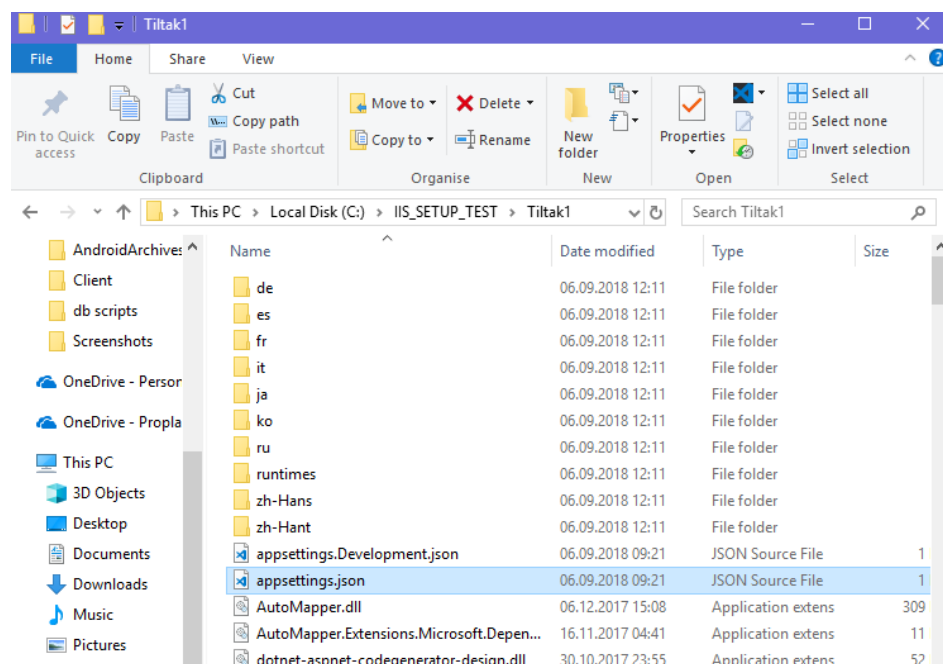


# Oppsett - Tiltak API

versjon 1.1

## Generelt oppsett, filer og settings

Før vi kan begynne med oppsett av IIS, ligg web-filene (Tiltak API) i en mappe på systemdisken. Dersom disse mottas som en zippet fil, må denne først pakkes ut. For å verifisere at dette er gjort korrekt, gå inn i mappa og se etter «appsettings.json». Dersom denne ligger i «rota» av mappa er alt slik det skal være.



## Connection strings

Åpne opp filen «appsettings.json».

Under «ConnectionStrings» (i toppen av fila) må du fylle ut 2stk connection string for APIet.

1. Klient-database (DefaultConnection)
2. System-database (SystemConnection)

OBS: Dersom det blir brukt backslashes (“\”) i connectionstringen må disse “escapes”. Dette gjør man med å legge til enda en backslash. “\” => “\\”.

For å gi et konkret eksempel: sql-server med instanser er ofte brukt backslashes i; “SERVERNAVN\INSTANS“. For at dette skal fungere ved runtime må man altså skrive “SERVERNAVN\\INSTANS”.

# Database endringer (HangFire)

APIlet tar i bruk et rammeverk for å kjøre planlagte oppgaver. I den forbindelse så oppretter rammeverket noen ny tabeller første gang det kjøres mot databasen, og eventuelt under en senere oppdatering. Vi bruker i dag dette for å periodisk sjekke lisensen til tiltak, for å forsikre oss om at installasjonen fortsatt har en gyldig lisens. Databasebrukeren til APIlet trenger rettigheter til å opprette tabeller i databasen første gang det kjøres, og etter en oppdatering. Disse rettighetene bør fjernes igjen etter at dette er gjort, da det ikke normalt er behov for dette.

Gi rettigheter til riktig bruker på database:

```
USE TiltakDatabase;  
GRANT CREATE TABLE TO ApiDatabaseBruker;  
GO
```

Fjern rettigheter:

```
USE TiltakDatabase;  
REVOKE CREATE TABLE TO ApiDatabaseBruker;  
GO
```

## Brukerautentisering

APIlet bruker en form for brukerautentisering som kalles JWT, for at dette skal være sikkert må alle installasjoner ha unike verdier i feltene:

- Tokens:Key
- Tokens:Issuer

```
appsettings.json  
1 {  
2   "ConnectionStrings": {  
3     "DefaultConnection": "Data Source=XX.XX.XXX.XXX;Encrypt=yes;TrustServerCertificate=yes;  
4     "SystemConnection": "Data Source=XX.XX.XXX.XXX;Encrypt=yes;TrustServerCertificate=yes;I  
5   },  
6   "Logging": {  
7     "IncludeScopes": false,  
8     "Debug": {  
9       "LogLevel": {  
10      | "Default": "Warning"  
11    }  
12  },  
13  "Console": {  
14    "LogLevel": {  
15      "Default": "Warning"  
16    }  
17  }  
18 },  
19 "Tokens": {  
20   "Key": "ABCDEF012345678987654321", //REPLACE WITH YOUR SITES UNIQUE CODE  
21   "Issuer": "http://proplanutvikling.no", //REPLACE WITH YOUR DOMAIN  
22   "ExpirationHours": 5 //NUMBER OF HOURS THE TOKEN IS VALID  
23 }  
24 }
```

«Key» anbefales generert via [denne](#) kodegeneratoren, men kan i teorien være «hva som helst». Den må være minimum 128bits.

Dersom kode mot all formodning skulle bli endret ved en senere anledning, vill alle utstedte tokens ugyldiggjøres – og påloggede brukere vil måtte logge inn på nytt.

«Issuer» settes til kundens eget domene (NB: trenger ikke å være det samme domenet som for selve APIet, det må bare ikke være i bruk av andre Tiltak API).

**OBS: Dersom to løsninger har samme “Issuer”- og “Key”- kombinasjon vil påloggings-”Token” være gyldig i begge løsninger - uavhengig av hvor den kommer fra!**

Eksempelvis blir det seende slik ut etter man har generert opp en sikker nøkkel og satt «Issuer»:

```
"Tokens": {  
  "Key": "qvjHkRB64Av580kFlozuwKRjFkXunN8g",  
  "Issuer": "https://kundensdomene.no",  
  "ExpirationHours": 5  
}
```

# Internet Information Services (IIS)

## Innallasjon/oppsett, IIS

For at APllet skal kunne kjøre på IIS trenger man runtime for "aspnetcore". Fullt oppsett og alt av *prerequisites* står godt beskrevet på denne linken (men jeg legger ved et utsnitt av det viktigste under):

<https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/iis/?view=aspnetcore-2.0>

Dersom IIS (Web Server) ikke er aktivert i servermiljøet er det en forklaring på hvordan dette settes opp [HER](#).

**OBS: IIS må installeres før ".Net Core Hosting Bundle", ellers må ".Net Core"-installeren kjøres på nytt i etterkant også!**

Kortversjon (dersom bare "runtime" mangler):

### Install the .NET Core Hosting Bundle

1. Install the *.NET Core Hosting Bundle* on the hosting system. The bundle installs the .NET Core Runtime, .NET Core Library, and the [ASP.NET Core Module](#). The module creates the reverse proxy between IIS and the Kestrel server. **If the system doesn't have an Internet connection, obtain and install the [Microsoft Visual C++ 2015 Redistributable](#) before installing the .NET Core Hosting Bundle.**
  - a. Navigate to the [.NET downloads page](#).
  - b. Under **.NET Core**, select the **Download .NET Core Runtime** button next to the **Run Apps** label. The installer's executable contains the word "hosting" in the file name (for example, *dotnet-hosting-2.1.2-win.exe*).
  - c. Run the installer on the server.

**Important!** *If the Hosting Bundle is installed before IIS, the bundle installation must be repaired. Run the Hosting Bundle installer again after installing IIS.*

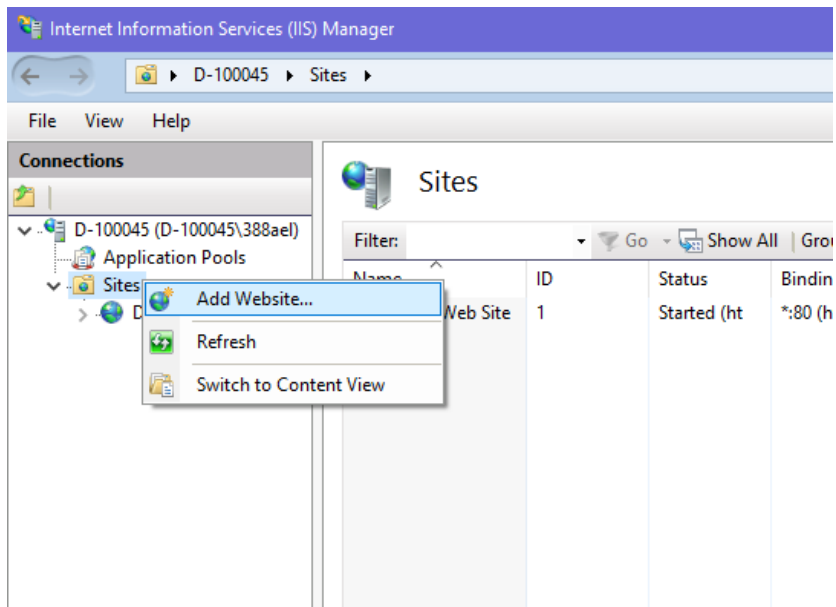
To prevent the installer from installing x86 packages on an x64 OS, run the installer from an administrator command prompt with the switch `OPT_NO_X86=1`.

1. Restart the system or execute **net stop was /y** followed by **net start w3svc** from a command prompt. Restarting IIS picks up a change to the system PATH, which is an environment variable, made by the installer.

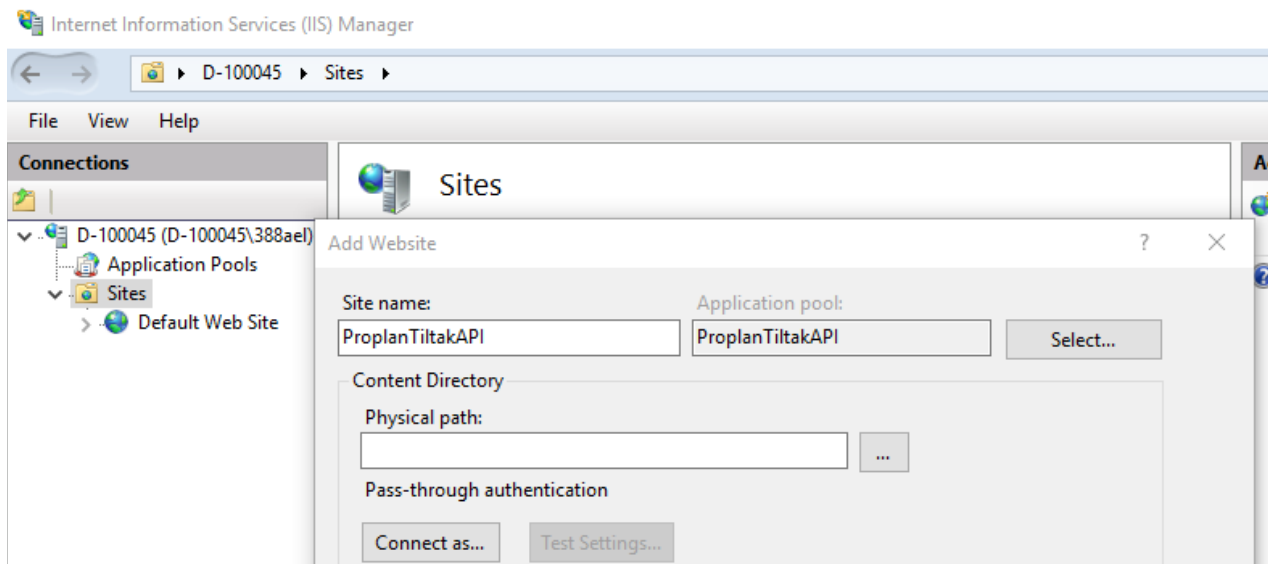
If the Windows Hosting Bundle installer detects that IIS requires a reset in order to complete installation, the installer resets IIS. If the installer triggers an IIS reset, all of the IIS app pools and websites are restarted.

## Innstallere og sette opp APllet i IIS

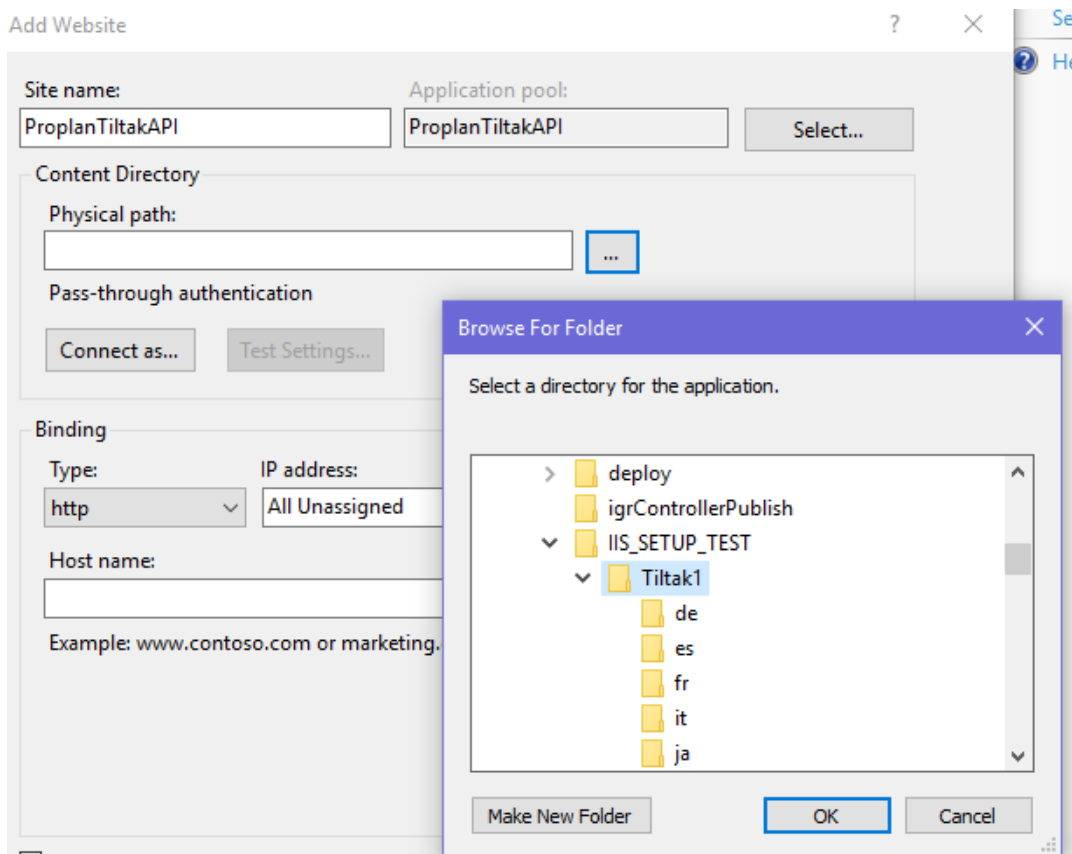
Åpne opp Internet Information Services (IIS) og legg til ny webside.



Gi denne et gjenkjennelig navn (valgfritt). Når du er ferdig med dette vil IIS ha foreslått å opprette en egen «ApplicationPool» for APIet, dersom dere ønsker det kan denne endres – men generelt sett er dette *ikke* å foretrekke.

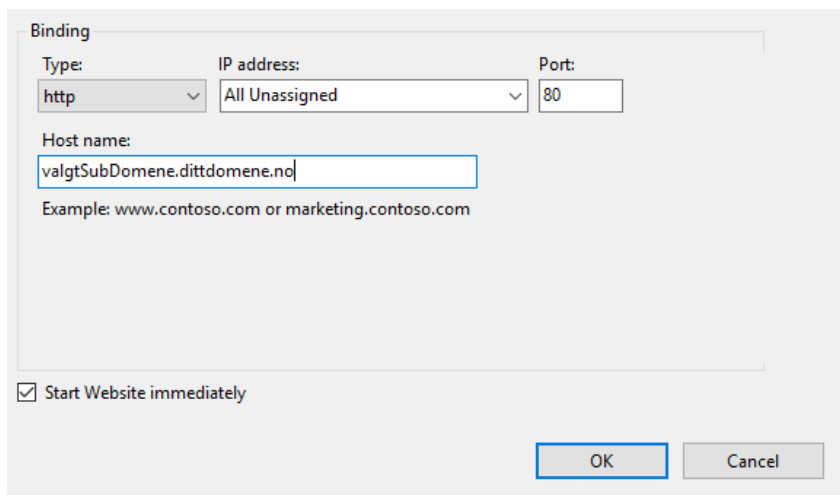


Under «Physical Path» finner du frem til der du la web-filene, og velger denne mappa.

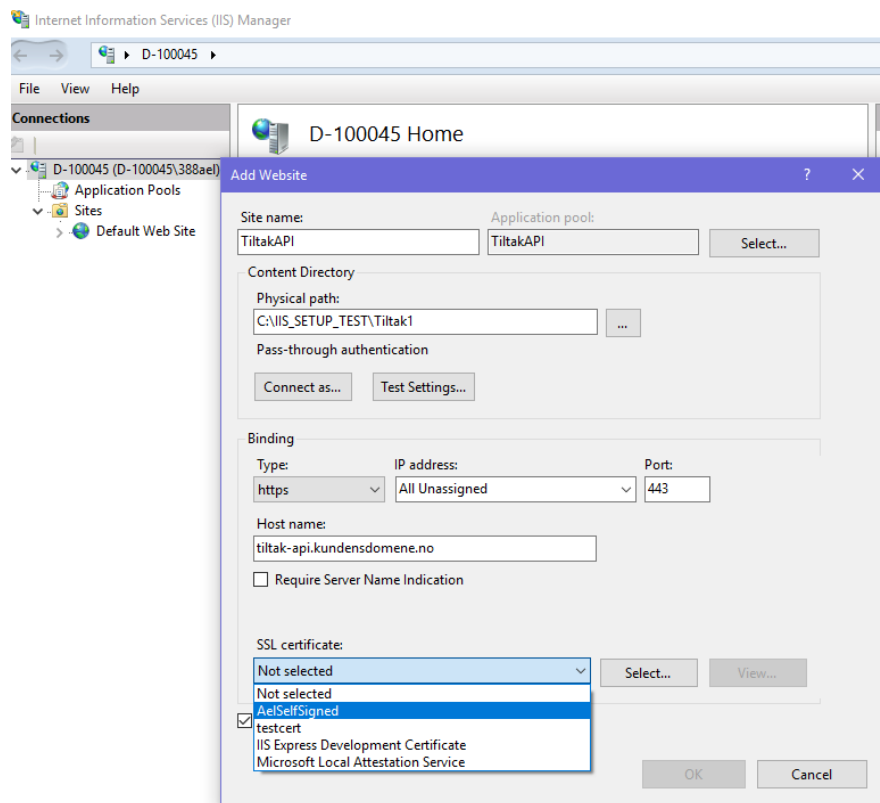


Etter mappen med web-filene for Tiltak er valgt vil du måtte sette opp *bindings* til APIet. Dette er hvilken IP-adresse, domene, protokoll og port APIet skal kjøre på.

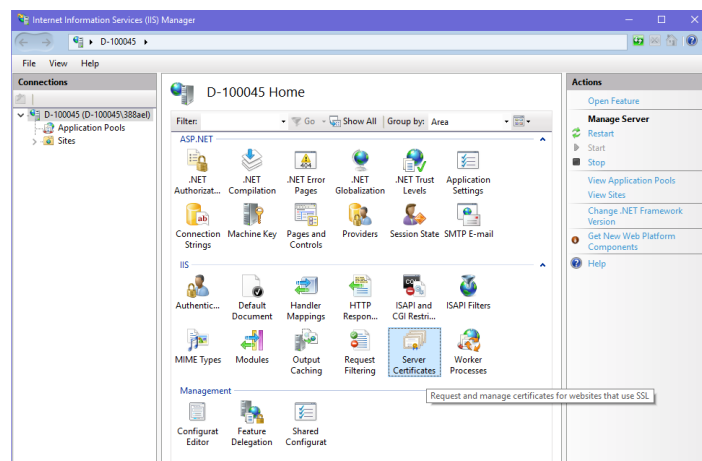
Generelt sett vil vi **ikke** anbefale å kjøre dette på port 80 (HTTP), men dette er opp til kunde/konsulent å bedømme.



For å sette opp Aplet med HTTPS (SSL) må kunden ha et SSL-sertifikat registrert på sitt domene.



Dersom du må legge til sertifikatet på kundens webserver selv, velger du serveren i menyen til venstre i skjermbildet (nivået over «Sites», her: «D-100045») og trykke på «Server Certificates» for så å importere sertifikatet.



OBS! Etter noe trøbbel med SSL-sertifikater som ikke er installert korrekt hos kunde, kan det være lurt å forsikre seg om at dette er satt opp korrekt (dette primært for løsninger som er tilgjengelig utenfor VPN). Dette kan valideres enkelt fra mange gratis verktøy på nettet (søk etter “ssl check”). Eksempelvis dette <https://www.ssllabs.com/ssltest/>

Har kunden ikke et sertifikat kan det skaffes gratis via <https://letsencrypt.org/>

## Verifiser at APllet er kjørende

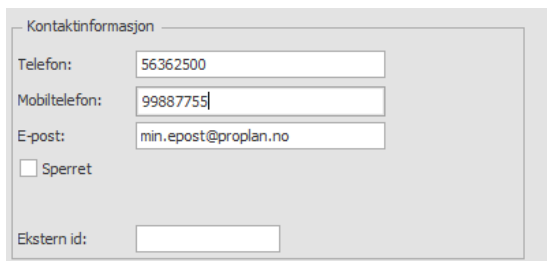
Gå inn på adressa til APllet (for eksempel: «tiltak-api.kundensdomene.no/api/values/test») fra en nettleser. Denne kodesnutten sjekker tilkobling mot systemdatabase og klientdatabase, samt sjekker om lisens har tilgang til mobil – og gir så ut en statusmelding i form av JSON.

```
{ ▼ 5 properties, 110 bytes
  "systemDbTestSuccess": true,
  "clientDbTestSuccess": true,
  "hasMobile": true,
  "dbVersion": "18.3.0.10",
  "messages": []
}
```

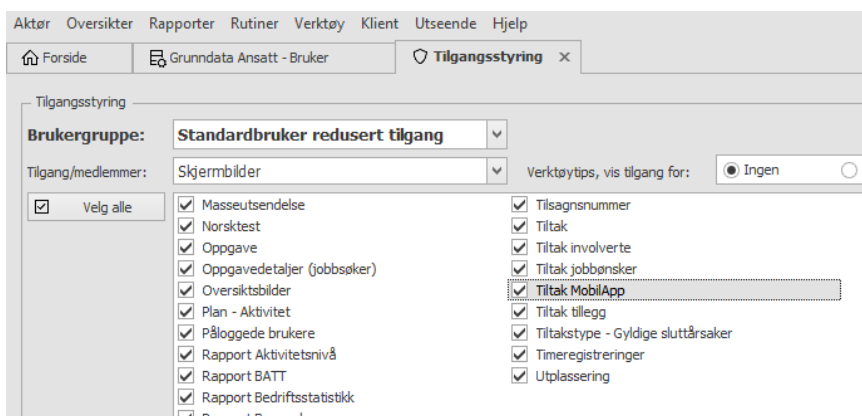


## Oppsett Tiltak

Brukeren som skal benytte seg av app må ha et korrekt registrert mobilnummer på sin tilknyttede "Aktør" av type "Ansatt".



Brukeren må også ha tilgang til skjermbilde "Tiltak MobilApp".

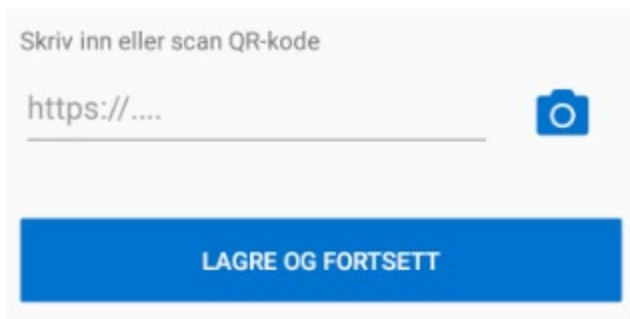


## Oppsett mobilklient

Ved første oppstart av app vil brukeren få spørsmål om å skrive inn URL eller scanne QR-kode.

URL'en brukerne må skrive inn er den samme som ble satt i IIS tidligere, med "/api" på slutten. For eksempel "https://tiltak-api.kundensdomene.no/api".

For å generere en QR-kode finnes det mange [gratisverktøy](#) på nettet.



Etter bruker har scannet/fylt ut API-endepunktets adresse og lagret (denne valideres før brukeren kommer videre) er oppsettet ferdig.