gebruikershandleiding Kluisbeheer*.

```
POST /citycodes/0000/locations/0000_001/sections/0000_001_1/places/111/actions
```

Body:

```
{
  "actionid": 11, // optioneel
  "type": "debug|info|notice|warning|error|critical", // in geval van een actionid
  dient type 'info' gebruikt te worden
  "timestamp": "2017-07-11T03:25:28",
  "idtype": 1, // optioneel
  "idcode": "12345", // optioneel
  "description": "Vrije tekst"
}
```

Is een actieid gegeven, kies dan voor type='info'. Dit type activeert de validaties binnen het fms die behoren bij het gegeven actie.

Mogelijk actieid's:

```
21 Deur geopend
22 Deur gesloten
31 Pas aangeboden
32 Code ingegeven
41 Slot ontgrendeld
42 Slot vergrendeld
51 Lamp groen
52 Lamp oranje
53 Lamp rood
62 Geen fiets gedetecteerd
61 Fiets gedetecteerd
```

Een kluis kan ook eenvoudigweg logberichten sturen, bijvoorbeeld om het voor de beheerder mogelijk te maken de kluis te monitoren. In deze gevallen verdient het de voorkeur type 'debug' mee te geven.

Een speciaal geval is type '**notice**'. Hiermee geeft de kluis aan dat de verbinding tussen kluis en de server van Veiligstallen nog okee is. Dit type kan zonder enige andere parameter worden doorgegeven.

Let op: notices dienen tenminste elk half uur te worden gegeven.

Als een kluis tenminste eenmaal een 'notice' heeft gestuurd, controleert de Veiligstallen-server ieder half uur of er een notice is ontvangen. Als dit niet het geval is, is de verbinding mogelijk weggefallen. Nadat deze nogmaals is gecontroleerd, zal de exploitant hierover een e-mail ontvangen.

### Bezettingsdata doorgeven:

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

Beveiliging: Basic-http-authentication

Rechthebbenden: operator, dataleverancier#2

Body:

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

### 3.4 Toelichting request

### 3.5 Belangrijke noten

**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:24, cost:0.50}. De laatste kostenperiode wordt doorlopend herhaald.

**Openingstijden:** Als er geen close is gegeven dan is de stalling altijd geopend. Open heeft in dat geval waarden tussen 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.

Maak bij requests die veel locaties omvatten gebruik van de parameters `depth` en `fields` om de respons op maat te snijden. Dit zal de performance ten goede komen!

De **kluisactiviteiten** (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-kluisen, 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.

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.

Let op: de actuele bezetting van sommige stallingen wordt bepaald aan de hand van in- en uitchecks. Het gaat hier om de locaties met `occupationsource: “FMS”`. Omdat het regelmatig voorkomt dat gebruikers niet in- of uitchecken, wordt de bezettingsdata bij benadering gegeven. De data wordt na een stallingscontrole, die meestal wekelijks plaatsvindt, gecorrigeerd. Deze gecorrigeerde data wordt niet aangeboden als open data, maar is alleen beschikbaar voor rechthebbenden.

In plaats van het doorgeven van alles stallingstransacties in een stalling (zie hoofdstuk 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.

### 3.6 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

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.

### 4.1 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/transactions
```

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-http-authentication

Rechthebbenden: operator, dataleverancier#1

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
}
```

#### 4.2 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

### 4.3 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.

### 4.4 Synchronisatie gestalde fietsen

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

Beveiliging: Basic-http-authentication

Rechthebbenden: operator, dataleverancier#2

Body:

```
{
  "transactiondate": "2019-11-05T00:03:04+02:00",
  "bikes": [
    {
      "idcode": "001234567890",
      "idtype": 1,
      "bikeid": "987654321",
      "transactiondate": "2018-06-15T14:45:27"
    },
    { ... },
    { ... }
  ]
}
```

Response:

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

In geval van error

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

### 4.5 Toelichting request

Verplichte velden: idcode, idtype, bikeid, transactiondate

### 4.6 Volledige transacties doorgeven

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



Beveiliging: Basic-http-authentication

Rechthebbenden: operator, dataleverancier#2

Body:

```
{
  "checkindate": "2017-07-25 9:00:02",
  "checkoutdate": "2017-07-25 13:00:02",
  "typecheckin": "user",
  "typecheckout": "controle"
}
```

Let op: op basis van volledige transacties berekent het FMS **niet** automatisch bezettingsdata!

## 5 Abonnementsvormen opvragen

### 5.1 Toelichting

Stallingsystemen kunnen per stalling opvragen welke abonnementsvormen 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 abonnementsvorm).

### 5.2 Request

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

Beveiliging: Basic-http-authentication

Rechthebbenden: geen restricties

Response:

```
[
  {
    "id": 1,
    "name": "Jaarabonnement Fiets",
    "biketypes": [1,2],
    "idtypes": [0,1],
    "duration": 12,
    "price": 55.6,
    "locationtype": "bewaakt"
  },
  ...
]
```

## 6 Abonnementen

### 6.1 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.

### 6.2 Request

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

Beveiliging: Basic-http-authentication

Rechthebbenden: operator

Response:

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

### 6.3 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.

#### 6.4 Tijdelijk code vervangen door definitief ID

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.

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

Of

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

Beveiliging: Basic-http-authentication

Rechthebbenden: operator

Body:

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

Response:

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

In geval van error

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

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.

## 6.5 Check of ID klant plek mag gebruiken

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.

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

Beveiliging: Basic-http-authentication

Rechthebbers: operator

Response:

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

## 6.6 Nieuwe abonnement

Abonnementen die via een kanaal van derden verkocht worden doorgegeven aan het FMS (par. 9.2). Als er wijzigingen plaatsvinden worden deze ook doorgegeven, bijvoorbeeld bij restitutie of een vroegtijdige beëindiging van het abonnement (par. 9.3)

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

Beveiliging: Basic-authentication

Rechthebbers: operator

Body:

```
{
  "price": 65.60 // optioneel. Indien niet gegeven wordt de prijs bepaald aan de hand van het subscriptiontypeid
  "startdate": "2020-11-16T00:00:00",
  "expirationdate": "2018-03-15T14:45:27", // optioneel. Indien niet gegeven wordt de aflooptdatum bepaald aan de hand van het subscriptiontypeid
  "idcode": "001234567890",
  "idtype": 0,
  "subscriptiontypeid": 5
}
```

Response:

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

```

    "subscription": {
      "idtype": 0,
      "startdate": "2020-11-16T00:00:00",
      "subscriptiontypeid": 5,
      "expirationdate": "2021-11-16T00:00:00",
      "idcode": "001234567890",
      "price": 65.60
    }
  }
}

```

#### In geval van error

```

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

```

Verplichte velden: startdate, idcode, idtype, subscriptiontypeid

Optionele velden: cost (default: subscriptioncost), expirationdate (default: startdate + abonnementsduur volgens subscriptionid)

Verplichte velden: startdate, idcode, idtype, subscriptiontypeid

Optionele velden: price (default: prijs van de abonnementsvorm), expirationdate (default: startdate + duur volgens abonnementsvorm)

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

## 6.7 Update bestaand abonnement

PUT /citycodes/:citycode/locations/:locationid/subscriptions/:subscriptionid

Beveiliging: Basic-authentication

Rechthebbers: operator

Body:

```

{
  "cost": "-10.00"
  "startdate": "2017-03-15T14:45:27",
  "expirationdate": "2018-01-15T14:45:27",
  "idcode": "001234567890",
  "idtype": 0|1|2,
}

```

Response:

```

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

```

Als "cost" positief is betekent dit dat de Staller dit bedrag heeft betaald aan de Exploitant voor het abonnement; een negatief getal betekent dat dit bedrag door de Exploitant is terugbetaald aan de Staller.

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

## 7 Extra functies

### 7.1 Server time

GET /servertime

Response:

2017-03-15T14:45:27

### 7.2 Fietstypen

GET /biketypes

Response

```
[
  {
    "name": "Fietsen",
    "id": 1,
    "singular": "Fiets"
  },
  {
    "name": "Bromfietsen",
    "id": 2,
    "singular": "Bromfiets"
  },
  {
    "name": "Speciale fietsen",
    "id": 3,
    "singular": "Speciale fiets"
  },
  {
    "name": "Elektrische fietsen",
    "id": 4,
    "singular": "Elektrische fiets"
  },
  {
    "name": "Motorfietsen",
    "id": 5,
    "singular": "Motorfiets"
  },
  {
    "name": "Mindervalidenfietsen",
    "id": 6,
    "singular": "Mindervalidenfiets"
  }
]
```

### 7.3 Paymenttypes

GET /paymenttypes

#### Response

```
[
  {
    "paymenttypeid": 1,
    "name": "Fysieke betaling/restitutie",
    "description": "Klant heeft betaald in de stalling. Bij een negatief bedrag
betekent deze code dat de klant geld heeft teruggekregen van de beheerder
(restitutie)"
  },
  {
    "paymenttypeid": 2,
    "name": "Kwijtschelding",
    "description": "Beheerder heeft stallingsschuld kwijtgescholden. Louter
administratieve handeling. Geen fysiek geld betrokken."
  }
]
```

### 7.4 Gebruikte fietsen

Stallingen kunnen opvragen welke fietsen bij andere stallingen in dezelfde gemeente gebruikt zijn.

Mogelijke toepassing: aan sleutelhangers gekoppelde fietsbarcodes, die in andere stallingen gekoppeld zijn, opvragen en op lokale server opslaan.

Alleen updates van niet langer dan 1 week geleden kunnen worden opgevraagd.

Rechthebbers: operator

GET /citycodes/:citycode/locations/:locationid/bikeupdates?from=2019-11-13T11:00:00

#### Response

```
{
  "data": [
    {
      "datemodified": "2019-10-30T18:12:34",
      "idtype": 1,
      "status": "in",
      "barcodebike": "4321",
      "biketypeid": 1,
      "idcode": "1233211111"
    },
    {
      "datemodified": "2019-10-30T18:17:15",
      "idtype": 1,
      "status": "out",
      "barcodebike": "1234",
      "biketypeid": 1,
      "idcode": "1234",
      "balance": -1.5 // alleen gegeven als stallingstegoed ongelijk is aan 0
    }
  ]
}
```

```

    }
  ],
  "citycode": "0000",
  "from": "2019-11-13T11:00:00"
}

```

## 7.5 Stallingstegoeden opvragen

Een stalling kan de stallingstegoeden van alle in dezelfde gemeente uitgegeven id's opvragen. Dit bijvoorbeeld om te zien of een staller bij het verlaten van een stalling cash moet betalen of dat het stallingstegoed toereikend is.

De stallingstegoeden kunnen in bulk worden opgehaald. Hierbij worden uitsluitend stallingstegoeden weergegeven die ongelijk zijn aan 0.

Ook kan het stallingstegoed van een bepaald id (idcode + idtype) worden opgevraagd.

### Bulk:

GET /citycodes/:citycode/locations/:locationid/balances

Beveiliging: Basic-http-authentication

Rechthebbenden: operator

Response:

```

[
  {
    "idtype": 0,
    "idcode": "0123456790",
    "balance": 5.5
  },
  {
    "idtype": 0,
    "idcode": "09876543210",
    "balance": -1.4
  },
  ...
]

```

### Stallingstegoed van 1 id

GET /citycodes/:citycode/locations/:locationid/idcodes/:idtype/:idcode/balance

(idtype zal over het algemeen 0 (sleutelhanger) of 1 (ov-chip) zijn)

Beveiliging: Basic-http-authentication

Rechthebbenden: operator

Response:

```

{
  "idtype": 0,

```



```
"idcode": "0123456790",  
"balance": 5.5  
}
```