UniLock
System 10

Manual
UniLock WebAPI

Projekt	PCS125-20
Version	1.0
Revision	240701

WebAPI giver en tovejs sikret webforbindelse for eksterne systemer til UniLock adgangskontrol.

Eksterne systemer, som SCADA, PSIM, overvågningssystemer, cloudsystemer, smartphone Apps mv., kan udføre driftsovervågning, kontrol og administration af UniLock adgangskontrol.

WebAPI anvender SignalR til kommunikation, som giver real-time tovejs funktionalitet via en Web forbindelse (WebSocket) mellem klienter og server. Udover at udveksle data giver dette også mulighed for at overvåge selve forbindelsen.

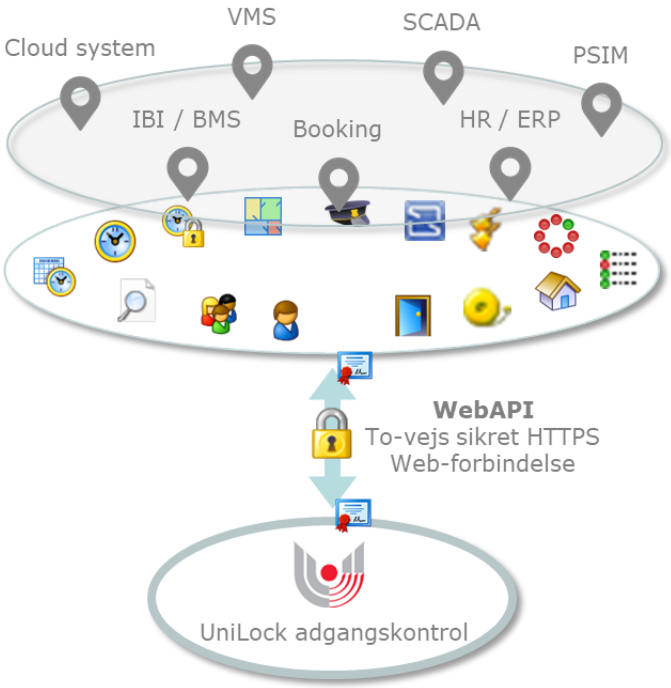
Der anvendes standard dataformatet JSON.

For at hjælpe udviklere godt i gang med at kommunikere med UniLock WebAPI medfølger SDK med fuldt funktionsdygtigt klientprogram inklusiv .NET kildekode og Visual Studio projekt. Derudover kan udviklingsværktøjet Swagger aktiveres i udviklingsfasen.

Indholdsfortegnelse

1. Beskrivelse

1.1 Generel beskrivelse



Anvendelse

WebAPI giver en tovejs sikret webforbindelse for eksterne systemer til UniLock adgangskontrol.

Eksterne systemer, som SCADA, PSIM, overvågningssystemer, cloudsystemer, smartphone Apps mv., kan udføre driftsovervågning, kontrol og administration af UniLock adgangskontrol.

Driftsovervågning udføres ved abonnere på aktuel status for relevante døre, DAS, tyverialarmer, indgange og udgange, logninger, områder, alarmer mv.

Kontrol giver mulighed for at fjernkontrollere døre og DAS (tyverialarmer).

Administration giver mulighed for at oprette/hente/opdatere/slette personer, persongrupper, k-punkter, DAS, tidsstyringer mv.

Anvend dit eget system ovenpå UniLock adgangskontrol.

Beskrivelse

WebAPI anvender SignalR til kommunikation, som giver real-time tovejs funktionalitet via en Web forbindelse (WebSocket) mellem klienter og server. Udover at udveksle data giver dette også mulighed for at overvåge selve forbindelsen.

Der anvendes standard dataformatet JSON.

For at hjælpe udviklere godt i gang med at kommunikere med UniLock WebAPI medfølger SDK med fuldt funktionsdygtigt klientprogram inklusiv .NET kildekode og Visual Studio projekt. Derudover kan udviklingsværktøjet Swagger aktiveres i udviklingsfasen.

Beskrivelse af UniLock adgangskontrol funktionalitet i pc-programmet findes i manual for pc-programmet.

1.2 Begrænsninger

WebAPI kan anvendes fuldt ud når der er erhvervet rettighed til WebAPI-modulet, og i demo-mode kan WebAPI anvendes tidsbegrænset i forhold til antal gange det aktiveres.

En række objekttyper kan allerede styres gennem WebAPI, mens flere objekttyper løbende tilføjes.

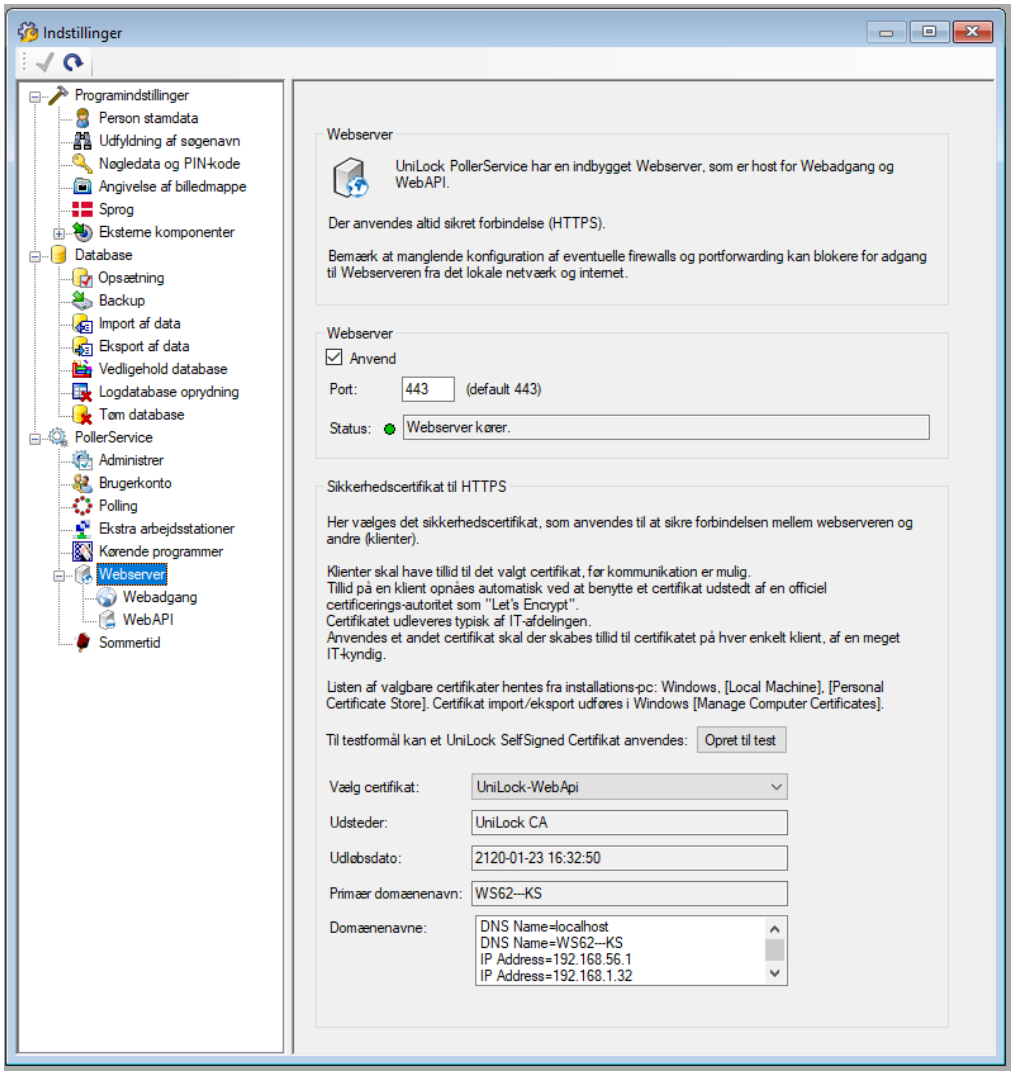
WebAPI returnerer information om manglende rettigheder når rettighedsbegrænsninger opleves.

2. Opsætning

WebAPI er hosted i den indbyggede Webserver i UniLock PollerService. Webserveren skal således aktiveres før WebAPI kan aktiveres.

2.1 Webserver

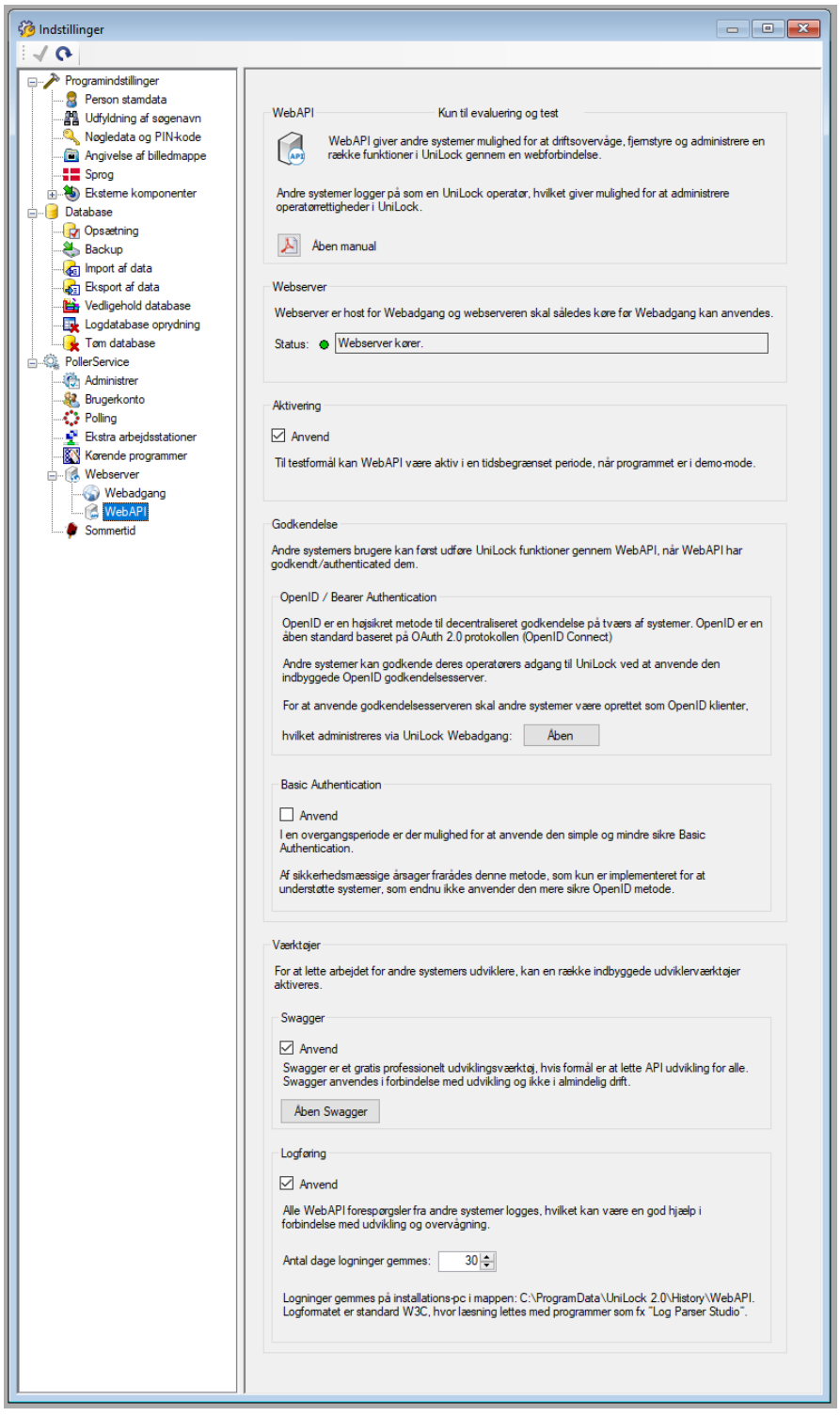
Opsætning af Webserver foretages I UniLock Adgangseditor: [Indstillinger], [PollerService], [Webserver].



UniLock Webserver anvender krypteret forbindelse (HTTPS) til kommunikationsforbindelsen. Til kryptering bør operatøren vælge sit eget certifikat eller alternativt generere et self-signed UniLock certifikat til evaluering og test. Certifikat til kryptering kan vælges ud fra de certifikater som har privat nøglen tilgængelig og placeret på installations-pc i [Windows Certificate Store], [Local Machine], [Personal]. UniLock genererer automatisk alarmer ved certifikatfejl og snarlig udløb af certifikat.

2.2 WebAPI

Opsætning af WebAPI foretages I UniLock Adgangseditor: [Indstillinger], [PollerService], [Webserver], [WebAPI].



Som udgangspunkt er logføring af forespørgsler til WebAPI slået til med 30 dages varighed. Logfilerne er formateret i henhold til standard W3C log, hvilket gør det muligt at læse og analysere logfilerne med gratisprogrammer som: "*Log Parser Studio*".

Logfiler placeres på installations-pc: "C:\ProgramData\UniLock 2.0\History\WebAPI\".

3. Kom godt i gang

For at hjælpe udviklere af klienter til UniLock WebAPI godt i gang medfølger følgende hjælpeværktøjer:

- Klientprogram for demonstration af funktionalitet (WebAPITestClient.exe).
- Projekt-filer og kildekode til klientprogram skrevet i .NET i Visual Studio.
- Indbygget Swagger support, som kan aktiveres/deaktiveres.
- Skemaer til JSON for autogenerering af .NET klasser, TypeScript mv. Se fx https://app.quicktype.io/#l=schema og https://www.jsonschemavalidator.net/.

Hjælpeværktøjerne findes på installations-pc i installationsmappen, hvor standardplacering er: ”C:\Program Files (x86)\UniLock 2.0\Documentation\WebAPI\ClientSourceCode.zip”

4. API-Datakommunikation

I dette afsnit er en overordnet beskrivelse af datakommunikation med UniLock WebAPI.

4.1 Dataforespørgsel

I dataforespørgsler kan klienter angive hvilken kategori, hvilken objekttype og eventuelt specifikt objekt der ønskes data for. Adressestien har denne generelle opbygning: ”https://[IP-adresse]:[Port]/api/[version]/[category]/[objecttype]/”

Muligheder for [category]:

- setup: Indstillinger for objektet.
- operation: Seneste status for objektet, som fx om døren er åben.
- logging: logninger og alarmer.

Muligheder for [objecttype]:

- ControlPoint
- DASGroup
- Gateway
- GatewayConnector
- TimeTable
- SecurityLevelTimeTable
- SharedException
- Person
- PersonGroup
- Operator
- OperatorGroup
- Log
- Area

Fx kan klienter sende en forespørgsel om driftsstatus for k-punktet med ID=752e9150-b9ac-4801-b197-59a35207e552 ved at anvende: ”https://[IP-adresse]:[Port]/api/v1/operation/controlpoint/752e9150-b9ac-4801-b197-59a35207e552”

4.2 Dataformater

UniLock WebAPI kan returnere data i formatet JSON.

4.3 Dato og tid

Som udgangspunkt anvendes ISO 8601 til formatering af dato og tid. Dette giver mulighed for at angive tid som UTC, UTC med et offset eller lokal tid. Lokal tid er Windows tid på installations-pc:

- YYYY-MM-DDTHH:mm:ss (lokal tid)
- YYYY-MM-DDTHH:mm:ssZ (Z = UTC)
- YYYY-MM-DDTHH:mm:ss+HH:mm (UTC+HH:mm)
- YYYY-MM-DDTHH:mm:ss-HH:mm (UTC-HH:mm)

De steder hvor tid angives uden dato, anvendes tidspunktet som udgangspunkt som lokal tid.

4.3.1 Særligt om gyldighedsperioder

Siden Build 186 er ValidityPeriodsDto ”Start” og ”End” en nullable DateTime. Start=null repræsenterer start tidspunkt uendeligt tilbage i tid, mens End=null repræsenterer slut tidspunkt uendeligt frem i tid.

4.4 Paging

Klienter kan forespørge på et udvalg af alle objekter, da WebAPI understøtter paging af objektsamlinger. Dette er implementeret som valgbare Query parametrene [skip] og [take]: ”https://[IP-adresse]:[Port]/api/v1/setup/controlpoint?skip=1&take=4”.

[skip] angiver hvor mange af de første objekter i listen, der skal springes over (default = 0).

[take] angiver hvor mange objekter, der ønskes retur (default = 100).

Eksempler:

Objekter	=	[1,2,3,4,5,6,7,8,9]
Objekter: skip (2)	→	[3,4,5,6,7,8,9]
Objekter: take (2)	→	[1,2]
Objekter: skip (3), take (2)	→	[4,5]

4.5 Datamængde

For at minimere mængden af data, vil standardsvaret til en klient kun indeholde et overblik (summary) over objekter i systemet. Fuld datarepræsentation for objekter kan klienten modtage ved at tilføje Query parameteren ”includeFullObject=true” eller ved at spørge på et specifikt objekt.

Når fuld datarepræsentation anvendes vil hver objekttype have et maksimalt antal objekter per forespørgsel, og paging anvendes hvis klienten skal bruge flere objekter. Header i svar på forespørgsler vil indeholde ”TotalRecordCount”, som angiver det totale antal objekter i systemet.

JSON summary objekt:

```
{
    "Guid": "aac30dae-deaf-4df9-b840-97682dde6efc",
    "Name": "Kontor"
}
```

4.6 Fejlbeskeder

Hvis der opstår fejl i serveren eller forespørgslen til serveren er ugyldig, vil svaret indeholde en beskrivende fejlbesked.

Et eksempel på en fejlbesked er JSON error summary for oprettelse af object, hvis søgenavn allerede er i brug:

```
{
    "ErrorType": "ValidationError",
    "ErrorCode": 2,
    "Message": "Validation failed",
    "Details": [
```

```
        "ErrorType": "NameAlreadyInUse",
        "ErrorCode": 12,
        "ErrorMessages": [
            "[Name] has to be unique"
        ]
    ]
}
```

4.7 Sprog

API'et understøtter sprogvarianter for fejlbeskeder, logninger osv.

For fejlbeskeder er [Details.ErrorMessage] oversat. For logninger er det alle felterne under [Description] der er oversat.

Standardsproget er Engelsk, men kan ændres ved at tilføje en [Language] header til API kaldet. Samme header anvendes når der abonneres på beskeder fra SignalR Hubs.

Der understøttes IEFT language tags for Dansk (da-DK) og Engelsk (en-GB).

5. Tovejs kommunikation via Web

I dette afsnit er en overordnet beskrivelse af hvordan klienter kan oprette tovejs kommunikationsforbindelse med UniLock via Web, som kan overvåges.

Tovejs kommunikationsforbindelse gør bl.a. at klienten kan overvåge om forbindelsen til UniLock er intakt og ikke mistet. Mistes forbindelsen kan klienten således reagere på dette ved at fx advisere sine egne operatører om dette.

Til tovejs kommunikationsforbindelse anvender UniLock SignalR, som er et Microsoft ASP.NET softwarebibliotek som kan bruges til at lave real-time funktionalitet via en Web forbindelse (WebSocket) mellem klienter og server. SignalR giver UniLock mulighed for at sende data til klienter, uden at klienterne hele tiden skal forespørge på data.

Ved forespørgsler til UniLock WebAPI anvendes stien: ”https://[IP-adresse]:[Port]/”.

5.1 SignalR Hubs

SignalR Hubs er forbindelsespunkter som gør det muligt for klienten og serveren at kalde metoder hos hinanden. Hubs gør det endvidere også muligt at kalde metoder med stærkt type parametre og muliggør modelbinding.

Herunder står informationerne omkring de forskellige hubs og hvad man kan med dem.

5.1.1 SetupHub

Med SetupHub kan klienter ved hjælp af SignalR få forbindelse til UniLock og derigennem få besked hver gang objekters indstillinger bliver ændret (notifikationer). En notifikation er et objekt med ID, objekttype og eventtype.

Eventtypen kan være:

1. Created
2. Changed
3. Deleted

Forbindelsesinformationer:

- Navn: SetupHub
- Krævet operatørrettighed i UniLock: [WebAPI]
- Event: ObjectUpdated, ObjectDepartmentSharingRemoved

Subscribing:

Klienter abonnerer ved hjælp af sin *SignalR.HubProxy* hvor man kalder *invoke*, med et af de nedenstående metodenavne som argument.

Fx: *HubProxy.invoke(”SubscribeChangesControlPoints”).*

Objekttype:	**Tilføje abonnement**
K-Punkt	”SubscribeChangesControlPoints”
DAS grupper	”SubscribeChangesDASGroups”

Lokalitet	”SubscribeChangesGateways”
Tidstabel	”SubscribeChangesTimeTables”
Sikkerhedsniveautidstabeller	”SubscribeChangesSecurityLevelTimeTables”
Delte undtagelser	”SubscribeChangesSharedExceptions”
Personer	”SubscribeChangesPersons”
Persongrupper	”SubscribeChangesPersonGroups”
Operator	”SubscribeChangesOperators”
Områder	”SubscribeChangesAreas”
Operatør grupper	”SubscribeChangesOperatorGroups”
Specialdagskalender	”SubscribeChangesSpecialDayCalenders”
Afdelinger	”SubscribeChangesDepartments”

Objekttype	**Opsige abonnement**
K-Punkt	”UnsubscribeChangesControlPoints”
DAS grupper	”UnsubscribeChangesDASGroups”
Lokalitet	”UnsubscribeChangesGateways”
Tidstabel	”UnsubscribeChangesTimeTables”
Sikkerhedsniveautidstabeller	”UnsubscribeChangesSecurityLevelTimeTables”
Delte undtagelser	”UnsubscribeChangesSharedExceptions”
Personer	”UnsubscribeChangesPersons”
Persongrupper	”UnsubscribeChangesPersonGroups”
Operator	”UnsubscribeChangesOperators”
Områder	”UnsubscribeChangesAreas”
Operatør grupper	”UnsubscribeChangesOperatorGroups”
Specialdagskalender	”UnsubscribeChangesSpecialDayCalenders”
Afdelinger	”UnsubscribeChangesDepartments”

Register events:

UniLock kan sende notifikationer til klienten, når klient har konfigureret sin eventhandler:

Fx:

Dynamic typed result:

HubProxy.On(”ObjectUpdated”, (data) => {});

Strongly typed result:

HubProxy.On<SetupEvent> (”ObjectUpdated”, (SetupEvent data) => {});

5.1.2 OperationHub

Med OperationHub kan klienter ved hjælp af en SignalR forbindelse få forbindelse til UniLock og derigennem abonnere på aktuel status af objekttyperne k-punkter og lokaliteter. Hver gang UniLock har hentet ny information fra hardware (k-punkter og lokaliteter), vil OperationHub sende notifikationer (statusobjektet) til alle klienter som abonnerer på det pågældende ID.

Forbindelses informationer:

- Navn: OperationHub
- Krævet operatørrettighed i UniLock: [WebAPI]
- Event: ObjectStatusUpdated, ControlPointStatusUpdated, GatewayStatusUpdated, GatewayConnectorStatusUpdated

Subscribing:

Klienter abonnerer ved hjælp af sin *SignalR.HubProxy* hvor man kalder *invoke*, med et af de nedenstående metodenavne som argument, og objektet's [Guid] som parameter.

Fx:

string id = "05c11e2e-94a6-43b0-b8b0-e312de7402b9";

HubProxy.invoke("SubscribeChangesControlPoint", id);

Objekttype:	**Antal**	**Tilføje abonnement**
K-Punkt	Enkelt	"SubscribeChangesControlPoint" Parameter: String | GUID
	Flere	"SubscribeChangesControlPoints" Parameter: String [] | GUID []
Lokalitet	Enkelt	"SubscribeChangesGateway" Parameter: String | GUID
	Flere	"SubscribeChangesGateways" Parameter: String [] | GUID []
Lokalitetsforbindelse	Enkelt	"SubscribeChangesGatewayConnector" Parameter: String | GUID
	Flere	"SubscribeChangesGatewayConnectors" Parameter: String [] | GUID []

Objekttype:	**Antal**	**Opsige abonnement**
Alle	Enkelt	"UnsubscribeChange" Parameter: String | GUID
	Flere	"UnsubscribeChanges" Parameter: String [] | GUID []
	Alle	"Unsubscribe"

Register events:

GUID sendes som String eller som System.GUID datatype.

UniLock kan sende objekt status til klienten, når klient har konfigureret sin eventhandler:

Fx:

Dynamic typed result:

HubProxy.On(”ObjectStatusUpdated”, (data) => {});

Strongly typed result:

HubProxy.On<OperationControlPoint> (”ControlPointStatusUpdated”, (OperationControlPoint data) => {});

5.1.3 AreaOperationHub

Med AreaOperationHub kan klienter ved hjælp af en SignalR forbindelse få forbindelse til UniLock og derigennem abonnere på områder.
Hver gang der sker en ændring i et område, kan der blive sendt 2 typer af events:

OperationAreaChanged:
Hvis et område er blevet oprettet, opdateret eller slettet. Så sender AreaOperationHub et event til klienter som abonnerer på det pågældende id, eller som abonnerer på alle ændringer for områder. Eventet indeholder informationer om typen af ændring, hvilket område det vedrører og hvad områdets aktuelle værdier som nuværende antal personer i området og det sidste event der er sket i området.

OperationAreaPersonChanged:
Hvis der er sket en ændring på en person, eller personens relation til et område. Så sender AreaOperationHub et event til alle klienter som er forbundet.
Eventet indeholder informationer om typen af ændring, hvilken person det omhandler og personens status i alle de områder som personen er til stede i.

Forbindelses informationer:

- Navn: AreaOperationHub
- Krævet operatørrettighed i UniLock: [WebAPI] og [Områder]
- Event: AreaChanged, AreaPersonChanged

Subscribing:

Klienter abonnerer ved hjælp af sin *SignalR.HubProxy* hvor man kalder *invoke*, med et af de nedenstående metodenavne som argument og objektet’s [Guid] som parameter.

Fx:

string id = “05c11e2e-94a6-43b0-b8b0-e312de7402b9”;

HubProxy.invoke(“SubscribeArea”, id);

Type:	**Antal**	**Tilføje abonnement**
Område	Enkelt	”SubscribeArea”

		Parameter: String | GUID
	Flere	”SubscribeAreas” Parameter: String [] | GUID []
	Alle	”SubscribeAllAreas”

Type:	**Antal**	**Opsige abonnement**
Område	Enkelt	”UnsubscribeArea” Parameter: String | GUID
	Flere	”UnsubscribeAreas” Parameter: String [] | GUID []
	Alle	”Unsubscribe”

Register events:

UniLock kan sende logninger til klienten, når klient har konfigureret sin eventhandler:

Fx:

Dynamic typed result:

HubProxy.On(”AreaChanged”, (area) => {});

Strongly typed result:

HubProxy.On<OperationAreaChanged> (”AreaChanged”, (OperationAreaChanged area) => {});

5.1.4 LogHub

Med LogHub kan klienter ved hjælp af SignalR få forbindelse til UniLock og derigennem få alle nye logninger der bliver lavet i systemet, og få af vide når alarmer bliver afstillet.

LogReceived:
Hver gang der bliver oprettet en logning som bliver sendt ud til adgangseditoren, bliver der sendt et ”LogReceived” event via LogHub til de klienter som abonnere på logningen, enten direkte på log typen, eller på logningens kategori.
Eventet indeholder informationer omkring logningen, og benytter datatypen ”Log”.

AlarmsCleared:
Når der bliver afstillet en eller flere alarm logninger i UniLock, så bliver der sent et event ud som indeholder en liste af de alarm logningers Id, som er blevet afstillet.

Forbindelses informationer:

- Navn: LogHub
- Krævet operatørrettighed i UniLock: [WebAPI] og [Poller]
- Event: LogReceived, AlarmsCleared

Subscribing:

Når man abonnerer på logninger, så de strenge som man kan sende med som parametre, er det som svarer til logningens [Type].

Type:	**Antal**	**Tilføje abonnement**
Log	Enkelt	”SubscribeLog” Parametre: String
	Flere	”SubscribeLogs” Parameter: String []
	Alle	“SubscribeLogs”
Alarm	Alle	“SubscribeAlarms”
Afstillet alarm	Alle	“SubscribeAlarmsCleared”
Info	Alle	“SubscribeInfoLogs”
Operatør	Alle	”SubscribeOperatorLogs”
System	Alle	”SubscribeSystemLogs”
K-Punkt	Alle	”SubscribeControlPointLogs”

Type:	**Antal**	**Opsige abonnement**
Log	Enkelt	”UnsubscribeLog” Parameter: String
	Flere	”UnsubscribeLogs” Parameter: String []
	Alle	“Unsubscribe”
Alarm	Alle	“UnsubscribeAlarms”
Afstillet alarm	Alle	“UnsubscribeAlarmsCleared”
Info	Alle	“UnsubscribeInfoLogs”
Operatør	Alle	”UnsubscribeOperatorLogs”
System	Alle	”UnsubscribeSystemLogs”
K-Punkt	Alle	”UnsubscribeControlPointLogs”

Register events:

UniLock kan sende logninger til klienten, når klient har konfigureret sin eventhandler:

Fx:

Dynamic typed result:

HubProxy.On(”LogReceived”, (data) => {});

Strongly typed result:

HubProxy.On<Log> (”LogReceived”, (Log data) => {});

5.2 Authentication

Klienter skal logge ind for at kunne hente data fra UniLock WebAPI. Der er tre måder at logge ind i WebAPI:

1. Authentication header:

   Den første mulighed er at tilføje en authentication header til sin SignalR forbindelse med samme fremgangsmåde som beskrevet i afsnit 6 Authentication.

2. Cookie:
   Den anden mulighed er at tilføje en cookie til sin SignalR forbindelse, hvor cookiens navn: ”Authorization”, og value skal være med samme fremgangsmåde som beskrevet i afsnit 6 Authentication.
3. Query:
   Den tredje mulighed er at placere sit login i querystring, med key: ”Authorization”, hvor value skal være med samme fremgangsmåde som beskrevet i afsnit 6 Authentication.

Klientens login skal være oprettet som operatør i UniLock og som minimum have operatørrettigheder til at lave API kald [WebAPI]. Den anvendte operatør skal ligeledes have operatørrettigheder til de objekttyper, som skal kunne anvendes via WebAPI.

5.3 Authorization

Klienter kan modtage notifikationer når:

1. Klienten er authenticated, som beskrevet i afsnit 5.2 Authentication.
2. Det anvendte UniLock operatør-login har rettighedsniveau på ”begrænset rettighed” eller højere for de ønskede objekter.

Hvis klienten fx vil:

- Lytte på notifikationer vedrørende k-punkter, så kræves operatørrettighed med minimum rettighedsniveau til at læse/se k-punkter mv.
- Anvende logninger, så kræves operatørrettighed til Logsøgning og Poller.

6. Authentication

For at kunne authenticate mod WebAPI, kræver det at operatøren man bruger, har en adgangskode og har nødvendige rettigheder i UniLock.

6.1 Basic authentication

UniLock WebAPI anvender Basic authentication fx ved standard HTTP requests.

For at lave valide request til WebAPI skal HTTP request indeholde [Authorization] header, med indholdet *"Basic {Login oplysninger}"*.

Loginoplysningerne skal være en tekst streng med formatet *"{username}:{password}"* og være Base64Encoded med tegnsættet "UTF-8".

Fx:

Username = "test user"

Password = 123456

Header = navn: "Authorization", værdi: "Basic dGVzdCB1c2VyOjEyMzQ1Ng=="

6.2 Bearer authentication: OpenID Connect

UniLock WebAPI benytter et certificeret OpenID Connect provider library (IdentityServer3[1]), hvilket gør at UniLock kan fungere som en identity provider til integrerende klienter.

For at authenticate op mod WebAPI findes en række certificerede klientbiblioteker hos OpenID Connect[2].

WebAPI authorityserveren findes på addressen "https://[IP-adresse]:[Port]/auth/v1/".
Addressen anvendes i OpenID klienter, som derefter selv finder nødvendig information.

6.2.1 Understøttede authentication flows

WebAPI understøtter på nuværende tidspunkt følgende OpenID authentication flow[3]

- Implicit
- Authorization Code
- Hybrid

Derudover understøttes også PKCE (rfc7636) flow "Authorization Code" og "Hybrid", som konfigures ved registrering af klienten.

Man kan kun registrere et flow per klient, grundet det biblotek vi benytter's begrænsninger.

6.2.2 Klient registrering

Anvendelse af OpenID kræver at klienter er authorized/trusted, hvilket opnås ved at:

[1] https://github.com/IdentityServer/IdentityServer3
[2] https://openid.net/developers/certified/
[3] https://openid.net/specs/openid-connect-core-1_0.html#Authentication

1. I UniLock Webadgang logger en administrator-operatør ind og åbner [Menu], [Indstillinger], [Opret ny klient] hvilket udløser et registreringstokentoken. Token passer kun til den pågældende UniLock installation og har en levetid på 2 timer.
2. Klienter registrerer sig med det genererede token.
3. Klienter anvender fremadrettet det returnerede client_id og client_secret i kommunikationen med WebAPI, og anvender det returnerede registration_access_token til senere konfiguraiton af klienten.

Klientregistreringen følger:
- OpenID Connect Dynamic Client Registration 1.0
- OAuth 2.0 Dynamic Client Registration Protocol (rfc7591)
- OAuth 2.0 Dynamic Client Registration Management Protocol (rfc7592)

Registreringen foregår på addressen "https://[IP-adresse]:[Port]/auth/v1/connect/register", men kan også findes igennem discovery dokumentet som feltet "registration_endpoint".

6.2.2.1 Klient informationer

UniLocks repræsentation af en klient, kan man finde i json skemaerne eller xsd.
Men generelt følger den meget af det fra openID dokumentet, med enkelte tilføjelser.

Specielle felter:
Ud over nogle af de standard felter fra openID, så indeholder klient informations objekterne også nogle ekstra parametere:
- "enabled":
  Beskriver om klienten er slået til, og kan bruges aktivt i systemet eller ej.
  Klienter kan ikke selv slå den til eller fra, dette er kun UniLock selv som kan.
- "require_proof_key":
  Bruges til at bestemme om det er PKCE varianten af authentication flowet, der skal bruges (undtagen implicit).

Mapping til flow:
For at udvælge hvilket authentication flow man gerne vil benytte med sin klient, skal man sørge for at klientens data indeholder så det matcher følgende skema.

grant_types:		**response_types:**			**Authentication flow:**
implicit:	**authentication_code:**	**code:**	**token:**	**id_token:**	
true	false	default			Implicit
false	true	true	false	False	Authorization code
		true	true/false	true/false	Hybrid
		default			Authorization code
default		default			Authorization code

6.2.2.2 Opret

For at oprette en klient, skal klienten sende et "POST: …/auth/v1/connect/register".

Header:
- Authorization: "Bearer {registrerings token}":

Registrerings token er det token som er udstedt af UniLock Webadgang, til registrering af en ny klient (Husk den begrænsede levetid på 2 timer).

Minimum informationer for at oprette en klient, er ”client_name” og en valid ”redirect_uri”.

I svaret fra UniLock, vil være det fulde klient information objekt, som UniLock har registered og gemt.

6.2.2.3 Hent

Klienter kan hente deres registrerede informationer med ”GET: …/auth/v1/connect/register/:id”, hvor Id’et er klientens eget “client_id” (det virker ikke til at hente andre klienters informationer).

Header:

- Authorization: ”Bearer {klient token}”:
  klient token er det token “registration_access_token” som blev retuneret sammen med “client_secret” og “client_id” som svar på oprettelsen.

I svaret fra UniLock, vil være det fulde klient information objekt, undtagen felter som “registration_access_token” og “registration_client_uri”.

6.2.2.4 Opdatér

Klienter kan opdatere deres informationer med “PUT: .../auth/v1/connect/register/:id”,
hvor Id’et er klientens eget “client_id”.
Header:

- Authorization: ”Bearer {klient token}”:
  klient token er det token “registration_access_token” som blev retuneret sammen med “client_secret” og “client_id” som svar på oprettelsen.

I svaret fra UniLock, vil være det fulde klient information objekt, som UniLock har registered og gemt.

Vigtigt:
Ved opdatering af klientens information, vil der blive udstedt nyt “registration_access_token” og “client_secret” som skal tages i brug, da det eksisterende ikke længere kan bruges.

6.2.2.5 Slet

Klienter kan slette deres informationer med “DELETE: .../auth/v1/connect/register/:id” ,
hvor Id’et er klientens eget “client_id”.
Header:

- Authorization: ”Bearer {klient token}”:
  klient token er det token “registration_access_token” som blev retuneret sammen med “client_secret” og “client_id” som svar på oprettelsen.

6.2.3 Klient administration

For at administrere klienter i UniLock's WebAdgang, skal man være logget ind med en operatør, og have valgt at benytte administrator gruppen.

For at finde siden til administrationen, kan man via adgangseditoren åbne vinduet Indstillinger, hvor man i menuen så vælger [Indstillinger], [PollerService], [Webserver], [WebAPI]. og herefter kan trykke på [Åbn] under punktet OpenID/Bearer Authentication.
Man kan også bare åbne webadgangen, og trykke på [Menu].[Indstillinger] i toppen af siden.

Når man er inde på panelet med listen over klienter, har man her muligheden for:

- Generere et nyt registrations token, til oprettelse af nye klienter
- Aktivere/deaktivere klienter
- Slette klienter
- Tildele en ny "access_token" til klienter, hvis de har mistet deres token/adgang.

6.3 Valg af specifik operatørgruppe

WebAPI understøtter muligheden for at vælge en specifik operatørgruppe til sine http requests.
For at vælge en operatørgruppe til sit request, kan man tilføje en header med navnet "OperatorGroup" og værdien værende operatørgruppens guid.

Det er ikke krævet for at kunne lave et request, men hvis operatøren er medlem af flere operatørgrupper, er det nødvendigt for at garantere at den altid bruger den operatørgruppe man forventer.

6.4 Hent nuværende operatør

Som et led i at man kan være medlem af flere operatørgrupper, kan det være nødvendigt at kunne se hvilke operatørgrupper, det nuværende login har mulighed for at bruge.

De informationer er altid tilgængelige ved at hente den nuværende login's operatør med request til "GET: …/setup/operator/current".
I operatør objektet, finder man listen af medlemskaber, som man kan bruge til at vælge en specifik operatørgruppe som beskrevet ovenfor.

7. K-punkter

Her beskrives datakommunikationen vedrørende objekttypen ”Controlpoint” (k-punkt/dør).

Det er muligt både at oprette/hente/ændre/slette k-punkter samt at modtage nuværende status.

7.1 Opsætning

K-punkters opsætning tilgås via: ”https://[IPadresse]:[Port]/api/v1/setup/controlpoint/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

7.1.1 Opret

Klienter kan oprette nye k-punkter med ”POST: …/setup/controlpoint/”.

Minimum informationer for at oprette et nyt k-punkt er Name og IdNumber, hvor øvrige objekt-informationer automatisk indstilles til default værdier.

7.1.2 Hent

Klienter kan forespørge på et specifikt k-punkt med ”GET: …/setup/controlpoint/:id”.

Klienter kan forespørge på en samling af k-punkter med ”GET: …/setup/controlpoint/”.

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Klienten vil maksimalt modtage 50 fulde (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging anvendes i tilfælde af at systemet indeholder flere objekter (svarets header indeholder ”TotalRecordCount”, som angiver det totale antal objekter i systemet).

7.1.3 Opdatér

Klienter kan opdatere et k-punkt delvist med ”PATCH: …/setup/controlpoint/:id”.

Patch virker som en delvis opdatering, hvor kun medsendt information opdateres.

7.1.4 Slet

Klienter kan slette et k-punkt med ”DELETE: …/setup/controlpoint/:id”.

7.1.5 Eksempel

Eksempel på minimum mængde af informationer i et POST kald for at oprette et nyt k-punkt

Body:

JSON:

```
{
    "Name": "Name",
    "Device": {
        "Identification": {
            "Polling": {
                "IdNumber": "4097"
            }
        }
    }
}
```

IdNumber anvendes som en hexadecimal værdi og angives med decimalværdien af den ønskede hexadecimale værdi, hvor IdNumber=4097 i WebAPI omregnes til det anvendte k-punkt ID-nummer=1001 i UniLock.

7.2 Status

Status for k-punkter er online status, aktuelle driftsparametre, niveau på indgange og udgange, DAS status, tidspunkt for seneste check mv.

K-punkters status tilgås via: ”https://[IP-adresse]:[Port]/api/v1/operation/controlpoint/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Klienter kan hentes status for et specifikt k-punkt med

”GET: …/operation/controlpoint/:id”

Klienter kan hente status for en samling af k-punkter med

”GET: …/operation/controlpoint/”

Klienten vil maksimalt modtage status for 50 (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging anvendes i tilfælde af at systemet indeholder flere objekter (svarets header indeholder ”TotalRecordCount”, som angiver det totale antal objekter i systemet).

7.3 Kommandoer

Klienter kan sende kommandoer til k-punkter med HTTP request method ”PUT”.

7.3.1 Oplås dør

Klienter kan oplåse døren i døroplåsningstiden, hvorefter døren automatisk låses igen, med kommandoen: ”PUT: …/operation/controlpoint/:id/unlockthenlock”.

Fx:

PUT “https://[IP-adresse]:[Port]/api/v1/operation/controlpoint/05c11e2e-94a6-43b0-b8b0-e312de7402b9/unlockthenlock”

7.3.2 DAS frakobling

Klienter kan frakoble DAS (Decentral Alarm Styring) i et k-punkt med kommandoen:

”PUT: …/operation/controlpoint/:id/ dasdisarm”.

7.3.3 DAS tilkobling

Klienter kan tilkoble DAS (Decentral Alarm Styring) i et k-punkt med kommandoen:

”PUT: …/operation/Controlpoint/:id/ dasarm”.

7.3.4 DAS tidsforskudt tilkobling

Klienter kan tidsforskyde den normale tilkobling af DAS (Decentral Alarm Styring) til et angivet tidspunkt i et k-punkt med kommandoen:

”PUT: …/operation/Controlpoint/:id/ dasarmingtimeoverride?timeofday=HH:mm”.

8. Tidstabeller

Her beskrives datakommunikationen vedrørende objekttypen ”TimeTable” (tidstabeller).

Det er muligt både at oprette/hente/ændre/slette tidstabeller.

Tidstabellers opsætning tilgås via: ”https://[IPadresse]:[Port]/api/v1/setup/timetable/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

8.1 Opret

Klienter kan oprette nye tidstabeller med ”POST: …/setup/timetable/”.

Minimum informationer for at oprette en ny tidstabel er Name, hvor øvrige objekt-informationer automatisk indstilles til default værdier.

8.2 Hent

Klienter kan forespørge på en specifikt tidstabel med ”GET: …/setup/timetable/:id”.

Klienter kan forespørge på en samling af tidstabeller med ”GET: …/setup/timetable/”.

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Klienten vil maksimalt modtage 100 fulde (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging anvendes i tilfælde af at systemet indeholder flere objekter (svarets header indeholder ”TotalRecordCount”, som angiver det totale antal objekter i systemet).

8.3 Opdatér

Klienter kan opdatere en tidstabel delvist med ”PATCH: …/setup/timetable/:id”.

Patch virker som en delvis opdatering, hvor kun medsendt information opdateres.

8.4 Slet

Klienter kan slette en tidstabel med ”DELETE: …/setup/timetable/:id”.

9. Sikkerhedsniveautidstabeller

Her beskrives datakommunikationen vedrørende objekttypen ”SecurityLevelTimeTable” (sikkerhedsniveautidstabeller).

Det er muligt både at oprette/hente/ændre/slette sikkerhedsniveautidstabeller.

Sikkerhedsniveautidstabellers opsætning tilgås via:
”https://[IPadresse]:[Port]/api/v1/setup/securityleveltimetable/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

9.1 Opret

Klienter kan oprette nye sikkerhedsniveautidstabeller med

”POST: …/setup/securityleveltimetable/”.

Minimum informationer for at oprette en ny sikkerhedsniveautidstabel er Name, hvor øvrige objektinformationer automatisk indstilles til default værdier.

9.2 Hent

Klienter kan forespørge på en specifik sikkerhedsniveautidstabel med

”GET: …/setup/securityleveltimetable/:id”.

Klienter kan forespørge på en samling af sikkerhedsniveautidstabeller med

”GET: …/setup/securityleveltimetable/”.

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Klienten vil maksimalt modtage 100 fulde (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging anvendes i tilfælde af at systemet indeholder flere objekter (svarets header indeholder ”TotalRecordCount”, som angiver det totale antal objekter i systemet).

9.3 Opdatér

Klienter kan opdatere en sikkerhedsniveautidstabel delvist med

”PATCH: …/setup/securityleveltimetable/:id”.

Patch virker som en delvis opdatering, hvor kun medsendt information opdateres.

9.4 Slet

Klienter kan slette en sikkerhedsniveautidstabel med
”DELETE: …/setup/securityleveltimetable/:id”.

10. Delte undtagelser

Her beskrives datakommunikationen vedrørende objekttypen ”SharedException” (delte undtagelser).

Det er muligt både at oprette/hente/ændre/slette delte undtagelser.

delte undtagelsers opsætning tilgås via: ”https://[IPadresse]:[Port]/api/v1/setup/sharedexception/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

10.1 Opret

Klienter kan oprette nye delte undtagelser med

”POST: …/setup/sharedexception/”.

Minimum informationer for at oprette en ny delte undtagelse er Name, hvor øvrige objekt-informationer automatisk indstilles til default værdier.

10.2 Hent

Klienter kan forespørge på en specifik delte undtagelse med

”GET: …/setup/sharedexception/:id”.

Klienter kan forespørge på en samling af delte undtagelser med

”GET: …/setup/sharedexception/”.

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Klienten vil maksimalt modtage 100 fulde (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging anvendes i tilfælde af at systemet indeholder flere objekter (svarets header indeholder ”TotalRecordCount”, som angiver det totale antal objekter i systemet).

10.3 Opdatér

Klienter kan opdatere en delte undtagelse delvist med

”PATCH: …/setup/sharedexception/:id”.

Patch virker som en delvis opdatering, hvor kun medsendt information opdateres.

10.4 **Slet**

Klienter kan slette en delte undtagelse med
”DELETE: …/setup/sharedexception/:id”.

11. Lokaliteter

Her beskrives datakommunikationen vedrørende objekttypen ”Gateway” (lokalitet).

Det er muligt både at oprette/hente/ændre/slette lokaliteter samt at modtage nuværende status.

11.1 Opsætning

lokaliteters opsætning tilgås via: ”https://[IPadresse]:[Port]/api/v1/setup/gateway/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

11.1.1 Opret

Klienter kan oprette nye lokaliteter med ”POST: …/setup/gateway/”.

Oprettelsen af en ny lokalitet, kræver alle objekt-informationerne udfyldt i forespørgslen.

11.1.2 Hent

Klienter kan forespørge på en specifik lokalitet med ”GET: …/setup/gateway/:id”.

Klienter kan forespørge på en samling af lokaliteter med ”GET: …/setup/gateway/”.

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Klienten vil maksimalt modtage 100 fulde (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging anvendes i tilfælde af at systemet indeholder flere objekter (svarets header indeholder ”TotalRecordCount”, som angiver det totale antal objekter i systemet).

11.1.3 Opdatér

Klienter kan opdatere en lokalitet med ”PUT: …/setup/gateway/:id”.

Alle objekt-informationer skal medsendes.

11.1.4 Slet

Klienter kan slette en lokalitet med ”DELETE: …/setup/gateway/:id”.

11.1.5 Eksempel

Eksempel på minimum mængde af informationer i et POST kald for at oprette en ny lokalitet.

Body:

JSON:


```
{
    "Guid": "752e9150-b9ac-4801-b197-59a35207e109",
    "Name": "Lager Vest",
    "GatewayType": "IP",
    "Polling": {
        "Enabled": false,
        "Method": "Continuous",
        "Interval": "00:00",
        "Start": "00:00",
        "End": "00:00",
        "AllDay": true
    },
        "Device": {
            "DeviceType": "CV72V2",
            "COMSetup": null,
            "USBSetup": null,
            "IPSetup": {
                "HostName": "CV72LagerVest",
                "IPPort": 7211,
                "Password": "",
                "EncryptionKey": ""
            },
            "RemoteModem": null
        },
    "Note": "Placeret i teknikrum R215"
}
```


11.2 Status

Status for lokaliteter er online status, aktuelle driftsparametre, tidspunkt for seneste check mv.

Lokaliteters status tilgås via: ”https://[IP-adresse]:[Port]/api/v1/operation/gateway/”

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Klienter kan hentes status for en specifik gateway med

”GET: …/operation/gateway/:id”

Klienter kan hente status for en samling af lokaliteter med

”GET: …/operation/gateway/”

Klienten vil maksimalt modtage status for 50 (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging anvendes i tilfælde af at systemet indeholder mere flere objekter (svarets header indeholder ”TotalRecordCount”, som angiver det totale antal objekter i systemet).

12. Lokalitetsforbindelser

Her beskrives datakommunikationen vedrørende objekttypen ”GatewayConnector” (lokalitsforbindelse).

Det er muligt at modtage den nuværende status.

12.1 Status

Status for lokalitetsforbindelser er online status, aktuelle driftsparametre mv.

Lokaliteters status tilgås via: ”https://[IP-adresse]:[Port]/api/v1/operation/gatewayconnector/”

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Klienter kan hentes status for en specifik gateway med

”GET: …/operation/gatewayconnector/:id”

Klienter kan hente status for en samling af lokaliteter med

”GET: …/operation/gatewayconnector/”

Klienten vil maksimalt modtage status for 50 (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging anvendes i tilfælde af at systemet indeholder mere flere objekter (svarets header indeholder ”TotalRecordCount”, som angiver det totale antal objekter i systemet).

13. Personer

Her beskrives datakommunikationen vedrørende objekttypen ”Person” (personer).

Det er muligt både at oprette/hente/ændre/slette personer.

Personers opsætning tilgås via: ”https://[IPadresse]:[Port]/api/v1/setup/person/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

13.1 Opret

Klienter kan oprette nye personer med

”POST: …/setup/person/”.

Minimum informationer for at oprette en ny person er Name, hvor øvrige objekt-informationer automatisk indstilles til default værdier.

13.2 Hent

Klienter kan forespørge på en specifik person med

”GET: …/setup/person/:id”.

Klienter kan forespørge på en samling af personer med

”GET: …/setup/person/”.

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Klienten vil maksimalt modtage 100 fulde (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging anvendes i tilfælde af at systemet indeholder flere objekter (svarets header indeholder ”TotalRecordCount”, som angiver det totale antal objekter i systemet).

13.3 Opdatér

Klienter kan opdatere en person delvist med

”PATCH: …/setup/person /:id”.

Patch virker som en delvis opdatering, hvor kun medsendt information opdateres.

13.4 Slet

Klienter kan slette en person med
”DELETE: …/setup/person/:id”.

13.5 Flyt

Klienter kan flytte en person til en anden afdeling med
”PATCH: …/setup/person/:id/MoveToDepartment”

Parametre der sendes med som Query parameter:

- departmentId [string]: id på den afdeling personen skal flyttes til.
- preserveAccess [true/false]: angiver om man vil bevare adgangsrettigheder til personen, efter personen er flyttet til den anden afdeling.
    - Hvis true: personen får oprettet en deling med den afdeling personen bliver flyttet fra. Evt. adgangsrettigheder i nuværende afdeling bevares og kan redigeres.
    - Hvis false (default): personen bliver ikke delt med den afdeling personen flyttes fra. Evt. adgangsrettigheder i nuværende afdeling vil blive fjernet.

Det er kun afdelingsadministratorer der kan flytte en person til en anden afdeling. En system administrator kan ikke. Derudover er kun muligt at flytte en person hvis:

1. Personen er i samme afdeling som den afdelingsadministratoren administrer.
2. Personen flyttes til en anden afdeling end ens egen. Dog kan man ikke flytte en person til system afdelingen.
3. Personen er ikke oprettet/vedligeholdt af en integration. Her skal integrationen slettes først.

14. Persongrupper

Her beskrives datakommunikationen vedrørende objekttypen ”PersonGroup” (persongrupper).

Det er muligt både at oprette/hente/ændre/slette persongrupper.

Persongruppers opsætning tilgås via: ”https://[IPadresse]:[Port]/api/v1/setup/persongroup/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

14.1 Opret

Klienter kan oprette nye persongrupper med

”POST: …/setup/persongroup/”.

Minimum informationer for at oprette en ny persongruppe er Name, hvor øvrige objekt-informationer automatisk indstilles til default værdier.

14.2 Hent

Klienter kan forespørge på en specifik persongruppe med

”GET: …/setup/persongroup/:id”.

Klienter kan forespørge på en samling af persongrupper med

”GET: …/setup/persongroup/”.

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Klienten vil maksimalt modtage 100 fulde (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging anvendes i tilfælde af at systemet indeholder flere objekter (svarets header indeholder ”TotalRecordCount”, som angiver det totale antal objekter i systemet).

14.3 Opdatér

Klienter kan opdatere en persongruppe delvist med

”PATCH: …/setup/persongroup/:id”.

Patch virker som en delvis opdatering, hvor kun medsendt information opdateres.

14.4 **Slet**

Klienter kan slette en persongruppe med
”DELETE: …/setup/persongroup/:id”.

15. Operatører

Her beskrives datakommunikationen vedrørende objekttypen ”Operator” (operatør).

Det er muligt både at oprette/hente/ændre/slette operatører.

Operatørernes opsætning tilgås via: ”https://[IPadresse]:[Port]/api/v1/setup/operator/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

15.1 Restriktioner

Operatører er delt op i forskellige typer, som hver især har sine egne restriktioner:

BuiltIn:

Operatører af denne type, er ”readonly” og kan derfor ikke oprettes, ændres eller slettes. Operatørtypen bruges internt til at beskrive eksempelvis at ”importkanalen” har oprettet en ny person mv.

Administrator:

Operatører af denne type kan kun delvists redigeres. Er forespørgslen lavet med Administratorens brugernavn og adgangskode, så kan Administrator’s felter redigeres, dog med undtagelse af Navn.

UniLock: (Default**)**

Operatører af denne type kan oprettes, ændres og slettes. For at kunne gøre dette, kræves det at operatøren som er benyttet til authentication, har de nødvendige rettigheder for at kunne ændre operatøren tilsvarende i adgangseditoren.

ActiveDirectory:

Operatører af denne type kan slettes og delvist ændres, men hvis Active Directory import stadig er aktiveret vil denne altid overskrive operatørens data med data fra Active Directory.

Navnet, kontooplysningerne og operatørrettighederne kan ikke ændres, da det er styret af Active Directory import.

Resten af felterne kan ændres, hvis man har de nødvendige rettigheder for at kunne ændre operatøren tilsvarende i adgangseditoren.

15.2 Opret

Klienter kan oprette nye operatører med

”POST: …/setup/operator/”.

Nødvendige informationer for at oprette en ny operatør er Name og Username, hvor øvrige objekt-informationer automatisk indstilles til default værdier.

15.3 Hent

Klienter kan forespørge på deres nuværende operatør for forespørgslen med

”GET: …/setup/operator/current”.

Klienter kan forespørge på en specifik operatør med

”GET: …/setup/operator/:id”.

Klienter kan forespørge på en samling af operatører med

”GET: …/setup/operator/”.

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Klienten vil maksimalt modtage 100 fulde (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging (Se afsnit 4.4 Paging) anvendes i tilfælde af at systemet indeholder flere objekter.

15.4 Opdatér

Klienter kan opdatere en operatør delvist med

”PATCH: …/setup/operator/:id”.

Patch virker som en delvis opdatering, hvor kun medsendt information opdateres.

15.5 Slet

Klienter kan slette en operatør med

”DELETE: …/setup/operator/:id”.

16. Operatørgrupper

Her beskrives datakommunikationen vedrørende objekttypen ”OperatorGroup” (operatørgruppe).

Det er muligt at hente operatørgrupper som en liste af summary objekter.

Operatørgruppernes opsætning tilgås via: ”https://[IPadresse]:[Port]/api/v1/setup/operatorgroup/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

16.1 Hent

Klienter kan forespørge på en specifik operatørgruppe med

”GET: …/setup/operatorgroup/:id”.

Klienter kan forespørge på en samling af operatørgrupper med

”GET: …/setup/operatorgroup/”.

Parametre:

- Skip [string]
- Take [string]

17. Specialdagskalender

Her beskrives datakommunikationen vedrørende objekttypen ”SpecialDayCalender” (Specialdagskalender).

Det er muligt at hente specialdagskalendere som en liste af summary objekter.

Specialdagskalenderens opsætning tilgås via:
”https://[IPadresse]:[Port]/api/v1/setup/specialdaycalender/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

17.1 Hent

Klienter kan forespørge på en specifik specialdagskalender med

”GET: …/setup/specialdaycalender/:id”.

Klienter kan forespørge på en samling af specialdagskalendere med

”GET: …/setup/specialdaycalender/”.

Parametre:

- Skip [string]
- Take [string]

18. Afdelinger

Her beskrives datakommunikationen vedrørende objekttypen ”Departments” (Afdelinger).

Det er muligt at hente afdelinger som en liste af summary objekter.

Afdelingens opsætning tilgås via: ”https://[IPadresse]:[Port]/api/v1/setup/department/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

18.1 Hent

Klienter kan forespørge på en specifik afdeling med

”GET: …/setup/department/:id”.

Klienter kan forespørge på en samling af afdelinger med

”GET: …/setup/department/”.

Parametre:

- Skip [string]
- Take [string]

19. Logninger

Her beskrives datakommunikationen vedrørende objekttypen ”Log”.

Det er kun muligt at hente logninger.

Logninger tilgås via: ”https://[IPadresse]:[Port]/api/v1/logging/log/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

Vigtigt:

- Elementer kan tilføjes og fjernes fra [LogType] uden at det betegnes som en ”breaking change”, hvorfor det anbefales at der tages hånd om ukendte logtyper i integrerende klienter.
- Logninger benytter ikke summaryobjekter og returnerer ikke en TotalRecordCount, som de andre endpoints gør.
- ObjectReferences findes kun på logninger fra installationsdatoen og frem.

19.1 Hent

Klienter kan forespørge på en specifik logning med

”GET: …/logging/log/:id”.

Parametre:

- IncludeDescription [true/false]
  true: Logninger skal også indeholde tekstbeskrivelser

Klienter kan forespørge på en samling af logninger med

”GET: …/logging/log/”.

Parametre:

- Skip [int]
- Take [int]
- IncludeDescription [bool]: *true: Logninger skal også indeholde tekstbeskrivelser.*
- FromDate [datetime]: *format: “yyyy-MM-dd”*
- ToDate [datetime]: *format: “yyyy-MM-dd”*
- FromTimeOfDay [datetime]: *format: “HH:mm”*
- ToTimeOfDay [datetime]: *format: “HH:mm”*
- LogType [string]: *komma separeret streng.*
- ControlPointName [string]:*Kan indeholde wildcard ’*’ i start eller slutning af teksten*
- ControlPoint [Guid]
- PersonName [string]:*Kan indeholde wildcard ’*’ i start eller slutning af teksten*
- Person [Guid]

- FromId [long]
- logalarmresetid [long]
- fromlogalarmresetid [long]
- OrderDirection [string]: (asc/desc*) : standard er desc*

20. Alarmer

Her beskrives datakommunikationen vedrørende skærmalarmer som benytter objekttypen ”Log”.

Det er muligt at hente og afstille alarm logninger.

Alarmer tilgås via: ”https://[IPadresse]:[Port]/api/v1/logging/alarm/”.

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

20.1 Hent

Klienter kan forespørge på alle de aktive alarm logninger med

”GET: …/logging/alarm/”.

Parametre:

- IncludeDescription [true/false]
  true: Logninger skal også indeholde tekstbeskrivelser

20.2 Afstil

Klienter kan afstille aktive alarm logninger med
”POST: …/logging/alarm”.

I forespørgslens indhold skal klienter sende en liste af de alarmlog Id’er, som skal afstilles.

21. Områder

Her beskrives datakommunikationen vedrørende objekttypen ”Area” (område).

Det er muligt at modtage den nuværende status for områder.

21.1 Status

Status for områder er antal personer i området, seneste hændelse og udløbstider for personer i området mv.

Områders status tilgås via: ”https://[IP-adresse]:[Port]/api/v1/operation/area/”

Overordnet anvendes disse headere:

- Accept:
  - Application/json
- Content-Type:
  - Application/json
- Authorization:
  - Basic [Base64Encoded login oplysninger]

Parametre:

- Skip [string]
- Take [string]
- includeFullObject [true/false]

Vigtigt:
Områder benytter ikke standart summary objekter, men sit eget specielle summary objekt.
For mere info, referes der til XSD’en eller Json Schema’erne.

Klienter kan hentes status for et specifikt område med:

”GET: …/operation/area/:id”

Klienter kan hente status for en samling af områder med:

”GET: …/operation/area/”

Klienten vil maksimalt modtage status for 50 (Query parameteren ”includeFullObject=true”) objekter ad gangen, hvor paging anvendes i tilfælde af at systemet indeholder flere objekter (svarets header indeholder ”TotalRecordCount”, som angiver det totale antal objekter i systemet).