

OS2sync

Installation Guide

Version: 2.0.0
Date: 25.11.2019
Author: BSG

Content

1	Introduction	3
2	Pre-requisites	3
3	Windows Service Installation	3
4	Docker Installation	4
5	Configuration	4
6	Database	6

1 Introduction

OS2sync is distributed both as

- Source code
- Precompiled Windows Service
- Docker image

This document covers the installation of the precompiled windows service, as well as the docker image.

2 Pre-requisites

Before OS2sync can be taken into use, a FOCES certificate must be procured, and all relevant service-agreements must be in place.

A service-agreement is created at KOMBITs administrationsmodul

<https://admin.serviceplatformen.dk>

It is important that the "Organisation V5" service, as well as the coupled "organisation2" service is part of the agreement.

Make sure to say "Ja" to the SeNavn and SeCpr data constraints (if that information is needed). Otherwise "Nej" must be entered, or the agreement will not work.

3 Windows Service Installation

Installing the Windows Service is as simple as running the EXE installer. It will install all the needed files.

The folder contains two relevant configuration files, that must be configured before the service is started.

- appsettings.json
- log.config

The log.config file contains the configuration for logging, and is configured to log to c:\logs\OS2sync, and store up to 10 logfiles of 1 MB each. The file can be edited to change these values, and the default loglevel of INFO can also be changed (e.g. to DEBUG to enable debug logging).

The appsettings.json file contains the full configuration for the application. See the configuration section for details.

Note that the configuration requires the FOCES certificate to be available as a file, and not installed in the Windows Keystore as usual.

Once the configuration files has been updated, the service should be configured to run under a service account that can write to the logfile, and access the FOCES keystore file.

The name of the Service is "OS2sync".

4 Docker Installation

It is assumed that the reader is familiar with Docker, and already have a running Docker environment.

The OS2sync container needs to be configured with access to the FOCES certificate, as well as the set of environment variables mentioned in the Configuration section. A small sample docker-compose.yml file is shown below

```
version: "2.0"
services:
  os2sync:
    image: os2sync/linux:2.0.0rc1
    ports:
      - 5000:80
    environment:
      ClientCertPath: "/cert/keystore.pfx"
      ClientCertPassword: "password"
      EnableScheduler: "false"
      Municipality: "12345678"
      Environment: "TEST"
      LogLevel: "INFO"
    volumes:
      - /path/to/keystore:/cert
```

In the above sample, only the environment values that differ from default is set, and in this case, the Scheduler (which sends data to STS Organisation) is disabled, so the installation can only be used for reading data.

A volume is mounted, which allows configuring OS2sync with a keystore, containing the FOCES certificate that grants access to STS Organisation.

Note that the default port inside the container is port 80 (http), and in the sample it is mapped to port 5000 externally. Change this to fit your own environment.

5 Configuration

The following settings can be added either to the appsettings.json or Docker environment settings. Note that the appsettings.json file does not have any default values, whereas the Docker container comes pre-configured with defaults, which can be overwritten by environment variables.

Key	Docker default	Description
ClientCertPath		The path to the PFX/P12 file containing the FOCES certificate

ClientCertPassword		The password to the above keystore
DisableRevocationCheck	false	<p>This true/false flag is used to control whether OS2sync performs revocation check on certificates.</p> <p>When running in the test-environment it is recommended to set this flag to true, as the service to check revocation against is not easily accessible in the test-environment of DanID/Nets.</p>
LogRequestResponse	false	<p>This true/false flag is used to tell OS2sync if it should log the full request/response payloads when calling STS Organisation. It is recommended to set this value to false in production, as the payloads are quite big.</p>
Municipality		<p>This is the CVR number of the municipality.</p> <p>It must be set to use the build-in GUI in OS2sync but can also be used to set the default CVR, so it is not required as a parameter when using the REST services.</p>
Environment		<p>This flag is used to set which environment of STS Organisation to call against. The allowed values are</p> <ul style="list-style-type: none"> ▪ TEST. This value will configure OS2sync to call against the test environment on the Serviceplatform. ▪ PROD. This value will configure OS2sync to call against the production environment on the Serviceplatform.
EnableScheduler	false	<p>When set to "true", the scheduler is enabled, and any data sent to the queue is synchronized with STS Organisation.</p> <p>This must be enabled to allow updating data in STS Organisation.</p>
DatabaseType	MSSQL	<p>When the scheduler is enabled, this must be set to either</p> <ul style="list-style-type: none"> ▪ MSSQL. To use Microsoft SQL Server as the queue ▪ MYSQL. To use a MySQL (or compatible) database as the queue
DBConnectionString		<p>When the scheduler is enabled, this must be set to the full connection string to connect to the database above.</p>

		<p>Example</p> <pre>server=172.17.0.1; user id=root; password=Test1234; persistsecurityinfo=True; port=3306; database=os2sync</pre>
ApiKey		<p>Can be set to require a password for calling the REST services. If left blank, no password is required.</p>
LogLevel		<p>Only relevant for Docker containers. Must be set to one of the following</p> <ul style="list-style-type: none"> ▪ DEBUG ▪ INFO ▪ WARN ▪ ERROR <p>And is used to configure the logging level. It is recommended to use INFO unless debugging is required.</p> <p>The Windows Service is configured through log.config instead</p>

6 Database

When enabling the scheduler, a database must be made available, with the configured schema pre-created (e.g. "CREATE DATABASE os2sync"). The user configured in the ConnectionString must be allowed to create tables, and the OS2sync software will automatically create the required tables.

Note that the success_ and failure_ tables are never truncated and will keep growing as data is synchronized. It is safe to truncate these tables periodically.