INTOACCESS
THE INTEGRATORS

Net2JsonRestServer

Handleiding 1.9

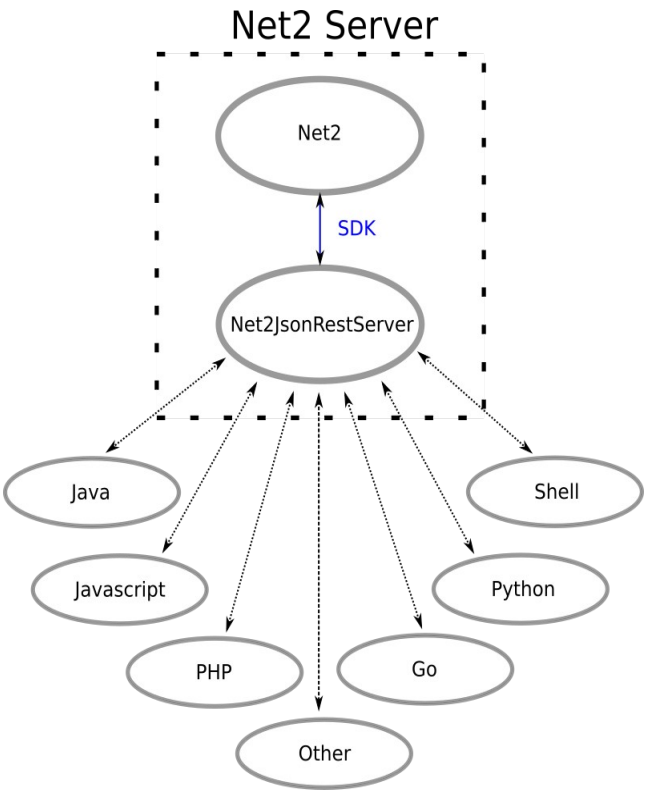
INTOACCESS B.V. THE INTEGRATORS
A.Hofmanweg 5A · 2031 BH · Haarlem · The Netherlands
info@intoaccess.com · www.intoaccess.com · +31(0)23 7505 138

Inhoudsopgave

Principe

De Net2JsonRestServer is een wrapper service rond de Net2 SDK, welke kan worden aangeroepen middels HTTP POST calls met een JSON inhoud.

Dit ontsluit de Net2 SDK voor talen (of operating systems) waar geen dotnet bindings voor bestaan. In het plaatje hier onder ziet u in welke context de applicatie functioneert:

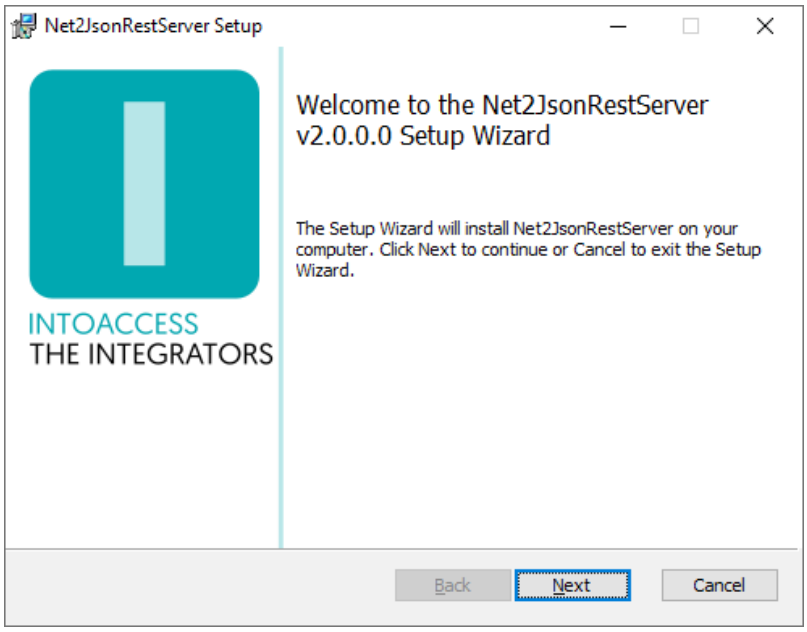


Afbeelding 1

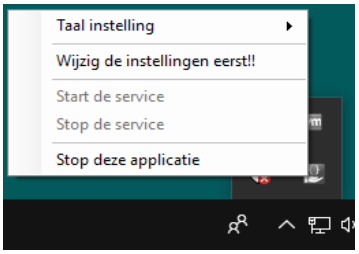
Installatie

De applicatie kan worden geïnstalleerd, middels een enkel installatiebestand: Net2JsonRestServer.msi

Het is niet noodzakelijk om de software op de Net2 server te installeren, maar we raden dit wel aan als meest robuuste configuratie.



Afbeelding 2

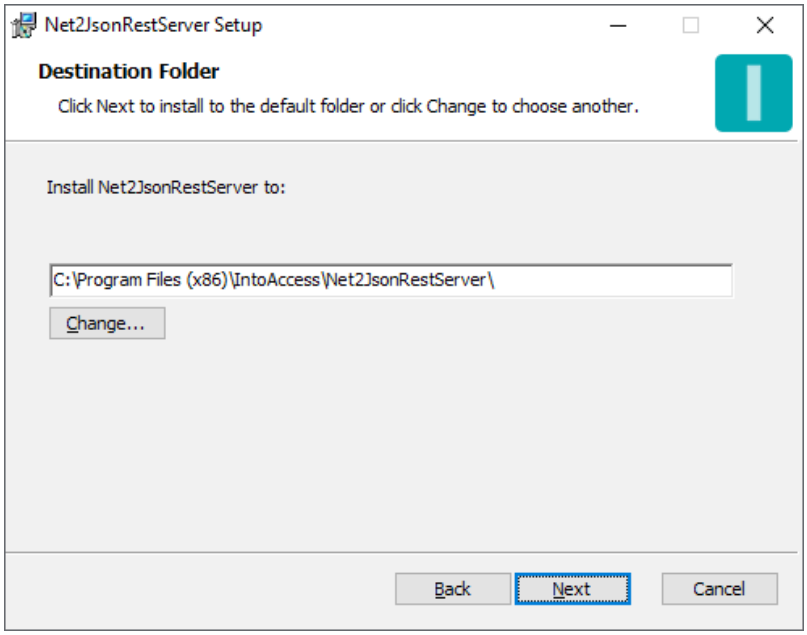


Afbeelding 3

Het eerste dialoogvenster zal aangeven welke applicatie-versie u gaat installeren. Noot: dit nummer zal bij u hoogstwaarschijnlijk anders zijn dan in Afbeelding 2.

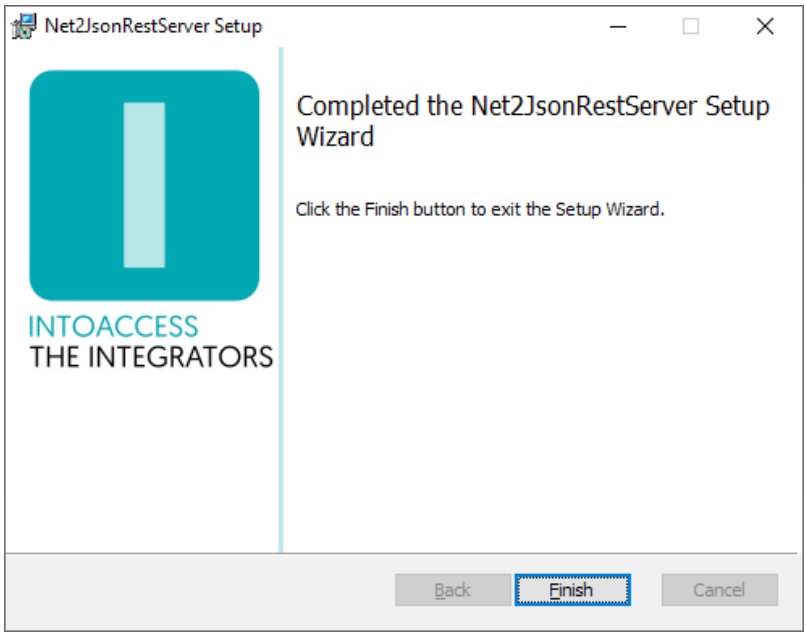
Updates

Hoewel een nieuwere versie een eventueel al aanwezige oudere versie zelf zou moeten vervangen, kun u voor de zekerheid ook eerst de bestaande applicatie de-installeren. De reeds bestaande configuratie instellingen worden daarbij niet verwijderd, dus na installatie van de nieuwe versie zou alles weer moeten werken, zonder opnieuw de configuratie-stappen uit te voeren.



Afbeelding 4

Het tweede dialoogvenster laat zien waar de applicatie zal worden geïnstalleerd. De standaard waarde is daar doorgaans prima.



Afbeelding 5

Het laatste dialoogvenster geeft aan of de installatie is gelukt.

Configuratie

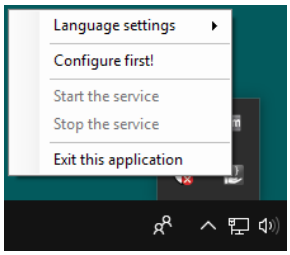
Als configuratie hulp, is de *Net2JsonRestServer manager* beschikbaar, welke u in staat stelt om te configureren:

- hoe de applicatie met Net2 dient te verbinden;
- hoe u de JSON REST server wenst te configureren;
- of u email wilt ontvangen over applicatie start/stop/fouten;

Bij het starten van de manager applicatie wordt een splash-screen getoond, waarna de applicatie onder normale omstandigheden verkleint tot een 'tray icon' in de rechter onder hoek van het scherm.

Afbeelding 6

Afbeelding 7



Afbeelding 8

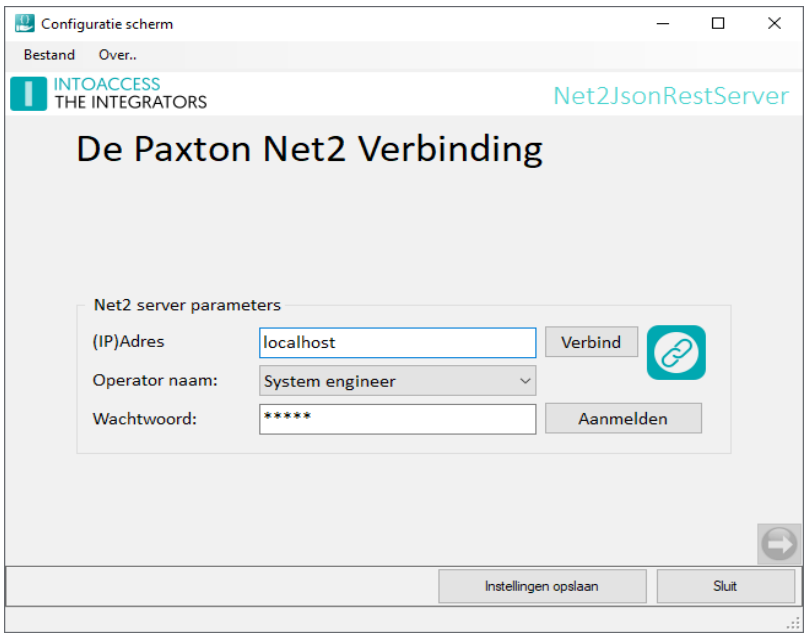
Door rechts te klikken op het icoon, verschijnt een pop-up menu met diverse opties (mits het configuratie-scherm niet al is geopend).

- Taal instelling: Kies uw taal;
- Wijzig instellingen (eerst): Start de applicatie configuratie;
- Start de service: Start de service (dat kan pas na configuratie);
- Stop de service: Stop de service (dat kan pas na configuratie);
- Stop deze applicatie: Stop de manager applicatie.

De kleur van het icoon is overigens een indicatie van of de service draait. Wanneer de service (nog) niet draait, is de kleur grijs; wanneer de service draait, is het icoon gekleurd.

Net2 verbinding

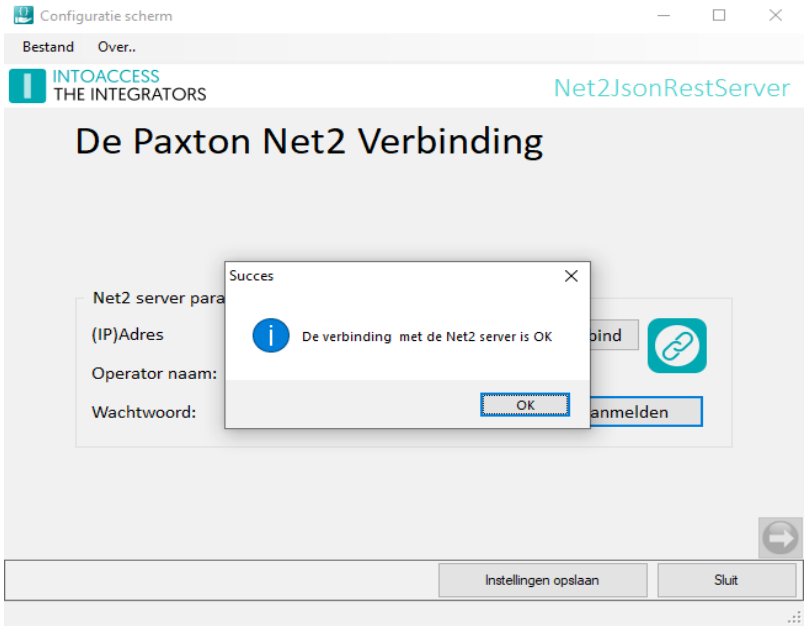
De eerste configuratie-pagina is voor het definiëren van de Net2 verbinding. Wanneer u voor de eerste keer de configuratie start, zal dit enkele stappen vergen, maar nadat deze instellingen zijn opgeslagen, zal het bij een volgende keer automatisch verlopen.



Afbeelding 9

- Voer het (ip)adres in van de Net2 server. Als u de aanwezigheid app installeert op de Net2 server zelf, kunt u de 'localhost' waarde laten staan. Gebruik geen extern adapter ip adres in dat geval!!
- Klik op de verbind knop; de app poogt nu bij Net2 de operators op te halen. (deze gebruikers vindt u onder “Net2 systeembeheerders” van de Net2 applicatie)
- Kies een operator waarmee u wilt dat de applicatie inlogt. Dit dient een operator te zijn met de “Systeembeheerder” rol.
- Voer het bijbehorende wachtwoord in.
- Klik op aanmelden.

Als alles goed gaat, zal er een bevestiging komen dat de verbinding is gelukt en de pijl in de rechter onder hoek van grijs naar gekleurd veranderen.

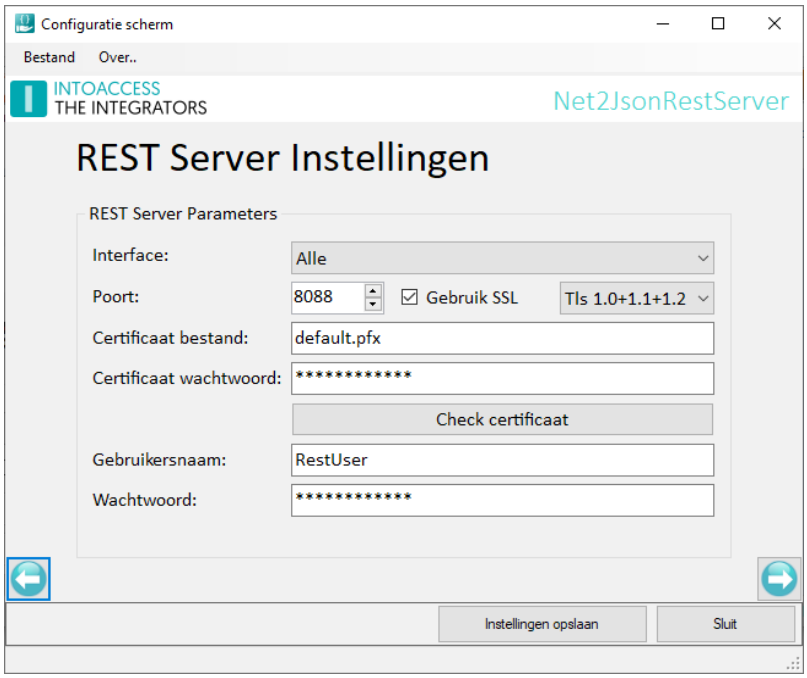


Afbeelding 10

U kunt nu naar het volgende configuratie-scherm, door op de pijl naar rechts te klikken.

REST Server instellingen

In dit configuratie scherm kunt u definiëren de REST server zich dient te gedragen:



Afbeelding 11

- Interface:

  Hier kunt u opgeven of de service 'luistert' op alle externe interfaces of alleen lokaal.

  - Alle: De service is toegankelijk via alle externe (IPv4) interfaces die aan een netwerk zijn verbonden.
  - Alleen lokaal: De service is alleen toegankelijk op de computer zelf.
- Poort:

  De TCP/IP poort waarop de service 'luistert'.
- Gebruik SSL:

  Indien u deze optie aanvinkt, zal de communicatie versleuteld zijn. Indien via een niet vertrouwd netwerk wordt gecommuniceerd, is het inschakelen van deze optie sterk aan te raden. U kunt tevens een keus maken uit de te ondersteunen SSL protocollen. SSL1/SSL2/SSL3 zijn wegens de onveiligheid geen keuze optie (meer).
- Certificaat bestand:

  Indien gebruik wordt gemaakt van SSL, is het noodzakelijk om een certificaat bestand aan te leveren. De applicatie biedt standaard een 'self signed' certificaat, maar niet alle (client) programma's zullen dit accepteren. Een officieel certificaat wordt uitgegeven door een certificaat autoriteit en is specifiek voor een bepaald domein. Een mogelijke leverancier van certificaten is bijvoorbeeld Xolphin: https://www.xolphin.com/

- Certificaat wachtwoord:

  Om een PFX type certificaat bestand te kunnen laden, is doorgaans een wachtwoord nodig, welke u hier kunt opgeven.

  Door op de knop “Check certificaat” te drukken, kunt u controleren of de applicatie het certificaat bestand correct kan interpreteren.

- Gebruikersnaam:

  Om gebruik te kunnen maken van de service, dient de client zich te authenticeren. Dit gebeurt middels ‘Basic Authentication’, wat een standaard is in HTTP communicatie.

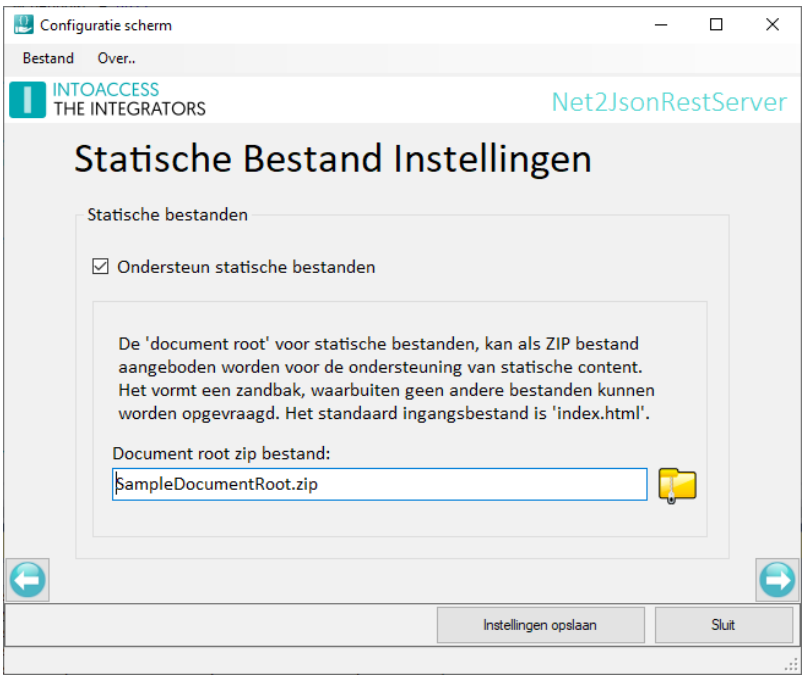
  U kunt in dit veld de gebruikersnaam opgeven. Standaard is dit “RestUser”.

- Wachtwoord:

  Een wachtwoord maakt ook deel uit van de ‘Basic Authentication’ en kan hier worden opgegeven. Standaard is dit “RestPassword”.

Statische Bestand Instellingen

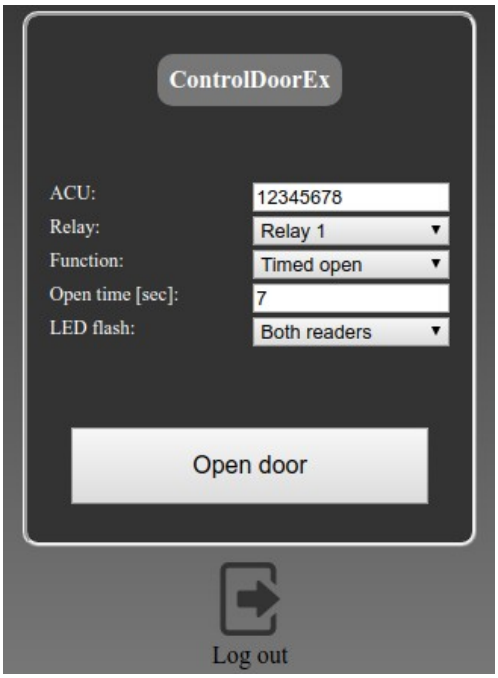
De service applicatie kan ook statische bestanden afhandelen. Om hier gebruik van te maken, dient u uw statische bestanden aan te leveren in een zip archief. Dit zip archief dient als 'zandbak' waarbuiten geen andere bestanden kunnen worden gelezen. Klik op het zip archief icoon om een zip file op het bestandssysteem te selecteren.



Afbeelding 12

Er is een voorbeeld implementatie meegeleverd, welke kan worden gevonden in de installatie folder. Dit bestand heet **SampleDocumentRoot.zip** en bevat een eenvoudige gebruikersinterface om de **ControlDoorEx** functie aan te roepen. U kunt dit voorbeeld gebruiken om een snelle start te maken met uw eigen web inplementatie. Om het eenvoudig te houden, is in het voorbeeld geen gebruik gemaakt van frameworks(zoals JQuery of Vue), maar u kunt gebruiken wat u wilt.

Indien geen specifiek html bestand is aangegeven bij het openen van de url naar de 'website' (zoals bijvoorbeeld http://localhost:8088), dan zal de applicatie proberen **index.html** te openen als het kan worden gevonden in het zip archief.



Afbeelding 13

Wees er van bewust dat de ControlDoorEx SDK aanroep ‘true’ zal retourneren, ongeacht of de opgegeven ACU daadwerkelijk bestaat.

Noot: Gebruik dit soort ‘doe het zelf’ tools alleen in een veilige netwerk omgeving.

ACU monitoring

Om ACU gerelateerde gebeurtenissen ongevraagd (niet pollend) te ontvangen, kunt u aangeven in welke ACU u geïnteresseerd bent. Zodra de service is gestart, zullen de gebeurtenissen van de aangevinkte ACU's naar iedere client applicatie worden gestuurd die middels een web socket is verbonden aan het **/Events** pad.

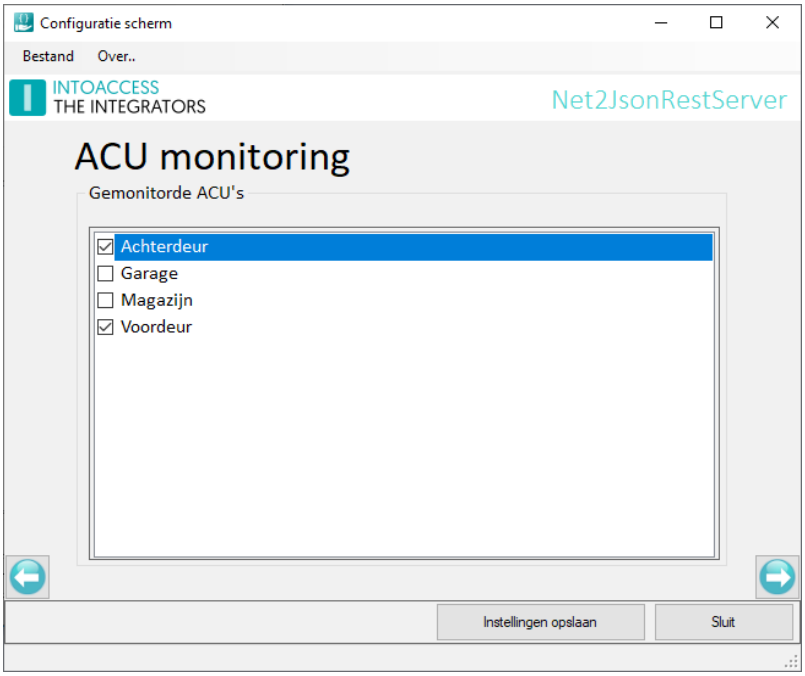


Image 14

Noot: het kost doorgaans enkele seconden om een ACU gebeurtenis te laten propageren naar de server.

Voor meer details over de structuur van de gebeurtenis data, zie hoofdstuk

Email instellingen

De email configuratie is optioneel en biedt de mogelijkheid om applicatie-meldingen naar een beheerder te laten sturen.

Dit is met name nuttig om vroegtijdig een indicatie te ontvangen dat er mogelijk iets mis is met de werking van de applicatie.

Webmail

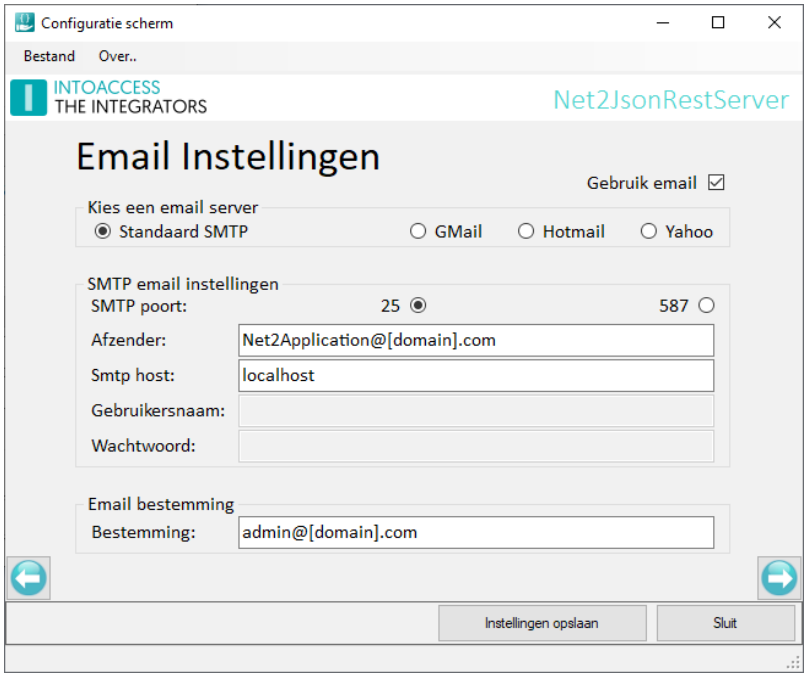
Voor het gebruik van webmail providers (zoals Gmail), kan het noodzakelijk zijn om de beveiliging te verlagen (provider account setting).

SMTP

The SMTP instellingen staan u toe een SMTP server en port te configureren. Indien u port 587 kiest, is geauthenticeerde SMTP mogelijk, waarbij u gebruikersnaam en wachtwoord kunt opgeven.

Email bestemming

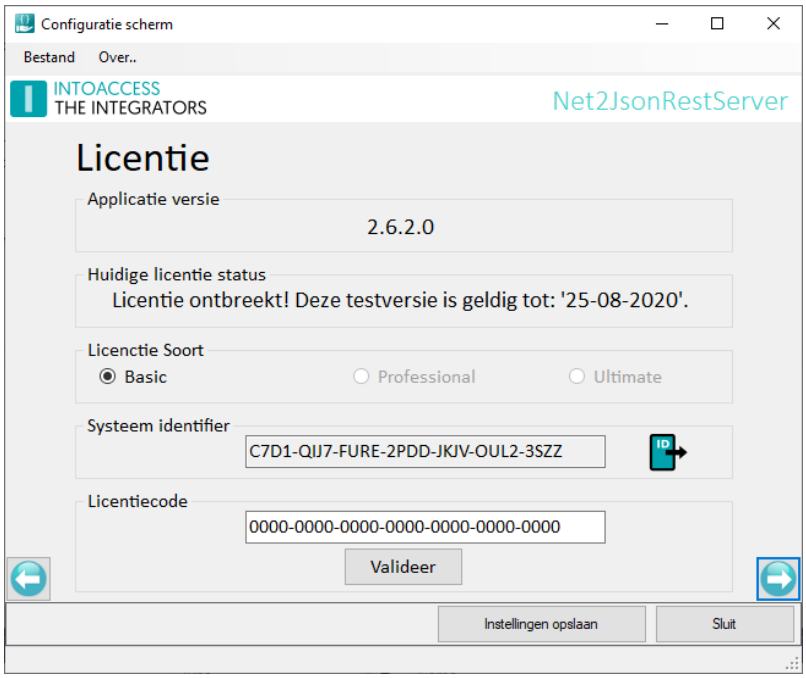
Om gelijktijdig meerdere personen te adresseren, kunnen gescheiden door een ';' extra email adressen worden toegevoegd.



Afbeelding 15

Applicatie licentie

De proef/test versie zoals deze van de IntoAccess website kan worden gedownload, is volledig functioneel maar zal zonder licentie na een bepaalde datum niet meer werken.



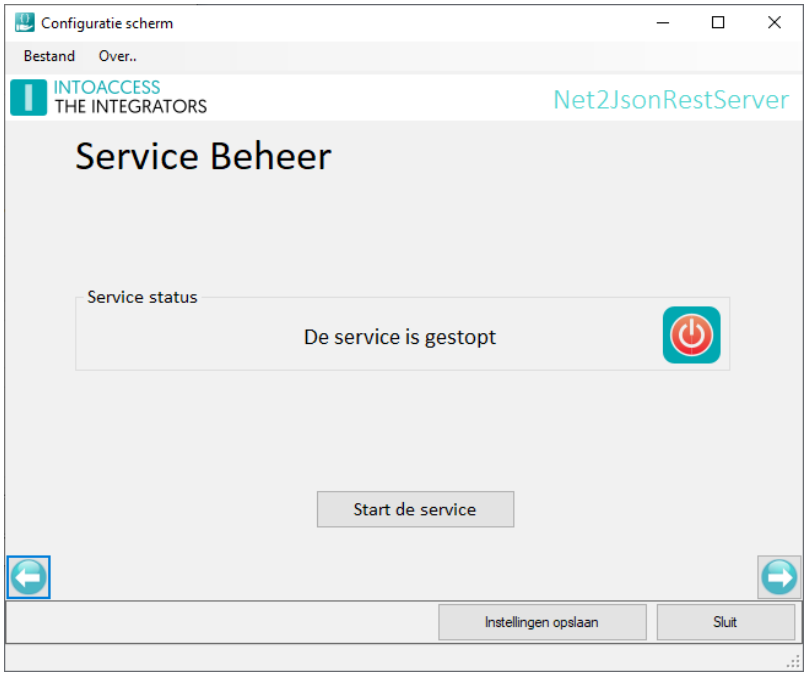
Afbeelding 16

Na aanschaf van een licentie, kunt u het “Systeem identifier” export bestand aanmaken (druk op icoon) en emailen aan IntoAccess (info@intoaccess.com), waarna u de licentiecode ontvangt.

Let op: de Systeem Identifier is verschillend per PC en daardoor ook de benodigde licentie code.

Service beheer

De service beheer pagina biedt de mogelijkheid om achtergrond service, die de periodieke import uitvoert, te stoppen en starten.



Afbeelding 17

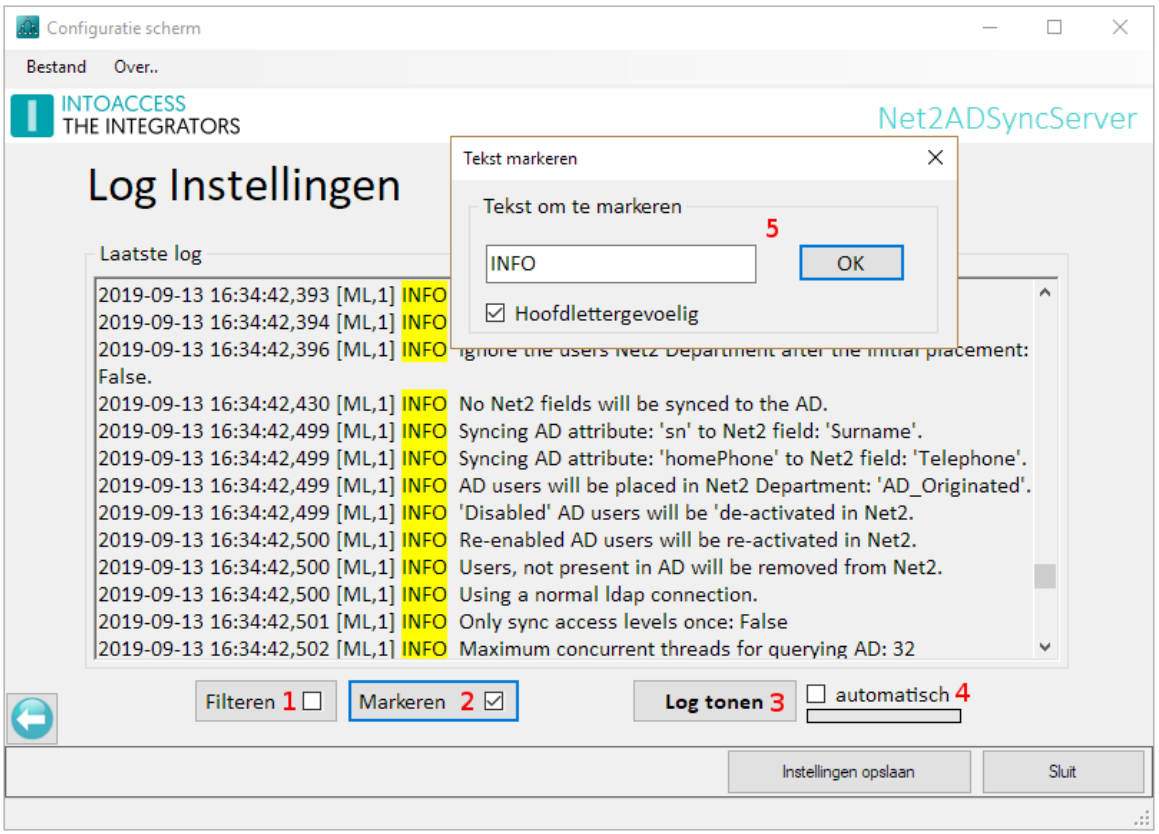
De overige manieren waarop de service kan worden gestart en gestopt zijn:

- Het tray icon pop-up menu;
- De Windows service manager (zoek “Net2JsonRestServer”);

Log instellingen

Deze pagina, zie afbeelding 18, biedt de mogelijkheid om de laatste (max. 500) regels uit de logfile te bekijken. De applicatie logt zeer gedetailleerd welke stappen de applicatie allemaal doorloopt. Mocht de applicatie met een onverwacht probleem geconfronteerd worden dan kan, met behulp van deze logfile, de oorzaak vaak snel worden gevonden.

Het loont zeker de moeite om even naar de laatste regels in deze logfile te kijken als de applicatie niet wil starten, of anderszins onverwacht gedrag vertoont.



Afbeelding 18

Deze pagina biedt ook de mogelijkheid om de logfile op bepaalde termen te filteren (1) en\of bepaalde termen te markeren (2). Een voor de hand liggende ‘filter term’ zou bijvoorbeeld het woord “ERROR” of “WARN” kunnen zijn. Als de applicatie goed werkt zouden beide termen niet voor mogen komen in de logfile.
Optie (4) biedt de mogelijkheid om de logfile automatisch met een vaste interval opnieuw te laden.

De log file(s) zelf bevinden zich in de folder:
c:\IntoAccess\Logging\Net2JsonRestServer

SDK calls

In dit hoofdstuk vindt u de geïmplementeerde Net2 SDK calls. Dit is een sub set van van het totale aantal beschikbare SDK calls, maar kan op aanvraag worden uitgebreid.

Het uitgangspunt voor de wrapper is dat alle informatie die middels de QueryDB call kan worden opgevraagd, niet op een andere wijze wordt aangeboden.

Http response codes

De volgende codes worden gebruikt:

- 200: succes
- 400: bad request; verstuurd als een aanroep ongeldig is (json of anders).
- 401: unauthorized; verstuurd wanneer er geen inlog gegevens opgegeven zijn.
- 403: forbidden; verstuurd wanneer de inlog gegevens foutief zijn.
- 404: not found; verstuurd als het url pad niet bestaat.
- 500: internal server error; verstuurd bij een algemene fout (catch all)

Kaart typen

De lijst met 'kaart typen'en hun id:

```
0     -     Niet gespecificeerd
1     -     Proximity kaart
2     -     Proximity ISO kaart
3     -     Sleutel
4     -     Handsvrije sleutel
5     -     WatchProx
6     -     Proximity ISO card zonder magneetstrip
7     -     Nummerplaat voertuig
8     -     Handsvrije sleutelkaart
9     -     Vingerafdruk verificatie kaart
```

AddAccessLevel

Voeg een nieuwe autorisatie toe.

Request

url: "http(s)://<ip>:<port>/AddAccessLevel"

JSON

```
{
    "accessLevelName": "SomeAlName",
    "accessLevelDetails": [
         {
             "timezoneId": 1,
             "areaId": 1000001
         },
         {
             "timezoneId": 1,
             "areaId": 1000000
        }
    ]
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='{
"accessLevelName": "SomeAccessLevel",
"accessLevelDetails": [{
"timezoneId": 1,
"areaId": 1000001
},
{
"timezoneId": 1,
"areaId": 1000000
}
]
}' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/AddAccessLevel"
```

AddCard

Voeg een nieuwe kaart/kentekenplaat toe bij een gebruiker.

Ter kennisgeving:

- De cardNo kan een kaartnummer “0-99999999” zijn of een kenteken.
- De cardTypeId dient het juiste type te hebben (7) in geval van een kenteken.
  (zie hoofdstuk Kaart typen)

Request

url: "http://<ip>:<port>/AddCard"

JSON

```
{
    "cardNo": "34567890",
    "cardTypeId": 1,
    "userId": 18,
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='{
"cardNo": "34567890",
"cardTypeId": 1,
"userId": 18,
}' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/AddCard"
```

AddDepartment

Voeg een nieuwe afdeling toe.

Ter kennisgeving:

* De parameter is de afdeling naam.

* Geeft false terug als de afdeling reeds bestaat.

Request

url: "http://<ip>:<port>/AddDepartment"

JSON

```
"SomeDepartment"
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='"SomeDepartment"' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/AddDepartment"
```

AddNewUser

Voeg een nieuwe gebruiker toe.

Ter kennisgeving:

- String waarden kunnen weggelaten worden of op null gezet, om ze leeg te laten..
- Als activationDate niet opgegeven of null is, wordt 'vandaag' gebruikt.
- Als expiryDate niet opgegeven of null is, is er geen vervaldatum.
- Minimale datum waarde: 1999-01-01.
- Als pinCode niet gezet of null is, wordt geen pincode toegekend.
- Als cardNumber niet gezet of null is, wordt geen kaartnummer toegekend.
- De eerste customFields waarde [0] wordt genegeerd en kan op null gezet worden.
- Als customFields niet ingesteld is, zijn de waarden blanco.

Request

url: "http://<ip>:<port>/AddNewUser"

JSON


```
{
    "accessLevelId": 1,
    "departmentId": 2,
    "antiPassbackInd": false,
    "alarmUserInd": false,
    "firstName": "John_50",
    "middleName": "Patrick_50",
    "surname": "Doe_50",
    "telephoneNo": "123456_30",
    "telephoneExtension": "78_10",
    "pinCode": "1234_8",
    "activationDate": "2019-05-01",
    "cardNumber": 12345678,
    "cardTypeId": 1,
    "active": true,
    "faxNo": "654321_30",
    "expiryDate": "2020-05-01",
    "customFields": [
        null,
        "Field1_100",
        "Field2_100",
        "Field3_50",
        "Field4_50",
        "Field5_50",
        "Field6_50",
        "Field7_50",
        "Field8_50",
        "Field9_50",
        "Field10_50",
        "Field11_50",
        "Field12_50",
        "Field13_memo",
        "Field14_50"
    ]
}
```


Reply

UserID van de nieuwe gebruiker, of-101 (AddNewUserFailed) indien mislukt.

JSON


```
123
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='{
"accessLevelId": 1,
"departmentId": 1,
"antiPassbackInd":false,
"alarmuserInd": false,
"firstName": "John",
"middleName": "Patrick",
"surname": "Doe",
"telephoneNo": "123456",
"telephoneExtension": "12",
"pinCode": "",
"activationDate": "2020-01-01",
"cardNumber": 98989898,
"cardTypeId": 3,
"active": true,
"faxNo": "654321",
"expiryDate": "2023-01-01",
"customFields": [],
}' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/AddNewUser"
```

AddTimezone

Voeg een nieuwe tijdzone toe:

Ter kennisgeving:

- Meerdere tijdzones met dezelfde naam zijn toegestaan.
- Minimale start tijd: 00:00:00
- Maximale eind tijd: 23:59:59
- 0=vakantie dag, 1=zondag (standaard)
- Maximaal 64 tijdzones mogelijk

Request

url: "http://<ip>:<port>/AddTimezone"

JSON

{

```
    "timezoneName": "SomeTimezone",
    "timeSlots": [
        {
            "day": 1,
            "start": "00:00:00",
            "end": "23:59:59"
        },
        {
            "day": 2,
            "start": "17:00:00",
            "end": "19:00:00"
        }
    ]
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='{
"timezoneName": "SomeTimezone",
"timeSlots": [
{
"day": 1,
"start": "00:00:00",
"end": "23:59:59"
},
{
"day": 2,
"start": "17:00:00",
"end": "19:00:00"
}
]
}
' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/AddTimezone"
```

ControlDoorEx

Bedien een deur.

Parameters:

- address: ACU serie nummer
- relais: 0 = relais 1, 1 = relais
- function: 0 = Sluit, 1 = Tijd open, 2 = Houd open
- doorOpenTime: Open tijd in msec
- ledFlash: 0 = geen knipperen, 1 =knipper bij lezer 1, 2 = knipper bij lezer 2, 3 = knipper bij beide lezers (werkt mogelijk niet volgens spec)

Request

url: "http://<ip>:<port>/ControlDoorEx"

JSON

```
{
    "address": 123,
    "relais": 0,
    "function": 1,
    "doorOpenTime": 3000,
    "ledFlash": 3
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='{
"address": 1156774,
"relais": 0,
"function": 1,
"doorOpenTime": 3000,
"ledFlash": 3
}' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/ControlDoorEx"
```

DeleteAccessLevel

Verwijder een autorisatie.

Ter kennisgeving:

- De parameter is de autorisatie id.
- Geeft true terug, zelfs als de autorisatie niet bestaat.

Request

url: "http://<ip>:<port>/DeleteAccessLevel"

JSON

```
123
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='123' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/DeleteAccessLevel"
```

DeleteCard

Verwijder kaart/kentekenplaat.

Ter kennisgeving:

- De parameter is het kaartnummer of kenteken.
- Geeft true terug, zelfs als de kaart/kentekenplaat niet bestaat.

Request

url: "http://<ip>:<port>/DeleteCard"

JSON

```
"34567890"
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='"34567890"' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/DeleteCard"
```

DeleteDepartment

Verwijder een afdeling.

Ter kennisgeving:

- De parameter is de afdeling id.
- Geeft true terug, zelfs wanneer de afdeling niet bestaat.

Request

url: "http://<ip>:<port>/DeleteDepartment"

JSON

```
123
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='123' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/DeleteDepartment"
```

DeleteTimezone

Verwijder een tijdzone.

Ter kennisgeving:

- De parameter is de tijdzone id.
- Geeft true terug, zelfs wanneer de tijdzone niet bestaat.

Request

url: "http://<ip>:<port>/DeleteTimezone"

JSON

```
123
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='123' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/DeleteTimezone"
```

EndLockdown

Haalt het systeem uit de lockdown staat.

Request

url: "http://<ip>:<port>/EndLockdown"

JSON

```
<leeg>
```

Reply

True bij succes, false bij een fout. (Geeft ook 'true' indien niet in lockdown staat)

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/EndLockdown"
```

GetListOfOperators

Geeft lijst met Net2 operators terug.

Request

url: "http://<ip>:<port>/GetListOfOperators"

JSON

<leeg>

Reply

Lijst met operator/user id en operator/user naam.

JSON

```
[
    { "0": "Systeembeheerder"},
    {"1279": "John Doe"},
    {"1446": "Mr. Supervisor"},
    {"1447": "Mr. Readonly"}
]
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/GetListOfOperators"
```

GetUserImage

Geeft de gebruiker foto terug als een base64 string, of null als geen foto wordt gevonden.

Ter kennisgeving:

- De parameter is de gebruiker id.

Request

url: "http://<ip>:<port>/GetUserImage"

JSON

```
1234
```

Reply

Base64 ge-encodeerd jpeg plaatje of null.

JSON

```
“/9j/4AAQSkZJRgABAQAAAQABAAD//gAqSW50ZWwoUikgSlBFRyBMaWJyYXJ5LCB2..."
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='1234' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/GetUserImage"
```

GetServerVersion

Geeft Net2JsonRestServer versie terug.

Request

url: "http://<ip>:<port>/GetServerVersion"

JSON

```
<leeg>
```

Reply

Versie nummer string.

JSON

```
"1.0.0.0"
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/GetServerVersion"
```

InitiateLockdown

Zet systeem in de lockdown staat.

Request

url: "http://<ip>:<port>/InitiateLockdown"

JSON

```
<leeg>
```

Reply

True bij succes, false bij een fout. (Geeft ook 'true' indien al in lockdown staat)

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/InitiateLockdown"
```

LastErrorMessage

Geeft de laatste Net2 foutmelding terug.

Ter kennisgeving:

- Het is de laatste foutmelding voor een (geldige) Net2 aanroep. Een ongeldige aanroep (bijv. foutieve JSON) zal deze niet instellen..
- Als de voorgaande (Net2) aanroep succesvol was, zal de aanroep een lege string terug geven.
- De GetServerVersion aanroep (non Net2) reset niet de laatste Net2 foutmelding.

Request

url: "http://<ip>:<port>/LastErrorMessage"

JSON

```
<leeg>
```

Reply

Laatste foutmelding string.

JSON

```
"Some error message"
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/LastErrorMessage"
```

PurgeUser

Verwijder gebruiker volledig.

Ter kennisgeving:

- De parameter is de gebruiker id.
- Het verwijderen van een gebruiker zal tevens de (specifieke) log referenties er naar toe verwijderen.

Request

url: "http://<ip>:<port>/PurgeUser"

JSON

```
123
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='1098' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/PurgeUser"
```

QueryDb

Voer een query uit op de Net2 database.

Tip: Gebruik onze gratis Net2Query tool om de database te verkennen en uw queries te testen: https://www.intoaccess.com/products/Net2Query

Request

url: "http://<ip>:<port>/QueryDB"

JSON

```
"select * from AccessLevels"
```

Reply

Noot: kolommen van het type 'varbinary/Byte[]', worden sinds v2.7.1 terug gegeven als een base64 gecodeerde byte array.

JSON

```
[
    {
        "AccessLevel": 0,
        "AccessLevelName": "No access"
    },
    {
        "AccessLevel": 1,
        "AccessLevelName": "All hours, all doors"
    },
    {
        "AccessLevel": 3,
        "AccessLevelName": "Working hours"
    }
]
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='"select * from AccessLevels"' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/QueryDB"
```

UpdateAccessLevel

Wijzig een bestaande autorisatie.

Ter kennisgeving:

- Meervoudige autorisaties met dezelfde naam zijn toegestaan.

- Maximaal 256 autorisaties toegestaan.
- Indien de naam wordt leeg gelaten, zal deze ongewijzigd blijven.

Request

url: "http://<ip>:<port>/UpdateAccessLevel"

JSON

```
{
    "accessLevelId": 11,
    "accessLevelName": "SomeOtherAccessLevel",
    "accessLevelDetails": [
         {
             "timezoneId": 2,
             "areaId": 1000001
         },
         {
             "timezoneId": 2,
             "areaId": 1000000
        }
    ]
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='{
"accessLevelId": 11,
"accessLevelName": "SomeOtherAccessLevel",
"accessLevelDetails": [{
"timezoneId": 2,
"areaId": 1000001
},
{
"timezoneId": 2,
"areaId": 1000000
}
]
```

```
}' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/UpdateAccessLevel"
```

UpdateCard

Wijzig een bestaande kaart/kentekenplaat.

Ter kennisgeving:

- De cardNo kan een kaartnummer “0-99999999” zijn of een kenteken.
- De cardTypeId dient het juiste type te hebben (7) in geval van een kenteken.
  (zie hoofdstuk Kaart typen)

Request

url: "http://<ip>:<port>/UpdateCard"

JSON

```
{
    "cardNo": "34567890",
    "cardTypeId": 1,
    "userId": 18,
    "lostCard": false
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Sample with wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='{
"cardNo": "34567890",
"cardTypeId": 3,
"userId": 18,
"lostCard": false
}' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/UpdateCard"
```

UpdateDepartment

Wijzig een bestaande afdeling

Ter kennisgeving:

* Een update naar een afdeling met dezelfde naam geeft false terug.

Request

url: "http://<ip>:<port>/UpdateDepartment"

JSON

```
{
    "deptId": 123,
    "deptName": "OtherName"
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='{
"deptId": 123,
"deptName": "OtherName"
}' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/UdpateDepartment"
```

UpdateTimezone

Wijzig een bestaande tijdzone.

Ter kennisgeving:

- Meerdere tijdzones met dezelfde naam zijn toegestaan.
- Minimale start tijd: 00:00:00
- Maximale eind tijd: 23:59:59

- 0=vakantie dag, 1=zondag (standaard)
- Maximaal 64 tijdzones mogelijk
- Als de naam wordt leeg gelaten, zal deze ongewijzigd blijven.

Request

url: "http://<ip>:<port>/UpdateTimezone"

JSON

```
{
    "timezoneId": 4,
    "timezoneName": "SomeOtherTimezone",
    "timeSlots": [
        {
            "day": 4,
            "start": "00:00:00",
            "end": "23:59:59"
        },
        {
            "day": 5,
            "start": "17:00:00",
            "end": "19:00:00"
        }
    ]
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='{
"timezoneId": 4,
"timezoneName": "SomeOtherTimezone",
"timeSlots": [
{
"day": 4,
"start": "00:00:00",
"end": "23:59:59"
},
```

```
{
"day": 5,
"start": "17:00:00",
"end": "19:00:00"
}
]
}
' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/UpdateTimezone"
```

UpdateUserRecord

Wijzig een bestaande gebruiker.

Ter kennisgeving:

- String waarden kunnen weggelaten worden of op null gezet, om ze ongewijzigd te laten.
- Als activationDate niet opgegeven of null is, wordt 'vandaag' gebruikt.
- Als expiryDate niet opgegeven of null is, is er geen vervaldatum.
- Minimale datum waarde: 1999-01-01.
- Als pinCode niet gezet of null is, wordt geen pincode toegekend.
- Als cardNumber niet gezet of null is, wordt geen kaartnummer toegekend.
- De eerste customFields waarde [0] wordt genegeerd en kan op null gezet worden.
- Als customFields niet ingesteld zijn, blijven de waarden ongewijzigd.
- Als een individueel customField niet ingesteld of null is, blijft de waarde ongewijzigd.

Request

url: "http://<ip>:<port>/UpdateUserRecord"

JSON

```
{
    "userId": 123,
    "accessLevelId": 1,
    "departmentId": 2,
    "antiPassbackInd": false,
    "alarmUserInd": false,
    "firstName": "John_50",
    "middleName": "Patrick_50",
    "surname": "Doe_50",
    "telephoneNo": "123456_30",
```

```
    "telephoneExtension": "78_10",
    "pinCode": "1234_8",
    "activationDate": "2019-05-01",
    "activeInd": true,
    "faxNo": "654321_30",
    "expiryDate": "2020-05-01",
    "customFields": [
        null,
        "Field1_100",
        "Field2_100",
        "Field3_50",
        "Field4_50",
        "Field5_50",
        "Field6_50",
        "Field7_50",
        "Field8_50",
        "Field9_50",
        "Field10_50",
        "Field11_50",
        "Field12_50",
        "Field13_memo",
        "Field14_50"
    ]
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='{
"userId": 50,
"accessLevelId": 0,
"departmentId": 1,
"antiPassbackInd":false,
"alarmuserInd": false,
"firstName": "John",
"middleName": "Patrick",
"surname": "Doe",
```

```
"telephoneNo": "123456",
"telephoneExtension": "12",
"pinCode": "",
"activationDate": null,
"activeInd": true,
"faxNo": "654321",
"expiryDate": null,
"customFields": [],
}' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/UpdateUserRecord"
```

ValidateOperator

Valideer operator met gebruikersnaam wachtwoord en ontvang de toegestane methoden terug.

Request

url: "http://<ip>:<port>/ValidateOperator"

JSON

```
{
    "userId": 0,
    "password": "admin"
}
```

Reply

Lijst met toegestane methodes (naam/nummer), of een lege lijst wanneer de operator niet bestaat, geen rechten heeft of het wachtwoord niet klopt. Let op: na een foute validatie, zal Net2 nog gedurende zo'n 10[s] ook bij het correcte password een reply geven alsof het niet valide is.

JSON

```
[
    { "AddAccessLevel": 1 },
    { "AddACU": 2 },
    { "AddArea": 3 },
    { "AddCamera": 4 },
    ...
]
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='{
```

```
"userId": 0,
"password": "admin"
}' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/ValidateOperator"
```

ViewUserRecords

Haal alle of slechts een subset van de gebruikers records op. Deze aanroep geeft iets meer terug dan middels een QueryDB op de UsersEx view kan worden verkregen. Zo is onder andere de LockdownExempt status van gebruikers op te vragen.

Request

url: "http://<ip>:<port>/ViewUserRecords"

JSON

```
<leeg>
of
"LockdownExempt=1 and UserId>0" (een optionele ‘where clause’)
```

Reply

JSON

```
[
    {
        "AccessLevelID": 1,
        "AccessLevelName": "All hours, all doors",
        "ActivateDate": "2020-03-18T00:00:00",
        "Active": true,
        "AlarmUser": false,
        "AntiPassbackUser": true,
        "CardNo": null,
        "CardTypeID": null,
        "DepartmentID": 0,
        "DepartmentName": "(none)",
        "ExpiryDate": null,
        "Extension": "",
        "Fax": "",
        "Field10_50": "",
        "Field11_50": "",
        "Field12_50": "",
        "Field13_Memo": "",
```

```
        "Field14_50": "",
        "Field1_100": "",
        "Field2_100": "",
        "Field3_50": "",
        "Field4_50": "",
        "Field5_50": "",
        "Field6_50": "",
        "Field7_50": "",
        "Field8_50": "",
        "Field9_50": "",
        "FirstName": "Stephanie",
        "Global": false,
        "ImageData": null,
        "IsAccessLevelUser": true,
        "LastAccessTime": "2020-04-02T16:29:29.69",
        "LastArea": "Storage (Out)",
        "LastAreaID": 124,
        "LastUpdated": "2020-04-06T11:48:43",
        "LockDownExempt": true,
        "MiddleName": "Erika",
        "PIN": "",
        "Picture": null,
        "StaffCategoryID": "0",
        "Surname": "Vandersloot",
        "Telephone": "",
        "UserGUID": "af544469-7e08-4548-a526-3ea98ea9b19e",
        "UserID": 1342,
        "UserName": "Vandersloot"
    },
]
```


Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \
--post-data='"LockdownExempt=1 and UserId>0"' \
--header='Content-Type:application/json' \
"http://$SERVER:$PORT/ViewUserRecords"
```

Ongevraagde data

De server applicatie kan ACU gebeurtenissen ongevraagd over een open web socket verbinding versturen. Om deze optie in te schakelen, zie het configuratie hoofdstuk ACU monitoringACU monitoring.

Url

url: "ws://<ip>:<port>/Events"

Gebeurtenis data structuur

De verzonden gebeurtenissen hebben de volgende structuur:

```
{
  "eventDateTime": "2020-01-30T15:16:50",
  "eventId": 22720,
  "eventType": 46,
  "eventSubType": 61,
  "address": 1484506,
  "subAddress": 11,
  "userId": 0,
  "cardNumber": 0
}
```

- EventDateTime: locale datum/tijd dat de gebeurtenis plaats vond.
- EventId: de database id van het gebeurtenis record.
- EventType: geeft het soort gebeurtenis aan (toegang verkregen, geweigerd, etc).
- EventSubType: geeft het sub type binnen een gebeurtenis soort aan.
- Address: ACU nummer
- UserId: de database id van de gebruiker, indien de gebeurtenis gerelateerd is aan een gebruiker (en deze bekend is).
- cardNumber: kaartnummer / pincode / ‘vertaald’ kenteken (0-99999999)

Eenvoudig node.js code voorbeeld

```
const WebSocket = require('ws');

// --- Adjust according to your own settings ---
// Url to connect to
const URL = 'ws://192.168.1.131:8088/Events';
// Basic authentication credentials
const USER = 'RestUser';
const PASSWD = 'RestPassword';

// Options for the websocket connection.
// In particular, the authentication part and explicit disable of perMessageDeflate.
const OPTIONS = {
  headers: {
    'Authorization': 'Basic ' + Buffer.from(USER + ':' + PASSWD).toString('base64')
  },
  perMessageDeflate: false
};

// Make a web socket connection
const ws = new WebSocket(URL, OPTIONS);
ws.on('open', () => {
  console.log('Connection opened');
});
ws.on('close', (code) => {
  console.log(`Connection closed, code ${code}`);
});
ws.on('error', (error) => {
  console.log('Error:', error);
});
ws.on('message', (data) => {
  try {
    // Parse received message to check is valid JSON
    const json = JSON.parse(data);
    // Pretty print received JSON data
    console.log(JSON.stringify(json, null, '  '));
```

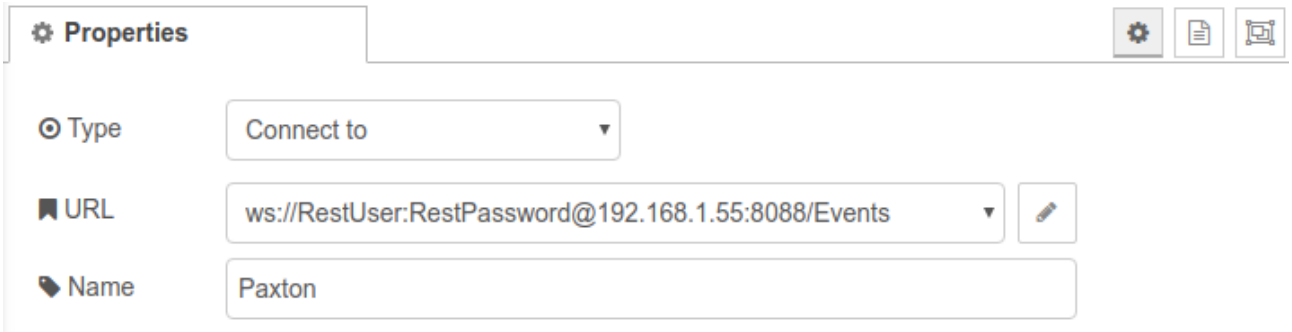
```
  } catch(err) {
    console.log('Received an invalid JSON object');
  }
});
```

Node-RED integratie

De Net2JsonRestServer, kan worden gebruikt in combinatie met de Node-RED applicatie/server. Indien gebruikt voor het verwerken van “ongevraagde” (websocket) data, adviseren we het gebruik van Node v14 of hoger, daar er een issue lijkt te zijn met de message deflation afhandeling in sommige versies van Node v12.

Ontvangen ongevraagde data

Zorg eerst dat ACU monitoring is ingeschakeld bij de Net2JsonRestServer. Maak vervolgens een Node-RED flow, gebruik makend van de standaard “websocket in” component, met de volgende instellingen:

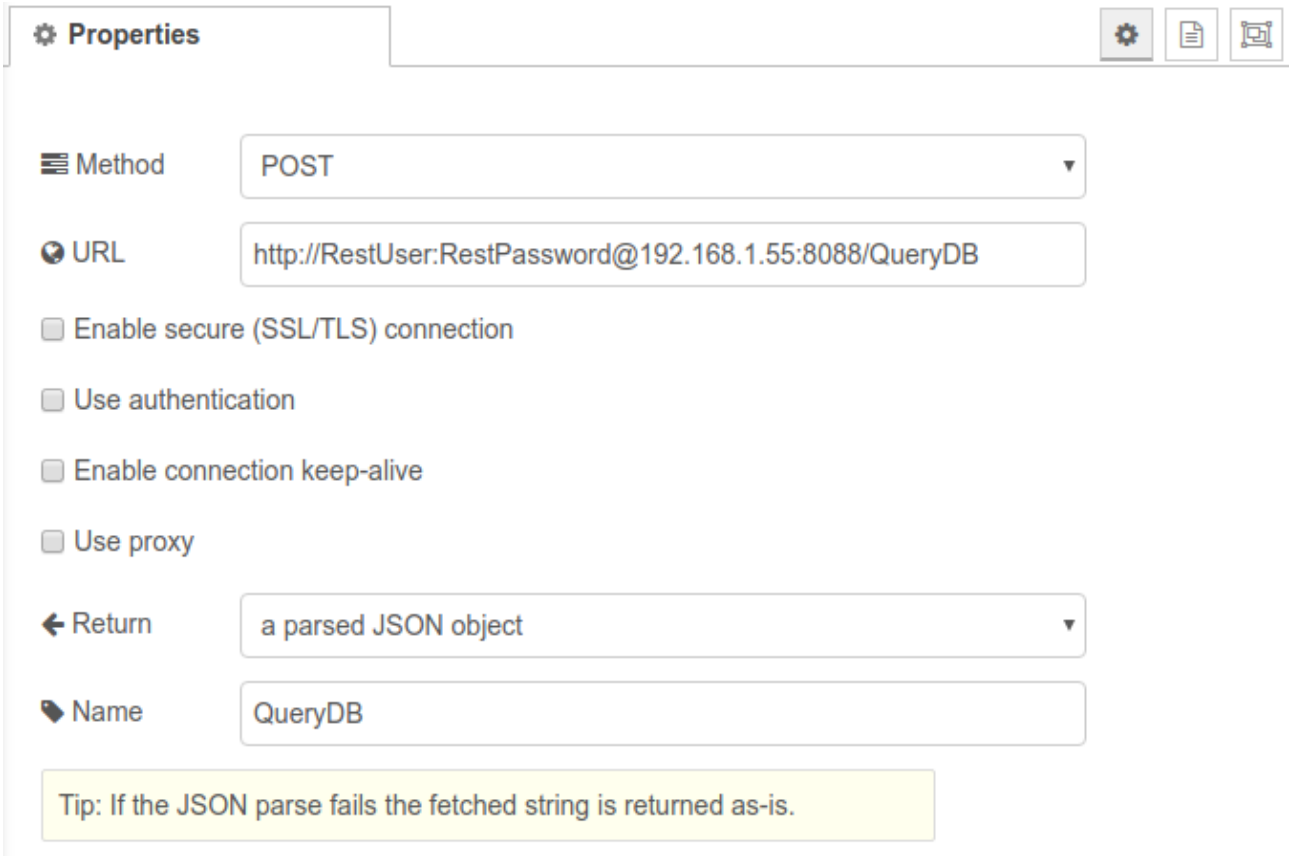


Wijzig de gebruikersnaam, wachtwoord en ip adres voor uw situatie.

Ophalen data

In het voorbeeld hier onder, wordt extra gebruiker informatie opgevraagd, nadat een ongevraagd bericht is ontvangen met een userId. Gebruik allereerst een “function”, welke een query string maakt om uit te voeren op Net2, gebruik makend van de userId als parameter.

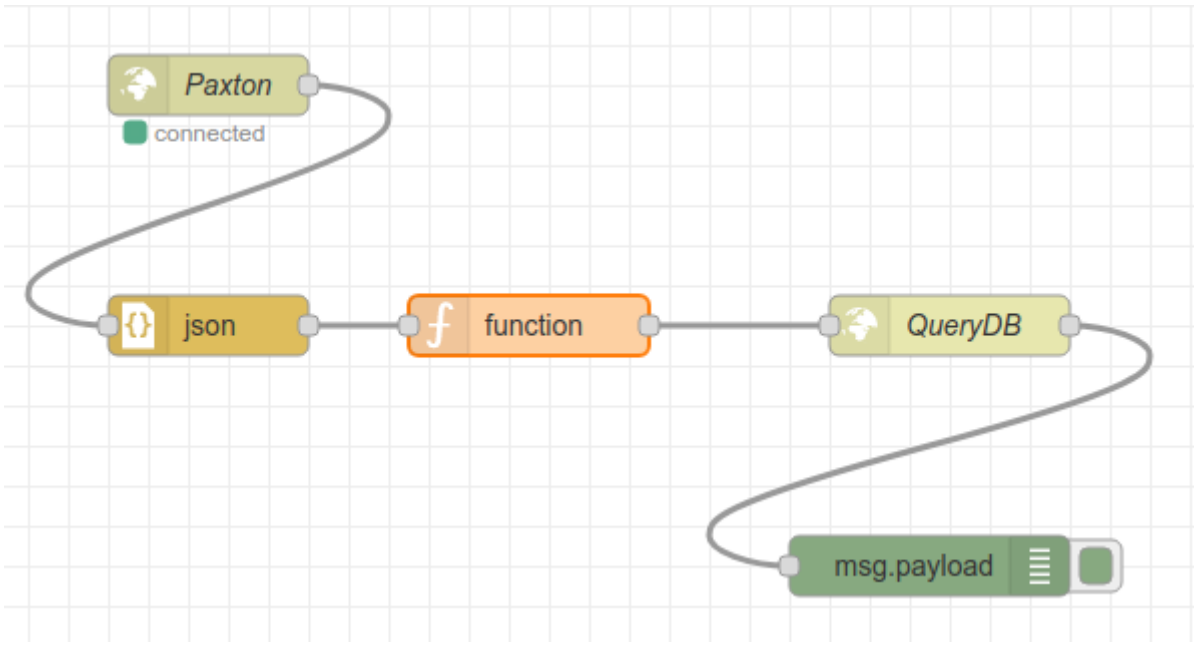
Koppel daarna dit component aan een “http request” component, welke de query zal uitvoeren en het resultaat zal terug/doorgeven:



Noot: u kunt in principe elke Net2JsonRest methode aanroepen op dezelfde wijze, zoals bijvoorbeeld het openen van een deur.

Voorbeeld flow

U kunt de code voor de onderstaande eenvoudige voorbeeld flow downloaden:



https://www.intoaccess.com/downloads/Paxton_Node-RED_sample.json

Het ontvangt ongevraagde data van Paxton Net2, haalt het volledige gebruikers-record op wat bij het event hoort en logt dit in een debug component.

Zorg eerst dat u de component instellingen aanpast aan uw configuratie. Daarna zou u data moeten zien verschijnen in de debug log, wanneer er een kaart/token wordt aangeboden.

Numerieke waarden gebruikt door de SDK

Dit hoofdstuk geeft enig inzicht in enkele 'magische' nummers die binnen de context van de SDK voor komen. Het is geen volledige lijst en standaard niet up to date, maar kan kan van nut zijn in sommige gevallen.

Device status flag

De StatusFlag kolomwaarde, zoals deze in de Devices view staat, zal een waarde bevatten die een bit patroon voorstelt. Het lijstje hier onder geeft aan welk bit welke betekenis heeft:

Naam	Bit
IntruderAlarm	0x01
PSUIsOK	0x02
TamperStatusGood	0x04
DoorContactClosed	0x08
AlarmTripped	0x10
DoorRelayEnergized	0x20

Event Types

Event types kunnen worden gevonden in de EventsEx view, alsmede in de 'ongevraagde' data die via de web socket wordt verstuurd.

Naam	Waarde
None	0
ControlUnitReset	1
SystemSetting	2
RealTimeClockSet	4
RealTimeClockError	5
PublicHoliday	6
ControlUnit	7
ChecksumFailure	8
AcuDiagnosticsAvailable	9
CardSwiped	10

DesktopReader	11
ValidCodeEntered	12
ValidPinEntered1	13
AccessPermitted1	15
AccessDenied1	16
AccessDenied2	17
AccessPermittedCardOnly	20
ValidPinEntered2	21
AccessPermittedCardCode	22
AccessDeniedInvalidCard	23
AccessDeniedInvalidPin	24
AccessDeniedInvalidCode	25
AccessPermittedPinOnly	26
AccessPermittedCodeOnly	27
DoorOpened1	28
DoorRelock	29
AccessPermittedAnpr	30
AccessDeniedAnpr	31
RequestAccessCheck	32
AcuApbCleared	33
Input5	34
Input6	35
Input7	36
Input8	37
Output1	38
Output2	39
Output3	40
Output4	41
Output5	42
Output6	43
Output7	44

Output8	45
DoorOpened2	46
DoorClosed	47
LockOpened	48
LockClosed	49
ReaderNotActive	50
DoorBellPressed	52
ReaderDetected	54
KeypadDetected	55
RelayToggled	60
DoorNotOpened	61
Relay1Toggled	62
Relay2Toggled	63
TriggerAndActionRuleWasRun	65
TimeAndAttendance	70
IntruderAlarm	75
FirmwareUpdated	80
SilenceAlarm	83
FireAlarmInput	85
FireDoor	86
PinNotValid	87
CardNotValid	88
KeypadTimeOut	89
Tamper	90
MainsFailed	91
DoorForced	92
DoorLeftOpen	93
ReaderTamper	94
DuressCode	95
KeypadHacker	96
InputShort	97

InputCut	98
InputVoltageSubstitution	99
InputPressed	100
InputReleased	101
AccessPermitted2	110
AccessDenied3	111
AccessDeniedLockownInProgress	112
DoorEnteredLockdownState	120
DoorExitedLockdownState	121
SetDoorOpenedTime	122
AcuNotResponding	500
AcuOnline	501
CheckCommunicationsInterface	502
IoBoardNotResponding	510
IoBoardOnline	511
IoBoardFirmwareUpdateSucceeded	512
IoBoardFirmwareUpdateFailed	513
CouldNotConnectToIoBoard	514
CouldNotConfigureIoBoard	515
IoBoardSettingsUpdated	516
LoggedOnAsOemClient	517
LoggedOffAsOemClient	518
Operator	550
AlarmActioned	551
BackupSucceeded	552
BackupFailed	553
ArchiveSucceeded	554
ArchiveFailed	555
Server	556
ServerError	557
AddAcu	558

AddAcuError	559
ServerWarning	560
SendMessageFailed	561
ObjectChangedLowPriority	568
Object	569
ObjectChanged	569
UserDetails	570
Timezone	571
AccessLevel	572
AcuConfiguration	573
Area	574
Report	575
Antipassback	576
SystemOperator	577
FirmwareUpgradeError	578
SiteGraphics	579
RequestDoorOpen	580
OpenDoorGroup	581
CloseDoorGroup	582
LockdownInitiatedByUser	590
LockdownEndedByUser	591
RemoteSite	600
EventTableCorrupted	700
FlashCrcChanged	701
EepromCrcChanged	702
InvalidSession	703
SessionStarted	704
SessionComplete	705
CompactStarted	706
CompactComplete	707
FactoryResetStarted	708

FactoryResetComplete	709
NanoDebugInformation	710
QueuedEventsLost	711
NanoCannotUnlock	720
NanoCannotLock	721
LowBattery	722
BatteryCriticallyLow	723
BatteryLevelUpdated	724

Event SubTypes

Event subtypes kunnen worden gevonden in de EventsEx view, alsmede in de 'ongevraagde' data die via de web socket wordt verstuurd.

Naam	Waarde
None	0
PowerOn	1
CopReset	2
InvalidInstruction	3
ClockFail	4
SoftwareInterrupt	5
RtcStop	6
RtcIcError	7
Initialised	8
PublicHolidayStart	10
PublicHolidayEnd	11
ProximityWakeUp	15
ScheduledWakeUp	16
AccessLevelNotValid	20
IndividualPermissionsNotValid	21
CardDetailsNotFound	22
CardDataAba	23
CardPaxtonUserCard	24
CardPaxtonNonUserCard	25
CardThirdParty	26
CardDataWiegand26	27
CardDataError	28
ClockIn	30
ClockOut	31
VehicleRegistrationNotRecognised	32
Opened	35
Closed	36

AntipassbackLogical	40
AntipassbackTimed	41
AntipassbackLogicalTimed	42
AntipassbackLostContactWithServer	43
NoAccessMade	45
GoneLow	50
GoneHigh	51
On	52
Off	53
WithTimezone	60
WithNetworkInstruction	61
WithExitButton	62
WithNet2DoorEntry	64
NotActive	70
UserOnHoliday	71
CardReportedLost	72
Armed	80
Disarmed	81
AlarmStillArmed	82
AnprAccessAttempt	110
DuringOfflineMode	120
LockdownInitiatedWithToken	131
LockdownEndedWithToken	132
LockDownStillActive	133
BreakReceived	500
Logon	550
Logoff	551
AlarmActionedBy	552
InvalidDirectory	560
StartStopError	569

Started	570
Stopped	571
ClientConnected	572
ClientDisconnected	573
DatabaseCreated	574
SoftwareUpdated	575
BackupRestored	578
AcuAddError	579
NewAcuAddedSuccessfully	580
TooManyDoors	581
ModificationError	589
Added	590
Modified	591
Deleted	592
FirmwareRefresh	598
Reinstated	599
ModemStatus	600
Connected	610
Disconnected	611
ConnectionFailed	612
ConnectionCancelled	613

Handleiding Net2JsonRestServer
Version 1.9