

WSAPI

Web-Service API

System Dokumentation til

Sundhedsvæsenets Kvalitets Instituts webservice 2.0

Udgiver:

Sundhedsvæsenets kvalitetsinstitut
Hedeager 3
8200 Aarhus

www.Sundk.dk

Version 1.2
Versionsdato: 12-09-2025



Indholdsfortegnelse

Indholdsfortegnelse	3
Introduktion til WSAPI	5
Tekniske specifikationer.....	5
Datamodel	5
Til udviklere og dataleverandører	6
Værktøjer	6
Swagger UI	6
Redocs	6
Webservice dokumentationer	6
Support	6
Adgang til test-miljøet.....	6
Adgang til Prod-miljøet.....	7
Verifikationsprocessen	7
Endpoint Opbygning.....	8
JSON-web-token (JWT).....	9
Access-token flow	9
Python kode eksempel.....	11
Groovy kode eksempel	11
Refresh-token flow	12
Indberetning af data	13
Data_Entry_ID (Indberetnings ID)	13
Server svar	13
Status koder	14
REST aktioner	14
Formatering: JSON-format.....	15
Til datamanagere	17
Værktøjer	17
Webservice dokumentationer	17
Indberetningsvariabler	17
System variabler	18

Introduktion til WSAPI

WSAPI anvender en anden teknologi stack og datamodel end "RKKP webservice 1.0", og dermed skal det forventes at der skal laves ændringer i indberetningssnitfladen hos dataleverandører og at der kan være ændringer i dataarbejdet hos datamanagere. Det er en prioritet i udviklingsteamet at der stilles værktøjer og dokumentation til rådighed til alle som arbejder med WSAPI.

WSAPI er udviklet med fokus på, at øge performance, skalering og brugen anvendelighed. Følgende er en række highlights omkring:

- **Versionsstyring:** Flere versioner af database endpoints online på samme tid, for at understøtte rullende opdatering hos dataleverandører.
- **Sikkerhed:** Ekstra sikkerhed med JWT authentication.
- **Nye dokumentationsværktøjer:** [Swagger UI](#) / [Redocs](#) / [Web dokumentation \(ex. DAH\)](#)
- **Nye bruger muligheder:**
 - Delvis opdatering af indberetninger (PATCH method).
 - Brugere kan finde og søge efter egne indberetninger.
 - Brugere kan ved kritiske fejl slette deres egne indberetninger.
 - Brugere kan se alle succesfulde transaktioner i endpointet /Dailytransaction/
 - Brugere kan se alle deres fejl i op til en måned på /ErrorLogs/

Tekniske specifikationer

WSAPI er en REST-ful API, med brugerkontrol gennem JSON-web-tokens (JWT). I WSAPI skal Indberetninger sendes i JSON format, og bliver valideret mod et prædefineret skema. Skema strukturen standartiseres med en overordnet datamodel under navnet "Den Logiske Datamodel".

WSAPI vil over tiden erstatte vores originale webservice-stack, som er baseret på en SOAP-protokol og som arbejder med WSDL/XML.

Datamodel

WSAPIs datamodel er en anvendelsesmodel som tager udgangspunkt i "Den Logiske Datamodel" som er udarbejdet internt i SundK til at kunne rumme sundhedsdata. Som udgangspunkt udvides anvendelsesmodellen løbende som behovet for mere komplekse webservices opstår.

Til udviklere og dataleverandører

Værktøjer

[Swagger UI](#)

Swagger UI er et værktøj som giver overblik over alle webservices på WSAPI, og er et værktøj tiltænkt udviklere som skal arbejde med WSAPI. Det bruges i høj grad internt i SundK til udvikling på WSAPI.

Swagger UI kan vise JSON-eksempler og API kald til indberetninger til specifikke webservices. Det kan bruges til at interagere direkte med WSAPI på udviklingsmiljøet i browseren (kræver brugerlogin).

[Redocs](#)

Redocs er en alternativ til swagger, hvor man går i dybden med bl.a. valideringen af endpoints, og se hvilke krav der er til de individuelle JSON-elementer.

[Webservice dokumentationer](#)

Dette er SundKs eget udviklede dokumentation til de individuelle webservices. Dette værktøj er målrettet til klinikkere, projekt ledere og dem som skal mappe data.

Du kan finde links til dokumentationssiderne inde på databasesiderne under "indberetning". Det anbefales at gå direkte til databaserne fra WSAPI egen underside på sundk.dk

Support

For teknisk support kan SundK kontaktes gennem [denne formular](#).

Adgang til test-miljøet

For at få adgang til test miljøet, kontaktes SundK gennem [denne formular](#) med følgende fil udfyldt og vedhæftet: <https://sundk.dk/media/clcabr1o/adgang-til-wsapi.docx>

Sundk vil derefter oprette jeres bruger og sende jer brugernavn og kodeord.

Test-miljøet er ikke på SDN, og er ikke et sikkert miljø at sende personfølsomme oplysninger til. Der skal derfor benyttes genereret testdata under udvikling. Da CPR-nummer bliver valideret som format kan det anbefales at bruge et officielt test CPR-nummer som f.eks. "2311143995".

Adgang til Prod-miljøet

For at få adgang til produktionsmiljøet påkræves det at der oprettes en service aftale med sundhedsdatanettet. Service aftale nummeret er **#3539 TCS-SDN-Services**

Der er følgende oplysninger for SDN på produktion:

SDN IP: 77.243.49.187

SDN DNS-navn: <https://tcs-sdn-prod1.netic.dsdn.dk/rkkp-wsapi-prod/wsapi-generic-service>

Port/Protokol: 443 / HTTPS

Verifikationsprocessen

For at kunne gå over til produktionmiljøet, skal dataleverandøreren udføre en verifikationsproces. Verifikationsprocessen har til formål at hjælpe dataleverandører med at komme rundt om alle funktionerne i WSAPI, således at de er klar over hvilke muligheder som står til rådighed til dem, samt at systemet anvendes optimalt. Det nuværende verifikationsdokument er vedhæftet e-mailen.

Endpoint Opbygning

Alle endpoints følger navnestandarden ***/DatabaseNavn/DatabaseDomain/DatabaseVersion/***

Desuden har alle endpoints de samme funktioner tilgængelige:

Request Method	Endpoint	Formål
POST	/	Indsend indberetning i JSON format
POST	/Batch/	Indsend et batch a flere indberetninger
GET	/	Få en liste af data_entry_ids i en given periode
GET	/entry_id	Udtræk af en specifikt data_entry_id
PUT	/entry_id	Opdater en indberetning (komplet) i WSAPI ved hjælp af en komplet JSON struktur
PATCH	/entry_id	Opdater en indberetning (delvist) i WSAPI ved hjælp af en delvis JSON struktur
DEL	/entry_id	Slet en indberetning
GET	/SanityCheck/	Få en row count i et specifikt dato interval
GET	/DailyTransactions/	Få en liste over de succesfulde transaktioner en given dag
GET	/Schema/	Få det bagliggende POST validerings skema tilbage
GET	/ErrorLogs/	Få en liste af fejl tilbage

JSON-web-token (JWT)

Systemet bruger JSON-web-tokens til brugerstyring. Der findes 2 forskellige tokens:

- Access-token (gyldig i 3 minutter)
- Refresh-token (gyldig i 30 minutter)

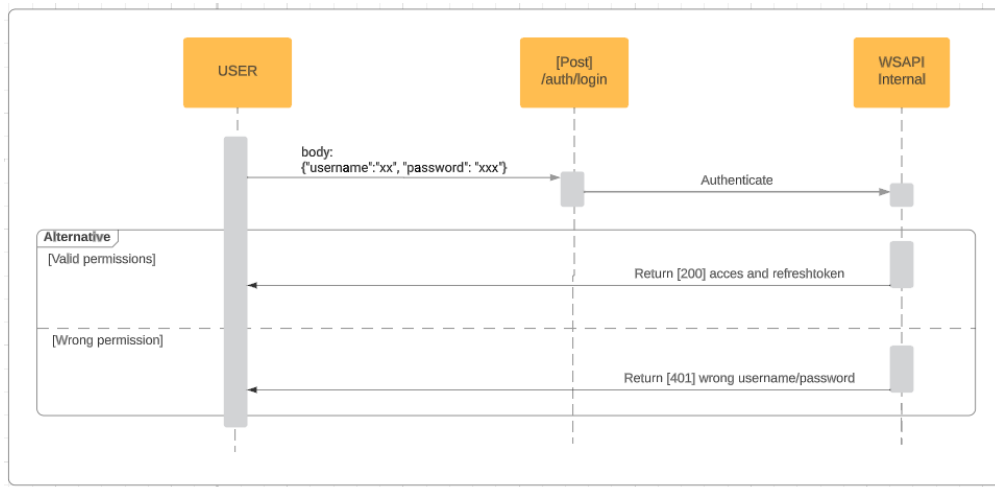
Alle interaktioner med et database endpoint påkræver en gyldig JWT, som sendes med i headeren i http requesten. Det kan se således ud i curl, ved f.eks. afsendelse af en indberetning:

```
curl -X 'POST' \
  'https://rkkp-wsapi.test____' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer Access-token \
  -H 'Content-Type: application/json' \
  -d '{ "Report": ____ }'
```

SundK har valgt at anvende "stateless authentication". Det vil sige at efter omkring 150 sekunder skal der laves et API kald til auth/refresh for at anmode om en ny access- og refresh-token. Ved denne implementering kan systemet permanent lave indberetninger. Vær opmærksom på at refresh-token er valid i 30 minutter.

Access-token flow

Herunder kan flowet for anmodning om en access-token ses.



For at anmode om en access-token skal systemet sende en POST-request til WSAPI's auth endpoint: `/auth/login`. I requesten tilføjes en JSON-struktur med email, og password. Hvis disse er korrekte vil WSAPI returnere en access- og refresh-token. De efterfølgende kodeeksempler er skrevet i Python og Groovy, og bør give en god idé om hvordan det kan se ud at anmode om en access-token:

Python kode eksempel

```
import json, requests

url = https://_____/auth/login

json_data = {"username": "xxx@example.com", "password": "xxx"}
server = requests.post(url, headers = { "Content-Type": "application/json"},
                        json=json_data)

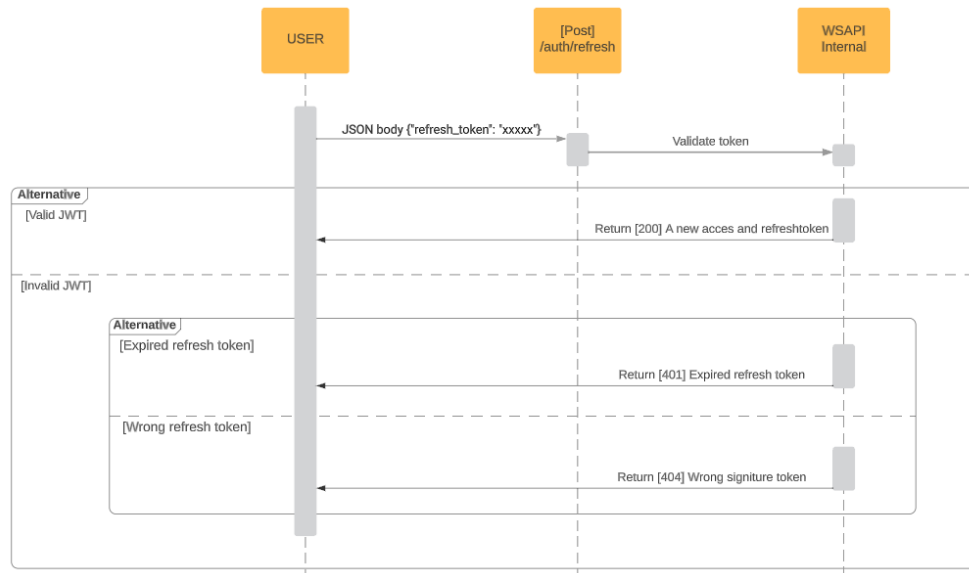
output = json.loads(server.text)
access_token = output["data"]["access_token"]
refresh_token = output["data"]["refresh_token"]
```

Groovy kode eksempel

```
RestBuilder restBuilder = new RestBuilder(connectTimeout: 2000, readTimeout: 2000)
restBuilder.restTemplate.setMessageConverters([new
StringHttpMessageConverter(Charset.defaultCharset.forName("UTF-8"))])
RestResponse response = restBuilder.post(https://_____/auth/login) {
    contentType("application/json")
    accept("application/json")
    json "{\"username\": \"userAccount\", \"password\": \"Password\"}"
}
```

Refresh-token flow

Herunder kan flowet for at anmode om en ny access- og refresh-token ved hjælp af en aktiv refresh-token.



For at anmode om nye tokens skal systemet sende en POST-request til WSAPI's auth endpoint: /auth/refresh. I requesten tilføjes en JSON-struktur med refresh_token. Hvis denne stadig er aktiv vil WSAPI returnere en access- og refresh-token.

Indberetning af data

Som beskrevet i JWT sektionen, skal der vedhæftes en JWT access-token i alle interaktioner med database endpoints.

Et kode eksempel på hvordan man kan interagere med endpoint for DAH databasen med en POST-request i Python:

```
endpoint = https://_____/DAH/FoersteKontakt/1.0.0/

server = requests.post(endpoint, headers = { "Authorization": "bearer ACCESS-
TOKEN", json=YOUR_DATA})

output = json.loads(server.text)
```

Data_Entry_ID (Indberetnings ID)

I det nye system har hver indberetning et "data_entry_id". Det er et auto genereret id til identifikation af indberetninger. Med denne ID kan i selv opdatere, slette eller delvis opdatere jeres eksisterende indberetninger. Serveren vil returnere dette "data_entry_id" ved POST kald.

I kan til enhver tid følge med i alle af jeres transaktioner ved brug af endpointen /logs/. Der kan i søge frem på transaktioner på specifikke endpoints, på specifikke statuskoder, på request metoder, datoer etc.

Server svar

Alle svar er blevet standardiseret, og vil altid komme i det samme format:

```
"timestamp": "2023-01-01 12:00:00.000000",
"status_code": xxx,
"errors ": {"error_1": "___"}
"data": {"Køn": "Kvinde"}
```

Status koder

Serveren kan returnere en flere statuskoder, de mest hyppige er:

200	Succesfuldt kald, og alt er gennemført som forventet.
401	JWT er udløbet
403	Adgang nægtet
404	Ressourcen findes ikke
406/409	Validerings fejl i data definitionen

REST aktioner

Vi understøtter alle standard operationer [POST, PUT, PATCH, DELETE, GET]

POST	Indsende selve indberetningen
GET	Serveren returnerer den anmodede indberetning
PUT	Opdatere all information i en indberetning. Alt data skal sendes én gang til.
PATCH	Del opdateringer på en eksisterende indberetning
DELETE	Slette en eksisterende indberetning. Vigtigt: dette skal kun foretages i meget særlige situationer hvor der tales om alvorlig fejl i forhold til patient oplysninger som CPR. Sletning kan kun udføres af de brugere/organisationer, som oprindeligt har indleveret indberetningen.

Formatering: JSON-format

JSON er case-sensitiv, derfor har vi valgt at alle (enums / udfald), bliver skrevet med småt. For nogle systemer kan det desuden være relevant at sætte encoding til **utf-8**.

Til datamanagere

Værktøjer

[Webservice dokumentationer](#)

Dette er SundKs egen udviklede dokumentation til de individuelle webservices. Formålet med værktøjet er at give en oversigt over datamodelleringen på en webservice, samt valideringen på indberettet data.

Du kan finde links til dokumentationssiderne inde på databasesiderne under "indberetning". Det anbefales at gå direkte til databaserne fra WSAPI egen underside på sundk.dk

Indberetningsvariabler

Indberetningsvariabler er standardiserede variabler som forekommer ens på alle webservices.

CivilRegistrationNumber_Patient	CPR nummer på patienten som indberetningen omhandler.
OrganizationType_Organization	Typen af "Identifier_Organization". Vil oftest være "SOR".
Identifier_Organization	En identificerende kode på indberetningsorganisationen. Typen af kode er i OrganizationType_Organization.

System variabler

WSAPI har nogle interne system variabler som eksterne leverandør ikke skal tage højde for. Disse bliver automatisk sat af systemet

entry_id	WSAPI interne ID for indberetningen. Hvis du har brug for at spore data hele vejen til kilden (WSAPI), eller har brug for at kontakte udviklingsteamet om en bestemt indberetning fra WSAPI.
_report_version	Versionen for webservicen. Hvis du i fremtiden har brug for at vide hvilken version af webservicen (og derved dataspecifikation) data originalt kommer fra.
_provider_id	WSAPI brugernavnet på den indberettende dataleverandør. Hvis du har brug for at vide hvilken dataleverandør, der har indberettet data. Kontakt udviklingsteamet hvis du skal have _provider_id oversat.
_report_name	Databasens forkortelse.
_report_domain	Domænet er WSAPI's mulighed for at have mere end én webservice tilkoblet en database, som f.eks. ved Basis/Trombolyse på DanStroke. Kan ignoreres når kun der er ét domæne/skema, og der ikke forventes at skulle udvide med flere domæner/skemaer senere.
_request_type	API variabel fra WSAPI. Begrænset antal muligheder, men kan bruges til at se om en indberetning er uændret ("POST"), eller opdateret ("PUT/UPDATE"). Ved mest dataarbejde kan den nok ignoreres.
_timestamp	Hvornår indberetningen er sendt til WSAPI. Oftest ingen relevans ift. det udførte kliniske arbejde.