



HR Selfservice

REST API Webservice Interface Description

Version 20251028

Contents

Description	3
Limitations	3
Activation	3
Configuration	3
Certificate generation	3
Step 1. Install OpenSSL library	3
Step 2. Generate certificates	4
Test your credentials	5
Authentication	6
URL	6
Allowed Operations	6
GET /api/absences/GetAbsenceCodes/{language}	6
GET /api/employees/GetEmployees/{language}	6
POST /api/absences/AbsenceReport	7
GET /api/absences/AbsenceReport	8
GET /api/absences/GetRemainingCallsForToday	9

Description

The HR Selfservice Absences Webservice allows you to integrate the data registered in HR Selfservice Absences into your own (planning) application. Therefore, we provide a Web Service API available via the REST protocol secured with OIDC Client Credentials Flow.

Limitations

The HR Selfservice Webservice allows a maximum number of function calls per day.

	Maximum number of requested months per day	Count
Limit	1000	
Example1	Request absences for a company of 100 employees, period=august	100x1=100
Example2	Request absences for 1 employee, period=2019	1x12=12

Activation

Contact support@sdworx.com to have the webservice activated and receive your webservice ClientId. You need to pass your certificate (PEM certificate) to us to arrange access. More information on certificate creation can be found in this document.

Requests for access to the HR Selfservice API are only accepted from a contact person known to SD Worx.

Configuration

Certificate generation

Step 1. Install OpenSSL library

For the certificate generation example presented in this document we are using OpenSSL Library ([GitHub - openssl/openssl: TLS/SSL and crypto library](#)).

All operations below are performed on Windows 11.

Option 1 – Visual installer

Use visual installer available under following URL:
<https://slproweb.com/products/Win32OpenSSL.html>

Then open 'Command Line for OpenSSL'

Option 2 – Command line package manager

Use package manager like Chocolatey.

- Open the Terminal (PowerShell) as Administrator (or user with local administrator privileges) in Windows.
- Run following command to install Chocolatey package manager:

```
Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```
- Use a package manager to install OpenSSL:

```
choco install openssl
```
- Create a new directory:

```
mkdir Certificate
```
- Go to the Certificate folder:

```
cd Certificate
```
- You are now ready to generate certificates.

Step 2. Generate certificates

- Generate CSR and KEY using the following command:

```
openssl req -new -newkey rsa:4096 -nodes -keyout certificate.key -out certificate.csr
```
- Go through the configuration steps as shown in the console.
IMPORTANT! If you configure certificate with a 'challenge password' make sure you store this password for later use in the connection configuration.
- Generate the PEM and self-sign with KEY:

```
openssl x509 -req -sha256 -days 365 -in certificate.csr -signkey certificate.key -out certificate.pem
```
- The PEM file is a text file. It contains a private key section and a certificate section.
- Convert the certificate to the PFX format.

```
openssl pkcs12 -export -out certificate.pfx -inkey certificate.key -in certificate.pem
```

To display the different files in Windows Explorer window just execute following command (dot at the end of the command means 'use current location'):

```
explorer .
```

PFX file is used to 'sign' the connection to the API.

- ```
cat certificate.pem
```

This will show the contents of the certificate file in your terminal window.

- Copy the certificate section and send it with your API access request to support@sdworx.com.

The certificate section is the one between following dividers.

```
-----BEGIN CERTIFICATE-----
[...]
-----END CERTIFICATE-----
```

This is the part of the PEM file that needs to be passed to SD Worx support.  
Only send the certificate to SD Worx.

**Remember that you should not send the private key section from the PEM file or any of the other generated files. These should never be shared with anyone outside your organization.**

- In return you will get a ClientId from SD Worx.  
The ClientId is a long alphanumeric string.

## Test your credentials

Once you receive your ClientId, test your API access with the API test application.

- Download the test application [here](#).
- Unzip the downloaded file.
- Copy the certificate.pfx file you generated to the Certificates folder in the unzipped test application folder.
- Edit the .config file in the root folder of the test application.

```
<appSettings>
```

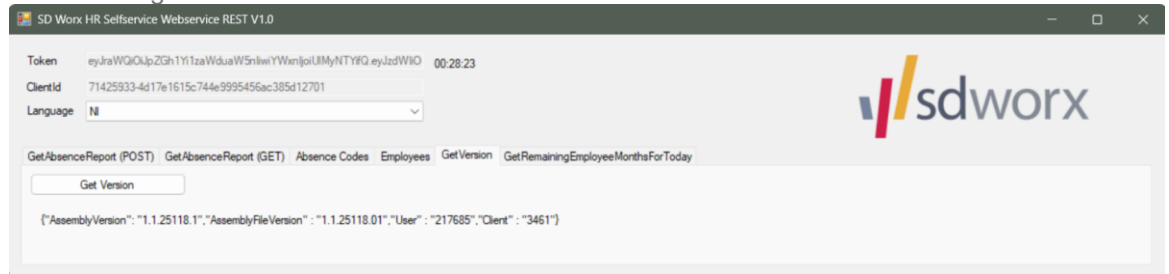
```
<add key="ClientId" value="PASTE YOUR CLIENT ID HERE"
/>
<add key="CertName" value="certificate.pfx" />
<add key="PfxPassword" value="CHALLENGE PASSWORD IF USED
OTHERWISE KEEP EMPTY" />
<add key="oidc:aud" value="hrssapi" />
<add key="oidc:scope" value="hrssapi" />
<add key="oidc:authority"
value="https://auth.sdworx.com/idhub/oidc"/>
<add key="ApiBaseUrl" value="https://api.sdworx.com/api-
hrselfservice"/>
```

```
</appSettings>
```

- Open the test application.
- Open the **Get version** tab.

- Click **Get version**.

You should get a result similar to the one shown below.



## Authentication

The HR Selfservice Absences API is using OIDC/OAuth Client Credentials Flow with a certificate. A Certificate signed OIDC authority request is used to obtain the JWT token which can then be used for further API calls. The token expiration time is 30 minutes. Refreshing of the JWT token has to be addressed in your solution.

## URLs

OIDC Authority URL: [auth.sdworx.com/idhub/oidc](https://auth.sdworx.com/idhub/oidc)

OIDC Authority Configuration: <https://auth.sdworx.com/idhub/oidc/.well-known/openid-configuration>

API Endpoint URL: <https://api.sdworx.com/api-hrselfservice>

## Allowed Operations

### **GET /api/absences/GetAbsenceCodes/{language}**

Required Parameters: Language (int, 1 = NI, 2 = Fr, 3 = En)

Request URL:

<https://api.sdworx.com/api-hrselfservice/api/absences/GetAbsenceCodes/1>

Resultset:

- Id: int
- Label: string
- Code: string

### **GET /api/employees/GetEmployees/{language}**

Required Parameters: Language (int, 1 = NI, 2 = Fr, 3 = En)

Request URL:

`https://api.sdworx.com/api-hrselfservice/ api/absences/ GetEmployees/1`

Resultset:

- `EmployerNumber: string`
- `EmployeeNumber: string`
- `FirstName: string`
- `LastName: string`

## **POST /api/absences/AbsenceReport**

Sample request URL:

`https://api.sdworx.com/api-hrselfserviceapi/absences/AbsenceReport`

Required Parameters: class ReportRequest

```
{
 "Language": 0,
 "FromDate": "string",
 "UntilDate": "string",
 "EmployerNumber": "string",
 "EmployeeNumber": "string",
 "RequestType": 0,
 "RequestOvertime": 0,
 "ChangedSinceDate": "string"
}
```

Properties of ReportRequest:

- `Language: int ( 1 = Nl, 2 = Fr, 3 = En )`
- `FromDate: DateTime`
- `UntilDate: DateTime`
- `EmployerNumber: string ( optional )`
- `EmployeeNumber: string ( optional )`
- `RequestType: Enum.RequestType`
  - `Absences` : returns collective absences and individual leave requests
- `AbsencesAndSuspensions` : returns collective absences, individual leave requests and suspensions ( parental leave, part time absences due to sickness, ... )
  - `FullDaySpec` : returns collective absences, individual leave requests, suspensions ( parental leave, part time absences due to sickness, ... ) and the complete planning for every day
- `RequestOvertime: Enum.RequestOvertime`

- o WithOvertime : includes overtime in the result
  - o WithoutOvertime : excludes overtime in the result
- ChangedSinceDate: Optional parameter for querying data that has been changed since "LastModificationDate"

**Resultset:**

Returns an array of WebserviceResultLine objects:

- EmployerNumber: string
- EmployeeNumber:string
- FirstName: string
- LastName: string
- Date: DateTime
- DayPart: string
- OriginalAmount: decimal
- OriginalMeasureUnit : string
- InterpretedAmount : decimal
- InterpretedMeasureUnit: string
- AbsenceCodeId: string
- AbsenceCodeLabel: string
- Layer: string
- Status: string
- ShiftCode: string
- LastModificationDate: DateTime
- Id: string

### **GET /api/absences/AbsenceReport**

Sample request URL:

<https://api.sdworx.com/api-hrselfservice/api/absences/AbsenceReport?language=1&fromDate=2025-12-01&untilDate=2025-12-03&employerNumber=123&employeeNumber=124&requestType=1&requestOvertime=0&changeSinceDate=2025-12-01>

**Required Parameters:**

- languageId: int ( 1 = Nl, 2 = Fr, 3 = En )
- fromDate\_ddMMyyyy: string
- toDate\_ddMMyyyy: string
- requestType: int ( 0 = Absences, 1 = AbsencesAndSuspensions, 2 = FullDaySpec )
- withOvertime: int
- employerNumber: string
- employeeNumber: string
- ChangedSinceDate: Optional parameter for querying data that has been changed since "LastModificationDate"

**Resultset:**

- EmployerNumber: string
- EmployeeNumber: string
- FirstName: string
- LastName: string
- Date: DateTime
- DayPart: string
- OriginalAmount: decimal
- OriginalMeasureUnit : string
- InterpretedAmount : decimal
- InterpretedMeasureUnit: string
- AbsenceCodeId: string
- AbsenceCodeLabel: string
- Layer: string
- Status: string
- ShiftCode: string
- LastModificationDate: DateTime
- Id: string

**GET /api/absences/GetRemainingCallsForToday**

Parameters : None

Request URL:

[https://api.sdworx.com/api-hrselfservice/  
api/absences/GetRemainingCallsForToday](https://api.sdworx.com/api-hrselfservice/api/absences/GetRemainingCallsForToday)

**Resultset:**

- A string containing the remainder of the allowed function calls per day. See Limitations.