



INTOACCESS
THE INTEGRATORS



Net2JsonRestServer

Handleiding 1.20



Inhoudsopgave

Principe.....	6
Waarschuwing.....	6
Installatie.....	7
Configuratie.....	9
Net2 verbinding.....	11
REST Server instellingen.....	13
Geblokkeerde Aanroepen.....	15
Statische Bestand Instellingen.....	16
Server Scripting.....	18
Script instructies.....	19
ACU monitoring.....	20
Email instellingen.....	21
Webmail.....	21
SMTP.....	21
Email bestemming.....	21
Applicatie licentie.....	22
Service beheer.....	23
Log instellingen.....	24
SDK calls.....	25
Http response codes.....	25
Kaart typen.....	25
ActivateIOBoardRelay.....	26
Request.....	26
Reply.....	26
Voorbeeld met wget.....	26
AddAccessLevel.....	27
Request.....	27
Reply.....	27
Voorbeeld met wget.....	27
AddCard.....	29
Request.....	29
Reply.....	29
Voorbeeld met wget.....	29
AddDepartment.....	29
Request.....	30
Reply.....	30
Voorbeeld met wget.....	30
AddEventRecord.....	30
Request.....	30
Reply.....	31
Voorbeeld met wget.....	31
AddNewUser.....	31
Request.....	32
Reply.....	33





Voorbeeld met wget.....	33
AddTimezone.....	33
Request.....	34
Reply.....	34
Voorbeeld met wget.....	34
ControlDoorEx.....	35
Request.....	35
Reply.....	35
Voorbeeld met wget.....	36
DeleteAccessLevel.....	36
Request.....	36
Reply.....	36
Voorbeeld met wget.....	36
DeleteCard.....	37
Request.....	37
Reply.....	37
Voorbeeld met wget.....	37
DeleteDepartment.....	37
Request.....	37
Reply.....	38
Voorbeeld met wget.....	38
DeleteTimezone.....	38
Request.....	38
Reply.....	38
Voorbeeld met wget.....	38
EndLockdown.....	39
Request.....	39
Reply.....	39
Voorbeeld met wget.....	39
GetApplicationSetting.....	39
Request.....	39
Reply.....	40
Voorbeeld met wget.....	40
GetListOfOperators.....	40
Request.....	40
Reply.....	40
Voorbeeld met wget.....	40
GetUserImage.....	41
Request.....	41
Reply.....	41
Voorbeeld met wget.....	41
GetServerVersion.....	41
Request.....	41
Reply.....	41
Voorbeeld met wget.....	42





InitiateLockdown.....	42
Request.....	42
Reply.....	42
Voorbeeld met wget.....	42
LastErrorMessage.....	42
Request.....	43
Reply.....	43
Voorbeeld met wget.....	43
PurgeUser.....	43
Request.....	43
Reply.....	43
Voorbeeld met wget.....	44
QueryDb.....	44
Request.....	44
Reply.....	44
Voorbeeld met wget.....	44
RevokeUserTemporaryAccess.....	45
Request.....	45
Reply.....	45
Voorbeeld met wget.....	45
SendEmail.....	45
Request.....	45
Reply.....	46
Voorbeeld met wget.....	46
SetApplicationSetting.....	46
Request.....	46
Reply.....	46
Voorbeeld met wget.....	47
SetUserTemporaryAccess.....	47
Request.....	47
Reply.....	47
Voorbeeld met wget.....	47
UpdateAccessLevel.....	48
Request.....	48
Reply.....	48
Voorbeeld met wget.....	48
UpdateCard.....	49
Request.....	49
Reply.....	49
Sample with wget.....	50
UpdateDepartment.....	50
Request.....	50
Reply.....	50
Voorbeeld met wget.....	50
UpdateTimezone.....	51





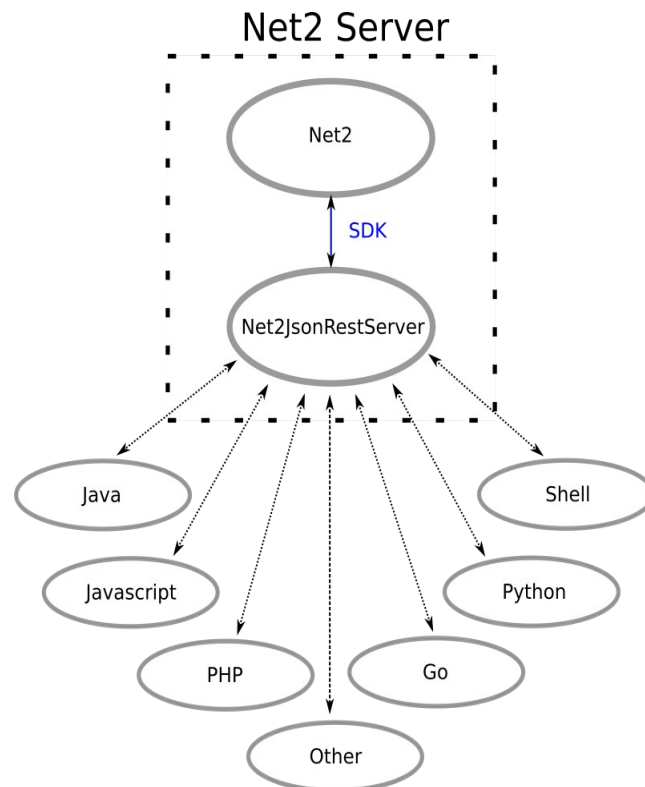
Request.....	51
Reply.....	51
Voorbeeld met wget.....	52
UpdateUserRecord.....	52
Request.....	53
Reply.....	54
Voorbeeld met wget.....	54
ValidateOperator.....	54
Request.....	54
Reply.....	55
Voorbeeld met wget.....	55
ViewIOBoardInputs.....	55
Request.....	55
Reply.....	55
Voorbeeld met wget.....	56
ViewIOBoardOutputs.....	56
Request.....	56
Reply.....	56
Voorbeeld met wget.....	56
ViewUserRecords.....	57
Request.....	57
Reply.....	57
Voorbeeld met wget.....	58
Ongevraagde data.....	59
Url.....	59
Gebeurtenis data structuur.....	59
Eenvoudig node.js code voorbeeld.....	60
Aanvullende code voorbeelden.....	61
Node-RED integratie.....	62
Ontvangen ongevraagde data.....	62
Ophalen data.....	62
Voorbeeld flow.....	64
Numerieke waarden gebruikt door de SDK.....	65
Device status flag.....	65
Event Types.....	65
Event SubTypes.....	71



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

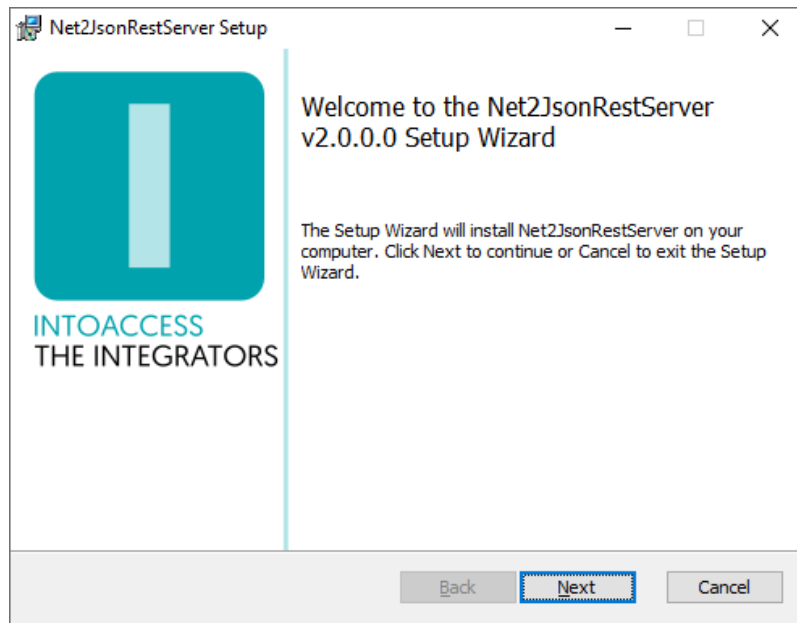
Waarschuwing

De naamgeving van methode parameters of kolommen binnen de dotnet Paxton SDK / DB, is niet altijd consequent. Deze wrapper volgt deze naamgeving en is hierdoor dus ook niet helemaal consequent. Let daarom goed op, met name bij het lezen & schrijven van persoonsgegevens, welke (attribuut)namen er meegegeven moeten worden.

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.



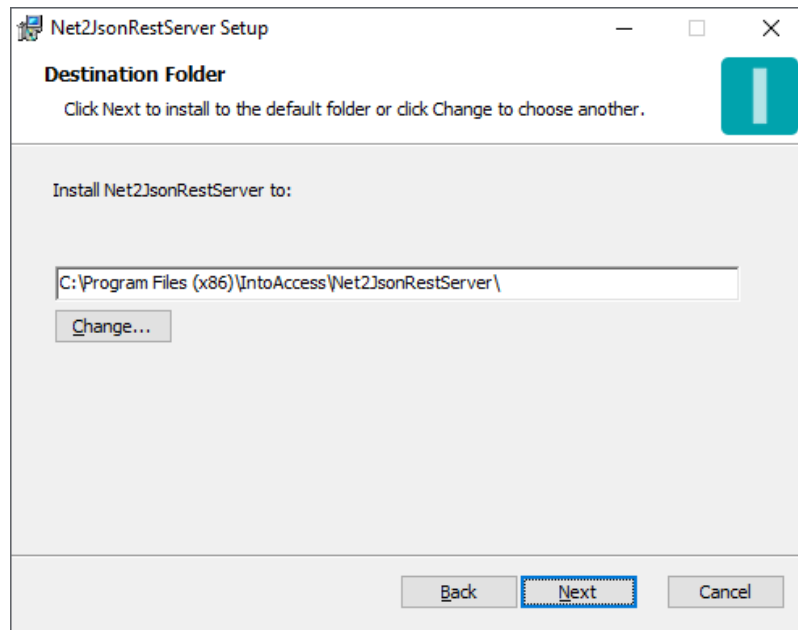
Afbeelding 2

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

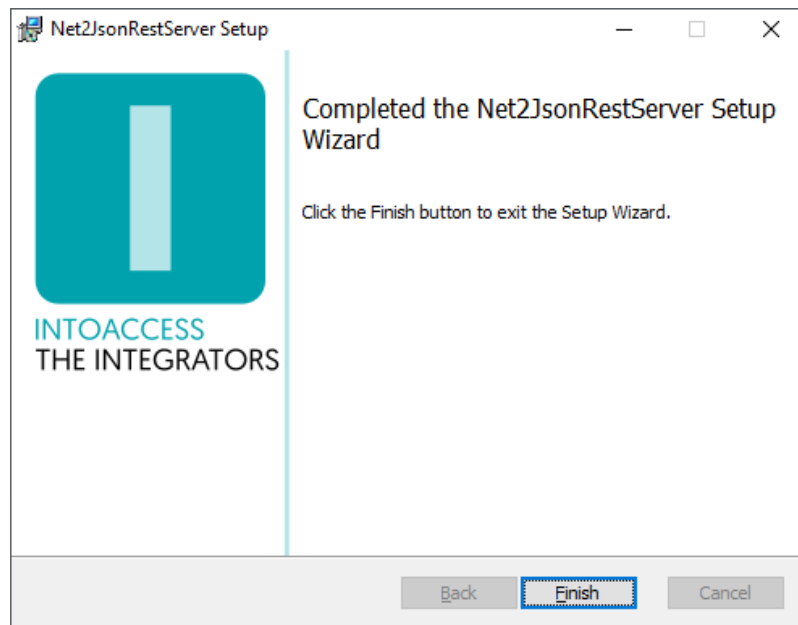
Hoewel een nieuwere versie een eventueel al aanwezige oudere versie zelf zou moeten vervangen, kun u voor de zekerheid ook eerst de bestaande applicatie de-installeren. De reeds bestaande configuratie instellingen worden daarbij niet verwijderd, dus na installatie van de nieuwe versie zou alles weer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





Afbeelding 3

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



Afbeelding 4

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;

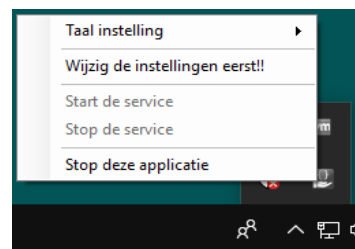
Bij het starten van de manager applicatie wordt een splash-screen getoond, waarna de applicatie onder normale omstandigheden verkleint tot een 'tray icon' in de rechter onder hoek van het scherm.



Afbeelding 5



Afbeelding 6



Afbeelding 7

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.



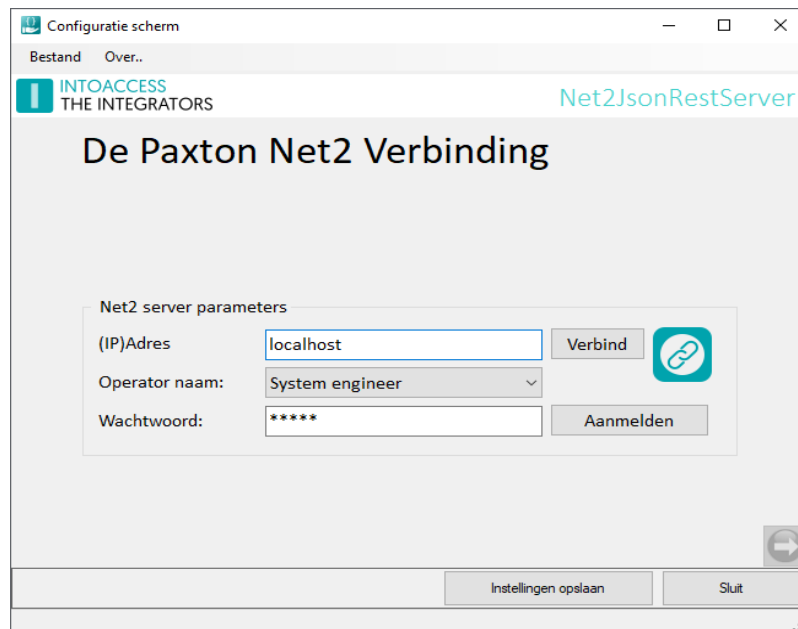


De kleur van het icoon is overigens een indicatie van of de service draait. Wanneer de service (nog) niet draait, is de kleur grijs; wanneer de service draait, is het icoon gekleurd.



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.

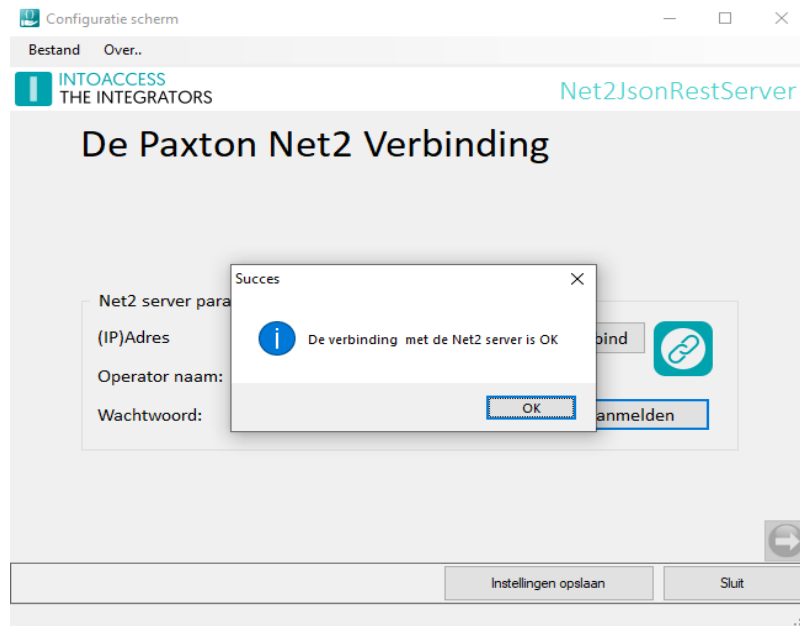


Afbeelding 8

- 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.



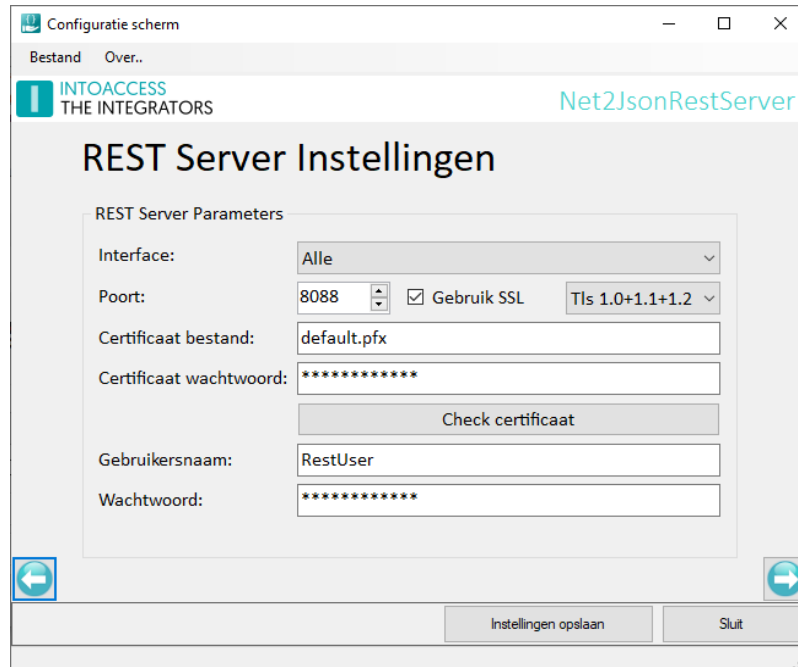


Afbeelding 9

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 10

- **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/>



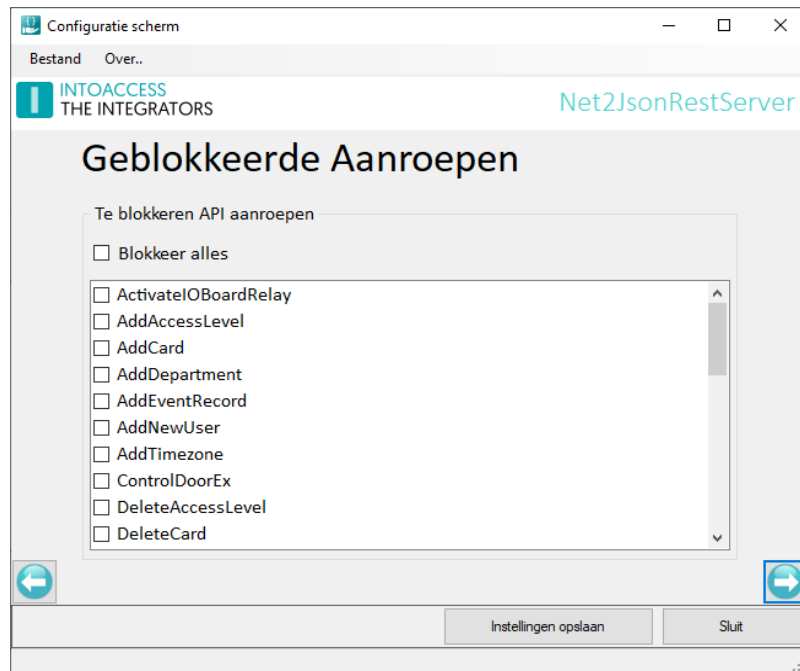


- **Certificaat wachtwoord:**
Om een PFX type certificaat bestand te kunnen laden, is doorgaans een wachtwoord nodig, welke u hier kunt opgeven.
Door op de knop “Check certificaat” te drukken, kunt u controleren of de applicatie het certificaat bestand correct kan interpreteren.
- **Gebruikersnaam:**
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”.



Geblokkeerde Aanroepen

Indien veiligheidsoverwegingen hier om vragen, kunt u API aanroepen die u niet beschikbaar wilt stellen blokkeren.



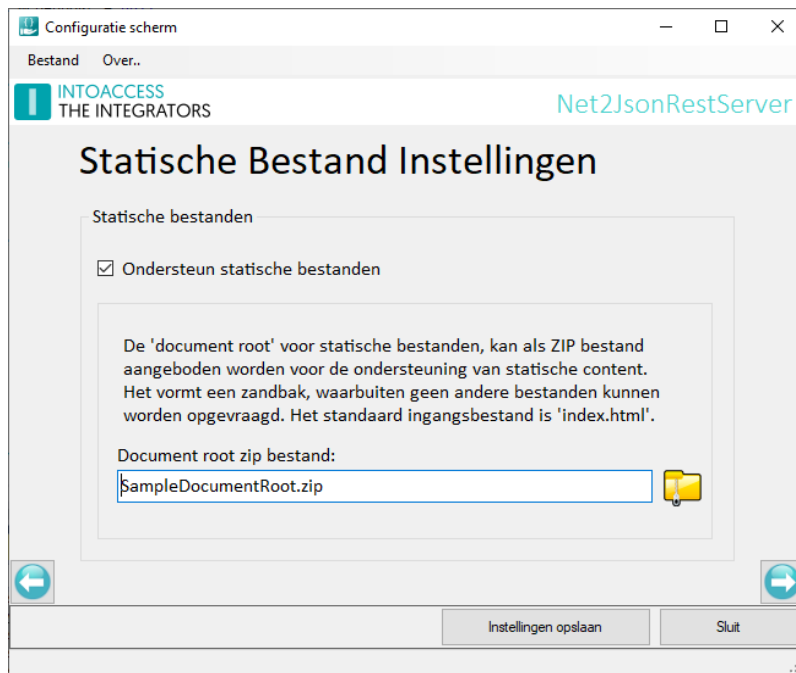
Afbeelding 11

Alle gemarkeerde aanroepen in deze lijst, zullen een 404 Niet gevonden HTTP code teruggeven wanneer (toch) gebruikt.



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 implementatie. 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.





The screenshot shows a web interface titled "ControlDoorEx". It contains the following elements:

- ACU: 12345678
- Relay: Relay 1 (dropdown menu)
- Function: Timed open (dropdown menu)
- Open time [sec]: 7
- LED flash: Both readers (dropdown menu)
- Open door button
- Log out button (with a right-pointing arrow icon)

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.



Server Scripting

De Net2JsonRestServer kan ook fungeren als een hulpprogramma om achtergrond scripts te kunnen draaien, die continue actief zijn. Indien ingeschakeld, zal het uw script starten, tezamen met de REST service, zodat het script automatisch wordt opgestart en kan communiceren met Net2.

Op dit moment ondersteunt de applicatie de Node.js de Python script engines voor server scripts. Om deze te kunnen gebruiken, download en installeer Node.js of Python voor 64-bit windows, op dezelfde PC als waar de Net2JsonRestServer is geïnstalleerd. Zorg er voor dat de script engine gevonden kan worden via het systeem zoek pad (doorgaans expliciet aan te vinken voor Python).

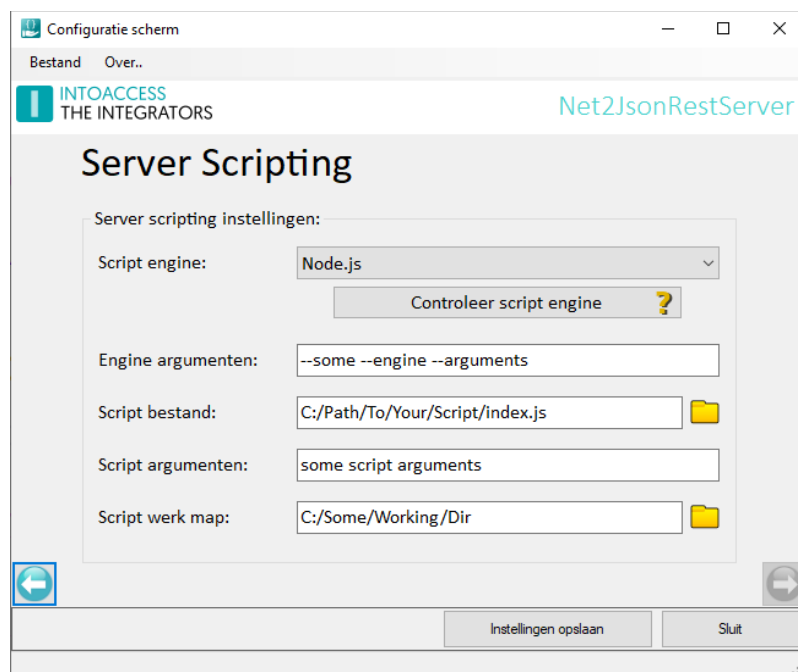


Image 14

Om te controleren of de script engine die u wilt gebruiken aanwezig is op het systeem, klik op de "Controleer script engine" knop en als alles goed is, zal dan de versie informatie worden getoond. Noot: indien u de script engine pas installeert nadat de manager applicatie al draait, kan het nodig zijn om de manager eerst geheel te stoppen en weer op te starten, voordat de controle knop een positief resultaat geeft.

Voor (script) engines anders dan Node.js of Python (zoals Ruby, Rust of Java), kunt u een custom engine definiëren, door een naam/pad op te geven van deze engine. De support hiervoor is beperkt, maar zou in de meeste gevallen goed moeten werken.

- De script engine argumenten zijn optioneel en de mogelijke waarden zijn afhankelijk van de gekozen script engine.
- Het script bestand is verplicht en moet het start script zijn van uw applicatie.





- De script argumenten zijn optioneel en kunnen worden gelezen door uw script.
- De script werk map is ook optioneel en zal (indien blanco gelaten) gelijk zijn aan de werk map van de Net2JsonRestServer.

Script instructies

Daar u (waarschijnlijk) uw achtergrond script blijvend wilt laten draaien, zal u het script zo moeten maken dat het in een oneindige lus draait of door een asynchrone interval timer periodiek de te herhalen script code laat aanroepen (en zo voorkomt dat het script eindigt).

Als uw script eindigt, zal het niet automatisch weer worden opgestart, dus zorg er daarom voor dat alle niet fatale fouten die kunnen optreden worden afgevangen.

Node.js

- Recente Node.js installatiebestanden, zullen Node.js installeren in een directory die kan worden gelezen door alle gebruikers en zal een pad toevoegen naar deze script engine in de systeem PATH variabele.
- Om uw script output in het logbestand van de Net2JsonRestServer te laten verschijnen, kunt u eenvoudig “console.log(xyz)” gebruiken. Fouten / stack traces zullen automatisch worden opgepikt vanaf “stderr”.

Python

- Bij het installeren van Python, zorg er voor dat deze wordt geïnstalleerd onder [C:\Program Files](#) en niet in een gebruiker specifieke map. Zorg er tevens voor dat de script engine kan worden gevonden in het systeem zoek PATH. Voor recente Python3 installatie programma's, selecteer daarvoor de volgende opties:
 - 1^{ste} scherm: vink “Add Python x.y to PATH” aan;
 - Klik op “Customize installation”;
 - 2^{de} scherm: optioneel selecteer alleen pip (de rest is niet essentieel);
 - 3^{de} scherm: “Install for all users” (dit wijzigt de installatie locatie);
 - Klik op de Install knop.
- Om uw script output in het logbestand van de Net2JsonRestServer te laten verschijnen, kunt u gebruik maken van het standaard Python print commando. Er moet echter wel expliciet ge'flushed' worden om het goed te laten werken. Voor Python2 kan die middels een extra “sys.stdout.flush()” aanroep, maar Python3 biedt daar een extra print optie voor: “print(xyz, flush=True)”.



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.

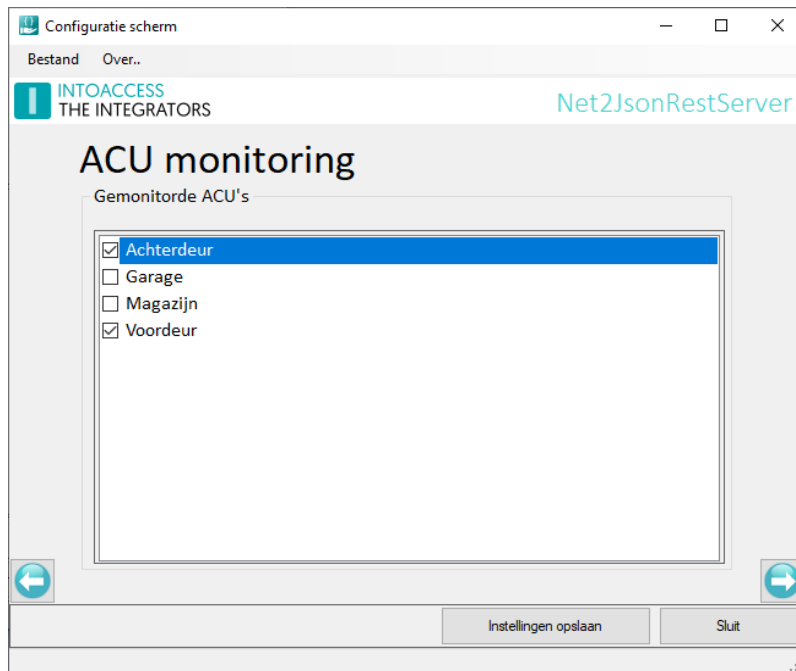


Image 15

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

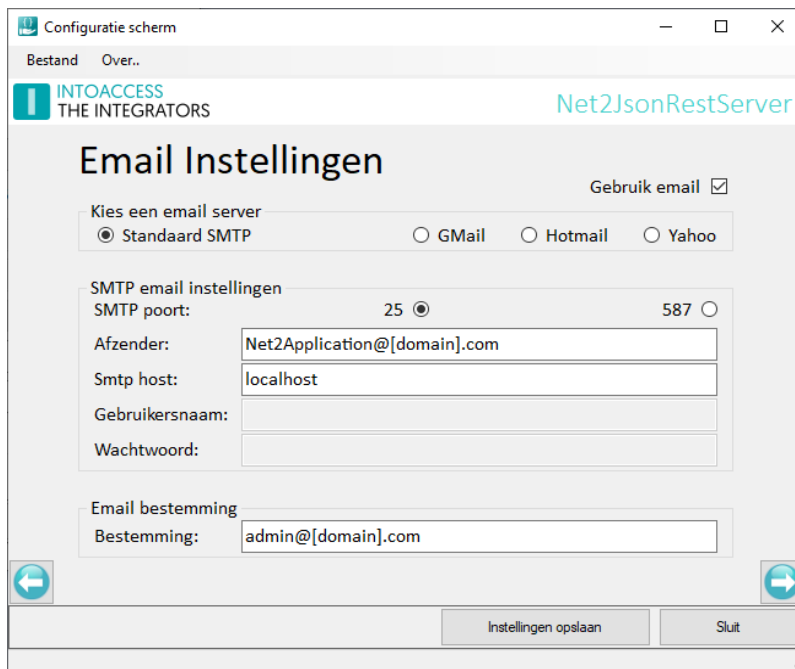
Voor het gebruik van webmail providers (zoals Gmail), kan het noodzakelijk zijn om de beveiliging te verlagen (provider account setting).

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.

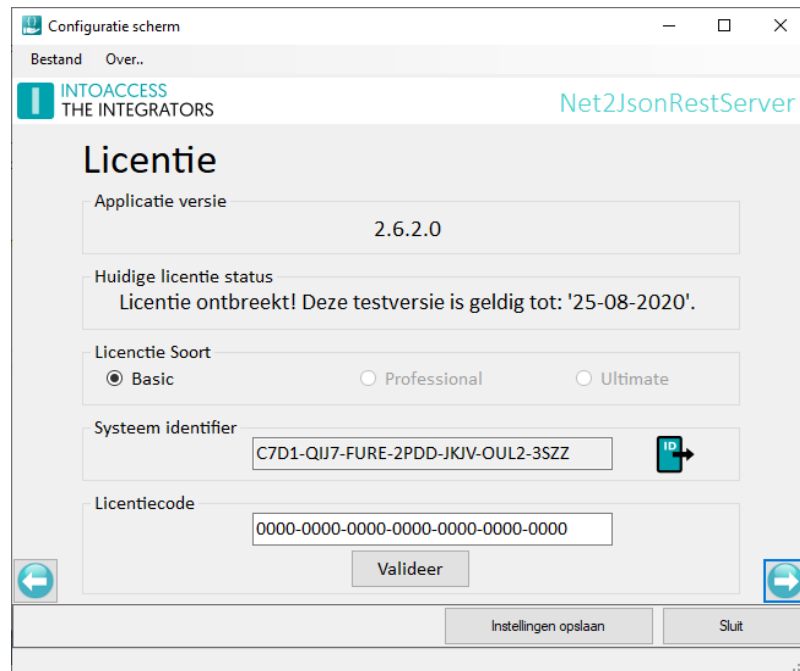


Afbeelding 16



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 17

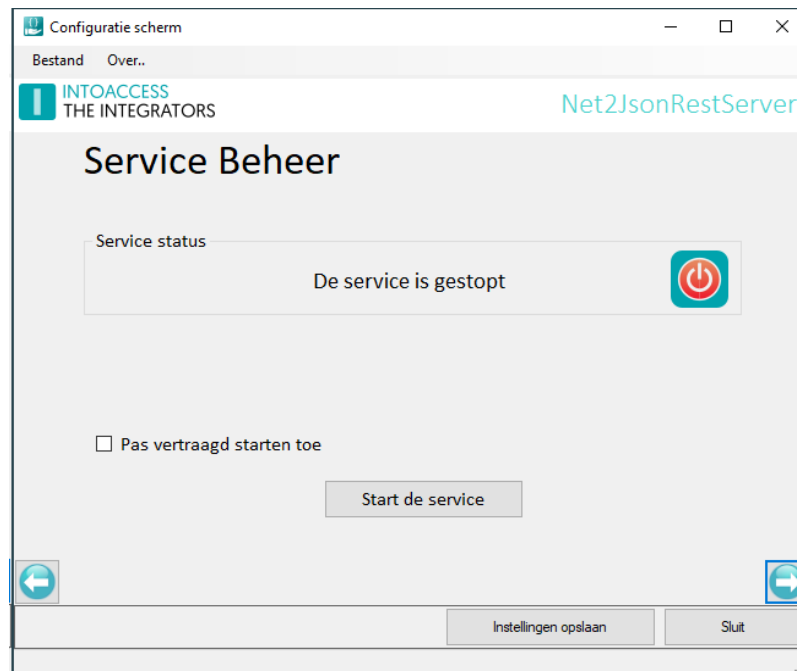
Na aanschaf van een licentie, kunt u het “Systeem identifier” export bestand aanmaken (druk op icoon) en emailen aan IntoAccess (info@intoaccess.com), waarna u de licentiecode ontvangt.

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 18

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

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

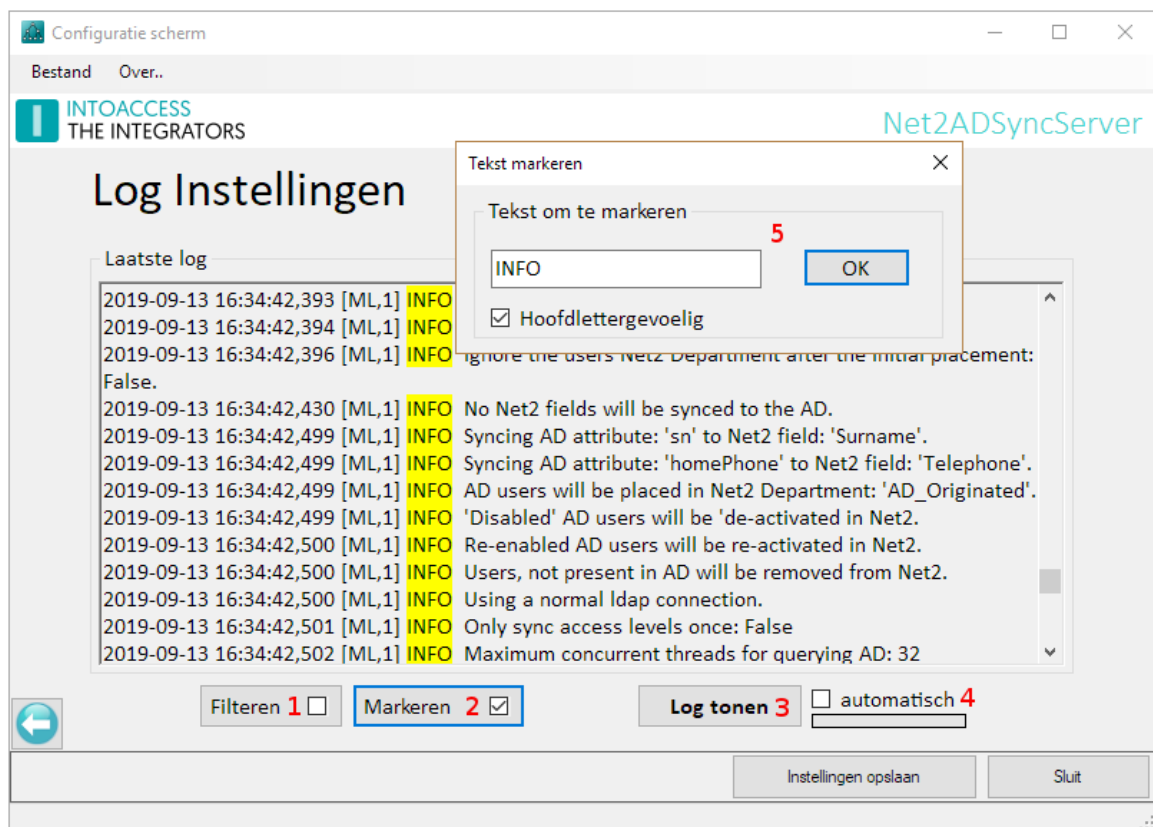
Standaard heeft de Net2JsonRestServer een service-afhankelijkheid van Net2, die er voor zorgt dat hij pas start wanneer Net2 al draait. Deze afhankelijkheid kan wat problemen veroorzaken bij sommige Net2 tools, die de Net2 service pogen te herstarten. Het tijdelijk uitschakelen van de Net2JsonRestServer is normaal voldoende om de tools zonder problemen te gebruiken. Wanneer dit voor u onhandig is, kunt u de optie Pas vertraagd starten toe aanvinken, waardoor de Net2 afhankelijkheid wordt verwijderd en vervangen door een 'vertraagde' start. Let op dat de PC moet worden herstart om het 'vertraagde' start type in de Windows service manager te zien te krijgen. Tevens is het zo dat deze instelling alleen relevant is wanneer de service draait op dezelfde PC als Net2.



Log instellingen

Deze pagina, zie afbeelding 22, 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 19

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





ActivateIOBoardRelay

Stuur een Paxton IO-board relais contact aan.

Ter kennisgeving:

- De ioBoardId kan middels QueryDB worden opgevraagd. (select * from sdk.IOBoards)
- De relayIndex waarde is 0-based (0 = relais 1, 1 = relais 2, etc)
- De relayOpenDuration is een waarde in seconden. Gebruik de waarde -1 voor permanente sturing.
- De isEnergized waarde bepaalt of het relais word in- (true) of uit- (false) geschakeld.

Request

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

JSON

```
{  
  "ioBoardId": 6,  
  "relayIndex": 0,  
  "relayOpenDuration": -1,  
  "isEnergized": false  
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \  
--post-data='{  
"ioBoardId": 6,  
"relayIndex": 0,  
"relayOpenDuration": -1,  
"isEnergized": true  
}' \  
--header='Content-Type:application/json' \  
"http://$SERVER:$PORT/ActivateIOBoardRelay"
```





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 "1-99999999" 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\"
```

AddEventRecord

Voeg een nieuwe log regel aan de Paxton log toe.

Ter kennisgeving:

* Het wijzigen van een geschreven log regel is niet mogelijk

Request

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

JSON

```
{  
"eventType": 20,  
"eventSubType": 0,  
"address": 1156774,  
"subAddress": 0,  
"userId": "7403",  
"cardNo": "12345678",  
"eventDetail": "whatever",  
"linkedEventId": 0,
```





```
"ioBoardId": 0,  
"inputId": 0,  
"outputId": 0  
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \  
--post-data='{  
"eventType": 20,  
"eventSubType": 0,  
"address": 1156774,  
"subAddress": 0,  
"userId": "7403",  
"cardNo": "12345678",  
"eventDetail": "whatever",  
"linkedEventId": 0,  
"ioBoardId": 0,  
"inputId": 0,  
"outputId": 0  
}' \  
--header='Content-Type:application/json' \  
"http://$SERVER:$PORT/AddEventRecord"
```

AddNewUser

Voeg een nieuwe gebruiker toe.

Ter kennisgeving:

- Zet active op true, anders is de gebruiker niet geldig/zichtbaar.
- 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.
- Als 'Geavanceerde bevoegdheden' zijn ingeschakeld, kan in plaats van het attribuut 'accessLevelId' het attribuut 'accessLevels' worden gebruikt, met een array van id's. Let hierbij op de beperking van Net2, dat 'overlappende' autorisaties niet kunnen worden gecombineerd.

Request

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

JSON

```
{
  "accessLevelId": 1,    (of "accessLevels": [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 of meerdere deuren tegelijk.

Parameters:

- address: ACU serie nummer **of** addresses: Array van ACU serie nummers
- relais: 0 = relais 1, 1 = relais 2
- 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,      (or "addresses": [123])  
  "relais": 0,  
  "function": 1,  
  "doorOpenTime": 3000,  
  "ledFlash": 3  
}
```

Reply

True bij succes, false bij een fout.

Noot: Deze functie geeft ook een true terug bij ongeldige ACU adressen

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"
```

GetApplicationSetting

Geeft een array or een base64 string retour, met de waarde van de opgegeven 'application setting'.

- Als de 'setting' niet bestaat, wordt een null terug gegeven.
- Noot: het is niet mogelijk een 'setting' te verwijderen; alleen leeg maken.

Request

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

JSON

```
{  
  "applicationName": "SomeAppName",  
  "settingName": "SomeSettingName",  
  "userName": "SomeUserName",      (bij weglaten "SYSTEM")  
  "asBase64": true                  (bij weglaten false)  
}
```





Reply

De inhoud van de gevraagde 'application setting' als een array van integer waarden of een base64 string.

JSON

[1,2,3] of "AQID" (afhankelijk van de asBase64 waarde in de aanvraag)

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \  
--post-data='{  
"applicationName": "SomeAppName",  
"settingName": "SomeSettingName",  
"userName": "SomeUserName",  
"asBase64": false  
}' \  
--header='Content-Type:application/json' \  
"http://$SERVER:$PORT/GetApplicationSetting"
```

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"
```





```
"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/4AAQSkZJRgABAQAAQABAAD//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"
```

RevokeUserTemporaryAccess

Trek een tijdelijke autorisatie van een gebruiker in.

Ter kennisgeving:

- De parameter is de gebruiker id.

Request

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

JSON

18

Reply

True bij succes, false bij een fout.

JSON

true

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \  
--post-data='18' \  
--header='Content-Type:application/json' \  
"http://$SERVER:$PORT/RevokeUserTemporaryAccess"
```

SendEmail

Verstuur een email via Net2JsonRestServer.

- Er dient wel een email configuratie gemaakt te zijn aan de server zijde.
- De afzender en ontvanger zijn gedefinieerd aan de server zijde.

Request

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

JSON

{





```
"msgText": "Test in plain text",  
"msgHtml": "<h1>Test in <u><i>HTML</i></u></h1>",  
"subject": "Test subject"  
}
```

Reply

True bij succes, false bij een fout.

JSON

```
true
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \  
--post-data='{  
"msgText": "Test in plain text",  
"msgHtml": "<h1>Test in <u><i>HTML</i></u></h1>",  
"subject": "Test subject"  
}' \  
--header='Content-Type:application/json' \  
"http://$SERVER:$PORT/SendEmail"
```

SetApplicationSetting

Zet een nieuwe of update een bestaande 'application setting'.

- Als een null waarde wordt aangeboden, zal de 'setting' niet geschreven/gewijzigd worden.

Request

```
url: "http://<ip>:<port>/SetApplicationSetting"
```

JSON

```
{  
  "applicationName": "SomeAppName",  
  "settingName": "SomeSettingName",  
  "userName": "SomeUserName",      (if left out "SYSTEM")  
  "asBase64": true                  (if left out, its value is false)  
  "value": [1,2,3],                 (set this when asBase64 is false)  
  "base64Value": "AQID",           (set this when asBase64 is true)  
}
```

Reply

True bij succes, false bij een fout.





JSON

true

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \  
--post-data='{  
"applicationName": "SomeApplicationName",  
"settingName": "SomeSettingName",  
"userName": "SomeUserName",  
"value": [1, 2, 3]  
}' \  
--header='Content-Type:application/json' \  
"http://$SERVER:$PORT/SetApplicationSetting"
```

SetUserTemporaryAccess

Geef een gebruiker een tijdelijke autorisatie.

Ter kennisgeving:

- U kunt ook een update van een eerdere setting uitvoeren met deze aanroep.

Request

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

JSON

```
{  
  "userId": 18,  
  "accessLevelId": 1,  
  "expiryDateTime": "2023-01-30 12:00" (indien voor 'nu': fout)  
}
```

Reply

True bij succes, false bij een fout.

JSON

true

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \  
--post-data='{  
"userId": 18,  
"accessLevelId": 1,
```





```
"expiryDateTime": "2023-01-20 12:00"  
}' \  
--header='Content-Type:application/json' \  
"http://$SERVER:$PORT/SetUserTemporaryAccess"
```

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 "1-99999999" zijn of een kenteken.
- De cardTypeld 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:

- Zet activeInd op true, anders is de gebruiker niet geldig/zichtbaar.
- 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. (zet de waarde op een lege string om een pincode te verwijderen)
- 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.





- Als 'Geavanceerde bevoegdheden' zijn ingeschakeld, kan in plaats van het attribuut 'accessLevelId' het attribuut 'accessLevels' worden gebruikt, met een array van id's. Let hierbij op de beperking van Net2, dat 'overlappende' autorisaties niet kunnen worden gecombineerd.

Request

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

JSON

```
{
  "userId": 123,
  "accessLevelId": 1, (of "accessLevels": [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"
```

ViewIOBoardInputs

Haal de input status op van alle inputs (4) van een gegeven Paxton IO-board.

Ter kennisgeving:

- De parameter is de IO-board id, welke kan worden opgevraagd middels QueryDB (select * from sdk.IOBoards).

Request

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

JSON

```
6
```

Reply

JSON

```
[
```





```
{ "InputID": 21, "Name": "Input 1", "IsPressed": false },  
{ "InputID": 22, "Name": "Input 2", "IsPressed": false },  
{ "InputID": 23, "Name": "Input 3", "IsPressed": false },  
{ "InputID": 24, "Name": "Input 4", "IsPressed": false }  
]
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \  
--post-data='6' \  
--header='Content-Type:application/json' \  
"http://$SERVER:$PORT/ViewIOBoardInputs"
```

ViewIOBoardOutputs

Haal de output status op van alle outputs (4) van een gegeven Paxton IO-board.

Ter kennisgeving:

- De parameter is de IO-board id, welke kan worden opgevraagd middels QueryDB (select * from sdk.IOBoards).
- De status waarde van outputs is niet altijd even betrouwbaar.

Request

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

JSON

6

Reply

JSON

```
[  
  { "OutputID": 21, "Name": "Output 1", "IsEnergised": true },  
  { "OutputID": 22, "Name": "Output 2", "IsEnergised": false },  
  { "OutputID": 23, "Name": "Output 3", "IsEnergised": false },  
  { "OutputID": 24, "Name": "Output 4", "IsEnergised": false }  
]
```

Voorbeeld met wget

```
wget -O- --user=$USER --password=$PASSWORD \  
--post-data='6' \  
--header='Content-Type:application/json' \  
"http://$SERVER:$PORT/ViewIOBoardOutputs"
```





```
"http://$SERVER:$PORT/ViewIOBoardOutputs"
```

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.

Ter kennisgeving:

- Deze methode is niet geschikt om de kaarten van een gebruiker op te vragen. Gebruik in plaats daarvan QueryDb.

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 (1-99999999)





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');  
}  
});
```

Aanvullende code voorbeelden

U kunt aanvullende code voorbeelden vinden in de volgende zip file:

https://www.intoaccess.com/downloads/Net2JsonRestServer_CodeSamples.zip





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:

The screenshot shows the 'Properties' panel for a 'Connect to' component in Node-RED. The 'Type' is set to 'Connect to'. The 'URL' is set to 'ws://RestUser:RestPassword@192.168.1.55:8088/Events'. The 'Name' is set to 'Paxton'. There are three icons in the top right corner: a gear, a document, and a refresh icon.

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.





The screenshot shows the 'Properties' window of a component. The 'Name' field is set to 'Name'. Below the name field are three tabs: 'Setup', 'Function', and 'Close'. The 'Function' tab is active, displaying a code editor with the following JavaScript code:

```
1 // Construct query
2 const query = `select * from UsersEx where userId=${msg.payload.userId}`;
3 return {
4   payload: JSON.stringify(query)
5 };
```

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

The screenshot shows the 'Properties' window of an HTTP request component. The 'Method' is set to 'POST'. The 'URL' is set to 'http://RestUser:RestPassword@192.168.1.55:8088/QueryDB'. There are several unchecked checkboxes: 'Enable secure (SSL/TLS) connection', 'Use authentication', 'Enable connection keep-alive', and 'Use proxy'. The 'Return' type is set to 'a parsed JSON object'. The 'Name' field is set to 'QueryDB'. A yellow tip box at the bottom states: 'Tip: If the JSON parse fails the fetched string is returned as-is.'

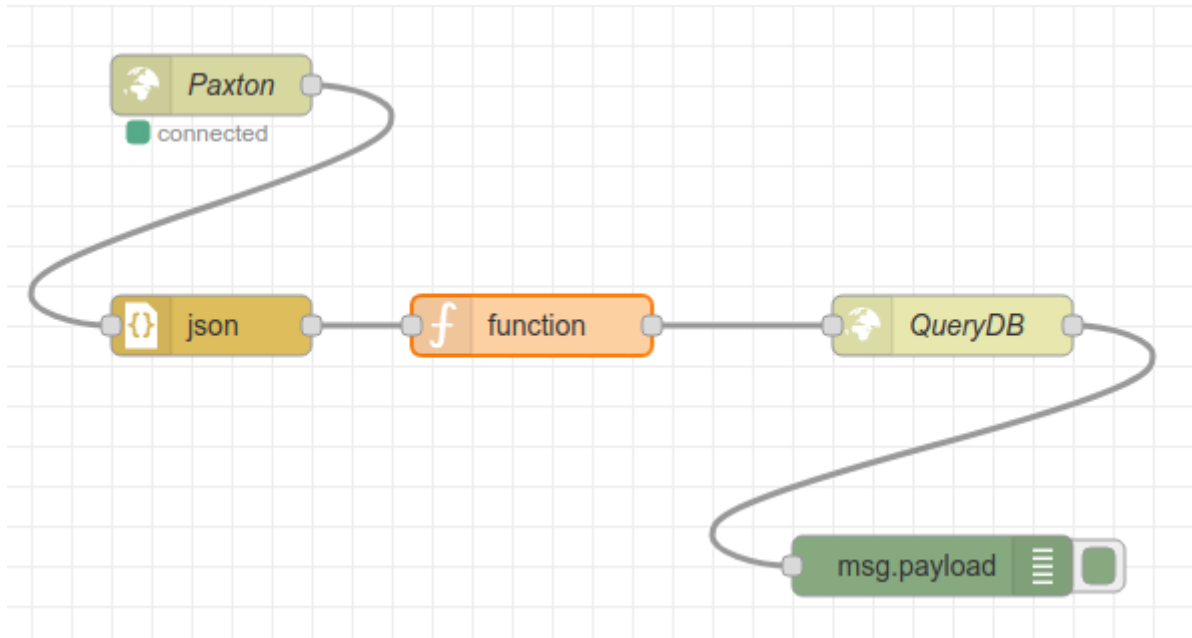
Noot: u kunt in principe elke Net2JsonRest methode aanroepen op dezelfde wijze, zoals bijvoorbeeld het openen van een deur.





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 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
AccessDeniedLockdownInProgress	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
Versie 1.20

