

Rollekataloget

Anvendelse af API

Date: 18.02.2022

Author: BSG

Indhold

1	Indledning	3
1.1	Forudsætninger.....	3
2	API dokumentation	3
2.1	Input	4
2.2	Output	4
2.3	Request/response eksempel	5
3	Adgang til API'et	5
4	Anvendelse af API'et, et par eksempler	7
4.1	Eksempel på udlæsning af data	7
4.2	Eksempel på opdatering af data (simpel rettighedstildeling)	8
4.3	Eksempel på opdatering af it-system (kompleks opdatering)	10
5	Forretningsmæssigt API overblik	12
5.1	AD FS API	13
5.2	Titles API	13
5.3	ItSystem API	13
5.4	Read API	13
5.5	Role Assignment API.....	13
5.6	Organisation Management API	14
5.7	AD Sync API	14

1 Indledning

Dette dokument er en grundlæggende vejledning i anvendelsen af de API'er der er udstillet på OS2rollekatalog. Alle API'er anvendes på samme måde, og den fulde liste over API endpoints og deres input/output er dokumenteret på info-siden på ens rollekatalog-installation.

Man kan altid hente nyeste udgave af dokumentationen, API beskrivelsen og anden relevant dokument på info-siden, som findes her (ret "kommune" til ens eget kommune-navn)

<https://kommune.rollekatalog.dk/info>

1.1 Forudsætninger

API'erne er tiltænkt teknikere og udviklere, og dette dokument forudsætter at man har en teknisk baggrund. Det er ikke en vejledning i brugen af udviklingsværktøjer, men en beskrivelse af hvordan man bruger API'erne vha udviklingsværktøjer.

Alle eksemplerne i dokumentet er udarbejdet som powershell eksempler, men det er ikke en forudsætning at man anvender powershell. Alle sprog og værktøjer er kan anvendes til at udføre REST kald, kan anvendes til at kalde API'erne.

2 API dokumentation

Et link til den konkrete API dokumentation for ens eget rollekatalog, kan findes på info siden som der er linket til ovenfor. Det er nedenstående link som skal anvendes til at tilgå API dokumentationen

Installationsvejledning

Installationsvejledningen kan hentes som en PDF fil nedenfor

- [Hent installationsvejledning Attribute Store](#)
- [Hent installationsvejledning til ADSyncService](#)

API dokumentation

API dokumentationen kan ses online her

- [Se API dokumentation](#)

Information om versionen af rollekataloget

Version: 2022 r1
BuildId: c384091
Timestamp: 2022-01-04

API dokumentationen er indelt i afsnit, og der kan navigeres mellem de enkelte afsnit og de API operationer der findes i hvert afsnit, via venstre-menuen.

Selve API dokumentation er automatisk genereret ud fra test-cases i rollekatalogets kodebase, og strukturen er derfor identisk for hver enkelt API dokumentation. Nedenfor vises et eksempel på en API operation i dokumentationen, hvor de enkelte punkter gennemgås

2.1 Input

Hver operation har en overskrift, der indeholder API operationens navn. I dette tilfælde har vi API operationen "Get Roles as a list", der anvendes til at lave et opslag på en bruger, og få alle de Jobfunktionsroller (userroles) og Brugersystemroller (systemroles) som er tildelt den bruger man laver opslag på.

De første 3 afsnit angiver de input parametre der skal angives, i hhv url-stien (path), http headers og url-parametre (request parameters). I eksemplet længere nede i dokumentationen er det lidt tydeligere hvad der menes med de 3 forskellige typer, og strukturen er identisk for alle API operationer.

Get Roles as a list

This operation will generate a list of userroles and systemroles assigned to the user for a given it-system.

Path Parameters

Table 2. /api/user/{userid}/rolesAsList

Parameter	Description
userid	The users userid (fx: bbog)

Request Headers

Name	Description
ApiKey	Secret key required to call API

Request Parameters

Parameter	Description
system	The identifier of the it-system (fx: SAPA) - if not supplied, all roles for all it-systems are returned

2.2 Output

Efter input-delen, kommer en beskrivelse af det output som API operationen returnerer ved et succesfuldt kald.

Output er altid i JSON formatet, og beskrivelsen angives i en tabel, hvor de enkelte felter beskrives med feltnavn (hvad data kaldes i JSON strukturen), hvilken type af data der er tale om, samt en kort beskrivelse af feltet.

Response Fields

Path	Type	Description
userRoles	Array	List of userroles assigned to the user
dataRoles	Array	List of dataroles assigned to the user
functionRoles	Array	List of functionroles assigned to the user
systemRoles	Array	List of systemroles derived from the list of other roles
nameID	String	Subject NameID in X.509 format
roleMap	Object	Map with id/name of roles in oioBPP structure

2.3 Request/response eksempel

Endeligt kommer et eksempel på et kald, hvor man kan se både input og output i eksempelformat. Dette skulle gerne hjælpe med forståelsen af den ovenstående dokumentation, så man kan sammenstille eksempel-data med felt-dokumentationen.

Da eksemplerne er dannet baseret på automatiske tests, er de ikke nødvendigvis 100% fyldestgørende med eksempel-data, men overordnet viser de strukturen i request/response, og der henvises til felt-dokumentationen for en mere fyldestgørende dokumentation af felter.

Example request

```
GET /api/user/bbog/rolesAsList?system=KOMBIT HTTP/1.1
ApiKey: f7d8ea9e-53fe-4948-b600-fbc94d4eb0fb
Host: www.rollekatalog.dk
```

Example response

```
HTTP/1.1 200 OK
Content-Length: 324
Content-Type: application/json; charset=UTF-8

{
  "nameID" : "C=DK,O=12345678,CN=Bente Børgesen,Serial=6b5bdbbb-9b9d-4636-80e4-dea991cb0e16",
  "userRoles" : [ "KOMBIT_2" ],
  "systemRoles" : [ "http://kombit.dk/roles/usersystemrole/se_sag/1" ],
  "dataRoles" : [ ],
  "functionRoles" : [ ],
  "roleMap" : {
    "KOMBIT_2" : "KOMBIT System role 2 (KOMBIT System)"
  }
}
```

3 Adgang til API'et

Adgang til API'et kræver at man har en API nøgle (der anvendes som ApiKey i HTTP headeren som vist i eksemplerne ovenfor). Man kan danne en sådan via rollekatalogets brugergrænseflade, hvor en administrator har adgang til at oprette og vedligeholde såkaldte klienter.

Det anbefales at man altid opretter en ny API nøgle til nye klienter, og ikke deler API nøgler, da det gør det svært at spore anvendelsen, samt lukke for udgående klienter over tid.

Se klienter

+ Opret klient

100 ▼ rækker per side

Søg

Navn	API nøgle	Rolle	Handler
Administrator	*****	Administrator	🔍 ✎ ✕
Powershell læser	*****	Læseadgang	🔍 ✎ ✕
Powershell skriver	*****	Rolleadministration	🔍 ✎ ✕

Viser 1 til 3 af 3 rækker

Forrige

1

Næste

Når man opretter en klient i rollekatalogets brugergrænseflade, skal man vælge en rolle til klienten. Denne rolle giver adgang til en eller flere dele af API'ets operationer. Følgende roller eksisterer

- Administrator
- Organisation
- Rolleadministratør
- Læseadgang
- Leverandør
- IT-system administrator
- KSP/CICS administrator

Den første rolle (administrator) har fuld adgang til API'et. Det anbefales ikke at anvende denne rolle, men i stedet at anvende en af de mere snævre adgange.

Den anden rolle (organisation), giver kun adgang til de API operationer der bruges til at indlæse (og udlæse) organisationsdata, herunder brugere, enheder og stillinger.

Den tredje rolle (rolleadministratør), giver både læse- og skriveadgang til alle data i rollekataloget, på nær organisationsdata. En rolleadministratør kan dermed anvende alle operationerne til at udlæse data, samt alle operationer der anvendes til at tildele, fratage eller vedligeholde rettigheder i rollekataloget.

Den fjernerrolle (læseadgang), giver adgang til at lave opslag på data. Denne rolle anvendes typisk til eksterne systemer, som har brug for at lave opslag på rettigheder. Dette kan fx være AD FS, eller et lokalt script der opdaterer lokale systemer på baggrund af data i rollekataloget.

Den femte rolle (leverandør), er reserveret til fremtidig funktionalitet, og vil give leverandører adgang til at lave opslag på eget it-system – denne funktionaliteten er dog ikke lavet endnu.

Den sjette rolle (it-system administrator), giver adgang til at administrere it-systemer, og kan bl.a. anvendes til synkronisering af it-system stamdata.

Den syvende rolle (ksp/cics administrator) anvendes til at skifte kodeord på KSP/CICS brugerkonti, og kræver en special opsætning, hvor man skal tage fat i Digital Identity før den kan anvendes.

Bemærk at når man læser API dokumentationen, så har hvert afsnit beskrevet hvilken rolle der er nødvendig for at anvende operationerne i dette afsnit, fx

AD FS API

The following operations are intended for AD FS integration, and expose information about a given user, which can be used to issue SAML assertions containing all relevant information about the user

Required Role

All the operations in the AD FS API are available to any clients that has at least the "Læseadgang" role assigned to them.

4 Anvendelse af API'et, et par eksempler

Nedenstående viser 3 eksempler på at anvende API'et. Den ene operation viser hvordan man læser data ud af rollekataloget, og de to sidste viser hvordan man læser data ind i rollekataloget.

4.1 Eksempel på udlæsning af data

Her har vi et eksempel på hvordan man udlæser stamdata på en rollebuket. Operationen findes under afsnittet "Read API" og hedder "Read one rolegroup". Dokumentationen angiver at vi skal bruge følgende input

Path Parameters

Table 13. /api/read/rolegroups/{id}

Parameter	Description
id	The id of the rolegroup

Request Headers

Name	Description
ApiKey	Secret key required to call API

Og eksemplet på et kald ser sådan her ud

Example request

```
GET /api/read/rolegroups/1 HTTP/1.1
ApiKey: f7d8ea9e-53fe-4948-b600-fbc94d4eb0fb
Host: www.rollekatalog.dk
```

Så hvis vi ønsker at gennemføre samme kald via powershell, så kan det fx gøres vha følgende powershell script

```
# Den fulde sti til den API operation vi ønsker at kalde
$URL = "https://demo.rollekatalog.dk/api/read/rolegroups/"

# ID'et på den rollebuket vi ønsker at lave et opslag på
$ROLEGROUP = "4"

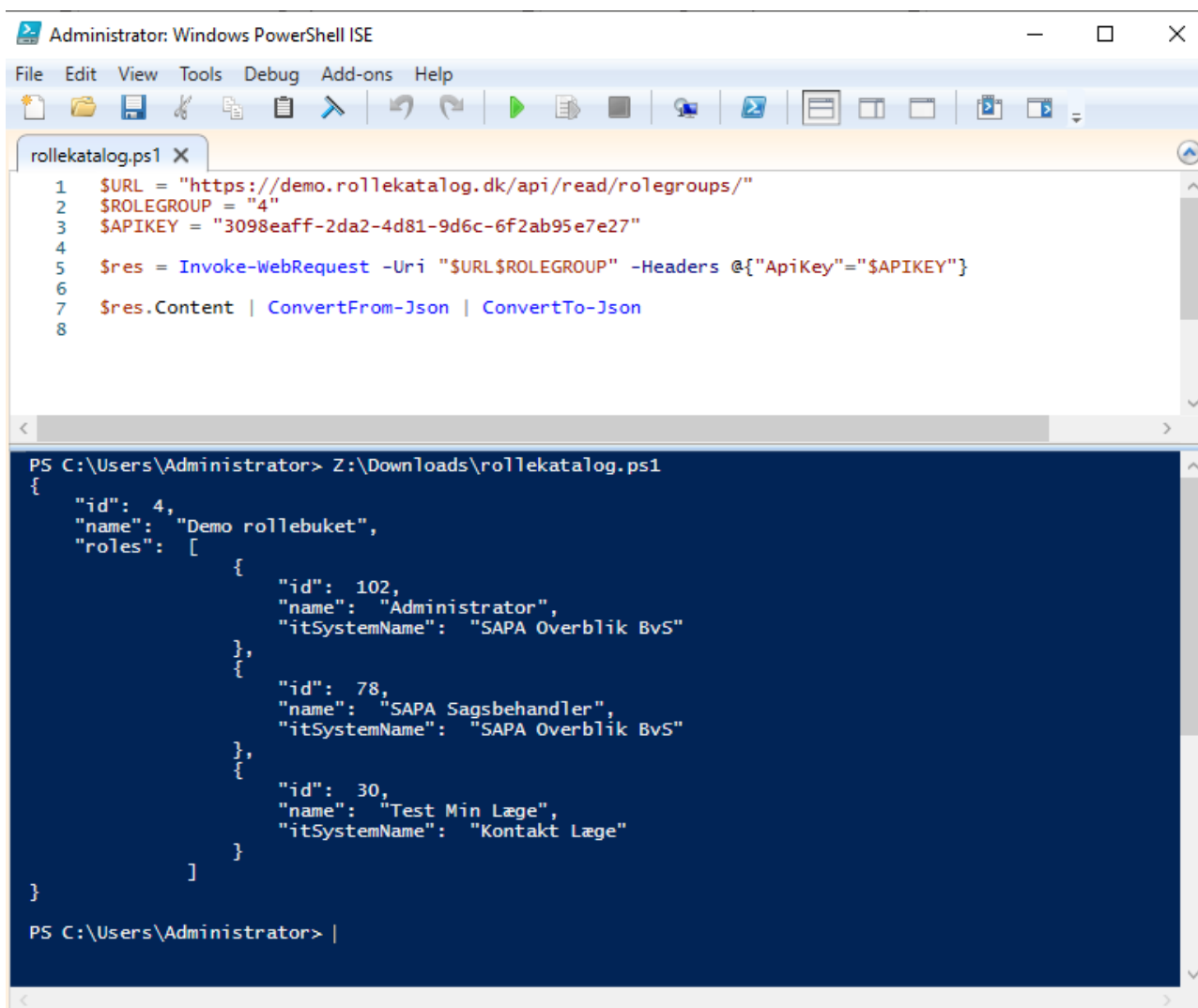
# API nøglen
$APIKEY = "3098eaff-2da2-4d81-9d6c-6f2ab95e7e27"

# Selve kaldet til OS2rollekatalogets API
$res = Invoke-WebRequest -Uri "$URL$ROLEGROUP" -Headers @{"ApiKey"="$APIKEY"}

# Et lille trick til at outputte resultatet som pænt formateret JSON
$res.Content | ConvertFrom-Json | ConvertTo-Json
```

Både URL og APIKEY skal selvfølgelig tilpasses ens eget rollekatalog, og ID'et på den rollebuket man laver opslag på kan så udfyldes med ID'et på en rollebuket fra ens eget rollekatalog.

Nedenfor er vist en afvikling af kaldet via en Powershell ISE



```

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
rollekatalog.ps1 X
1 $URL = "https://demo.rollekatalog.dk/api/read/rolegroups/"
2 $ROLEGROUP = "4"
3 $APIKEY = "3098eaff-2da2-4d81-9d6c-6f2ab95e7e27"
4
5 $res = Invoke-WebRequest -Uri "$URL$ROLEGROUP" -Headers @{"ApiKey"="$APIKEY"}
6
7 $res.Content | ConvertFrom-Json | ConvertTo-Json
8

PS C:\Users\Administrator> Z:\Downloads\rollekatalog.ps1
{
  "id": 4,
  "name": "Demo rollebuket",
  "roles": [
    {
      "id": 102,
      "name": "Administrator",
      "itSystemName": "SAPA Overblik BvS"
    },
    {
      "id": 78,
      "name": "SAPA Sagsbehandler",
      "itSystemName": "SAPA Overblik BvS"
    },
    {
      "id": 30,
      "name": "Test Min Læge",
      "itSystemName": "Kontakt Læge"
    }
  ]
}

PS C:\Users\Administrator> |

```

Outputtet fra kaldet vises i konsollen under ISE'en, hvor man kan se den JSON struktur man får tilbage.

Her kan man se at rollebukketten har 3 jobfunktionsroller i sig (roles sektionen), og hvilke ID'er og navne disse jobfunktionsroller har, samt hvilket ItSystem som jobfunktionsrollerne kommer fra.

4.2 Eksempel på opdatering af data (simpel rettighedstildeling)

Et almindeligt anvendelsesscenarie er at kunne foretage ændringer i tildelte rettigheder i OS2rollekatalog. Her er der bl.a. operationer til at tildele rettigheder direkte på brugere, og et sådan eksempel vises nedenfor

Her er det afsnittet "Role Assignment API" der er interessant, og vi kigger her på muligheden for at tildele en rollebuket (samme som vi lavede opslag på i eksemplet ovenover).

Operationen til dette hedder "Assign rolegroup to user", og dokumentationen siger følgende

Path Parameters

Table 16. /api/user/{userUuid}/assign/rolegroup/{roleGroupId}

Parameter	Description
userUuid	The user UUID or UserId
roleGroupId	The rolegroup id

Request Headers

Name	Description
ApiKey	Secret key required to call API

Example request

```
PUT /api/user/6b5bdbbb-9b9d-4636-80e4-dea991cb0e16/assign/rolegroup/1 HTTP/1.1
ApiKey: f7d8ea9e-53fe-4948-b600-fbc94d4eb0fb
Host: www.rollekatalog.dk
```

Her kan vi gøre brug af følgende powershell script til at foretage tildelingen.

```
# stien til den operation vi ønsker at kalde (bemærk at bruger-id 'bsg' indgår i stien)
$URL = "https://demo.rollekatalog.dk/api/user/bsg/assign/rolegroup/"

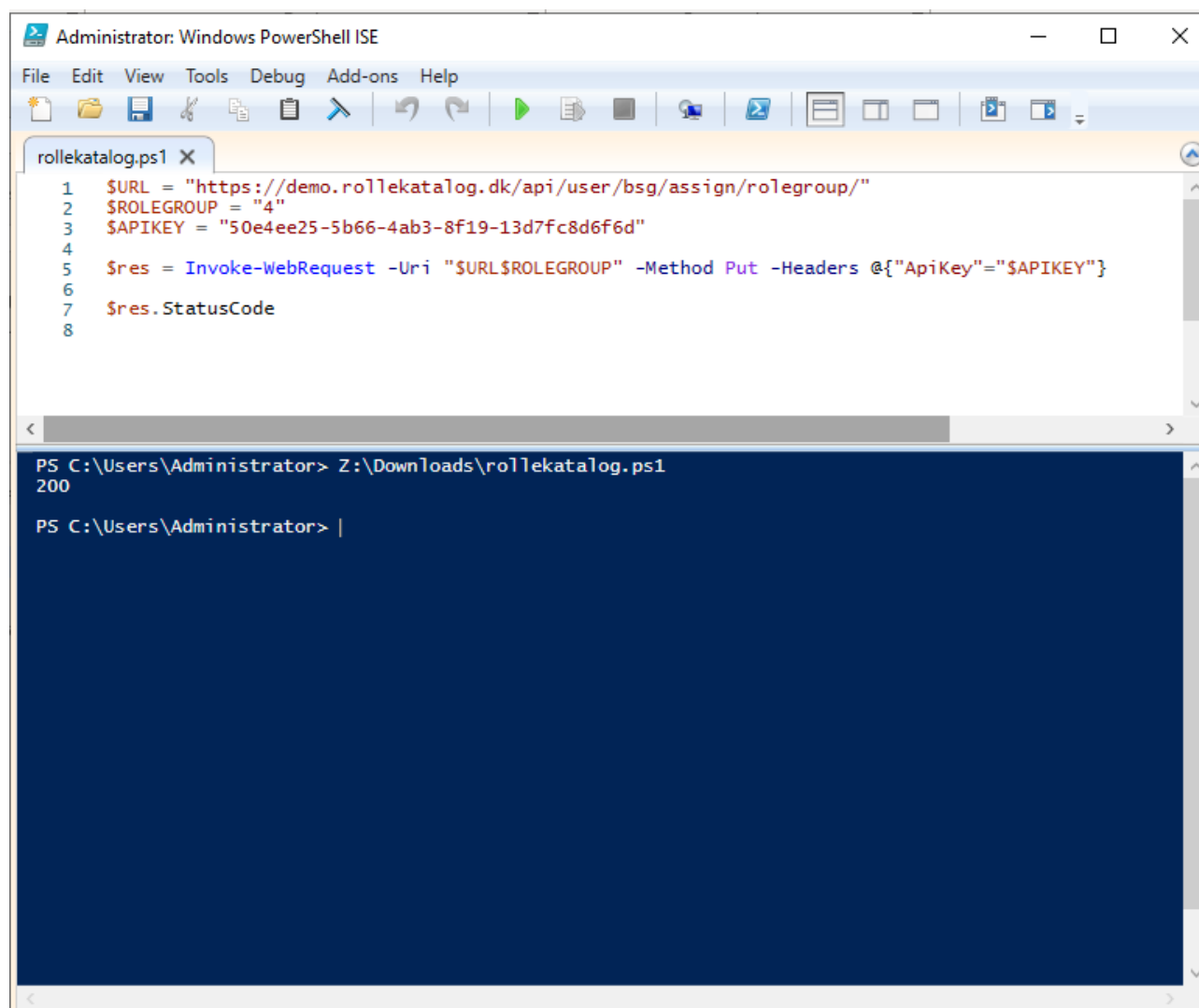
# rollebuketten vi ønsker at tildele
$ROLEGROUP = "4"

# API nøglen
$APIKEY = "50e4ee25-5b66-4ab3-8f19-13d7fc8d6f6d"

# foretag kaldet
$res = Invoke-WebRequest -Uri "$URL$ROLEGROUP" -Method Put -Headers
@{"ApiKey"="$APIKEY"}

# check returstatus
$res.StatusCode
```

Hvis vi afvikler dette i Powershell ISE, får vi følgende resultatet



```

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
rollekatalog.ps1 x
1 $URL = "https://demo.rollekatalog.dk/api/user/bsg/assign/rolegroup/"
2 $ROLEGROUP = "4"
3 $APIKEY = "50e4ee25-5b66-4ab3-8f19-13d7fc8d6f6d"
4
5 $res = Invoke-WebRequest -Uri "$URL$ROLEGROUP" -Method Put -Headers @{"ApiKey"="$APIKEY"}
6
7 $res.StatusCode
8

PS C:\Users\Administrator> Z:\Downloads\rollekatalog.ps1
200

PS C:\Users\Administrator> |
  
```

Som man kan læse af API dokumentationen er det ikke et output, så vi kigger alene på returkoden. Så længe den er 200, så er rettigheden tildelt, og alt er OK.

4.3 Eksempel på opdatering af it-system (kompleks opdatering)

Den sidste operation anvendes til at opdatere et helt it-system, hvor man vedligeholder listen af systemroller som et givent it-system udstiller. Dette anvendes fx hvis man indlæser stamdata om et it-system ind i rollekataloget fra en ekstern master-kilde.

Her skal vi have fat i afsnittet "ItSystem API", og operationen "Update one it-system", hvor dokumentationen til denne operation ser ud som følger

Path Parameters

Table 5. /api/itsystem/manage/{id}

Parameter	Description
id	The id of the it-system

Request Body

Path	Type	Description
name	String	Name of the it-system
identifier	String	Technical ID key for the it-system (not always unique)
systemRoles	Array	rray of systemroles currently on it-system
systemRoles[].name	String	Name of systemrole
systemRoles[].identifier	String	Unique identifier of systemrole
systemRoles[].description	String	Description of systemrole

Request Headers

Name	Description
ApiKey	Secret key required to call API

Her skal vi have opbygget et JSON request, der indeholder ovenstående stamdata der beskriver it-systemet og de systemroller der findes i it-systemet.

Via powershell kan det gøres på følgende måde

```
# stien til den operation vi ønsker at kalde
$URL = "https://demo.rollekatalog.dk/api/itsystem/manage/"

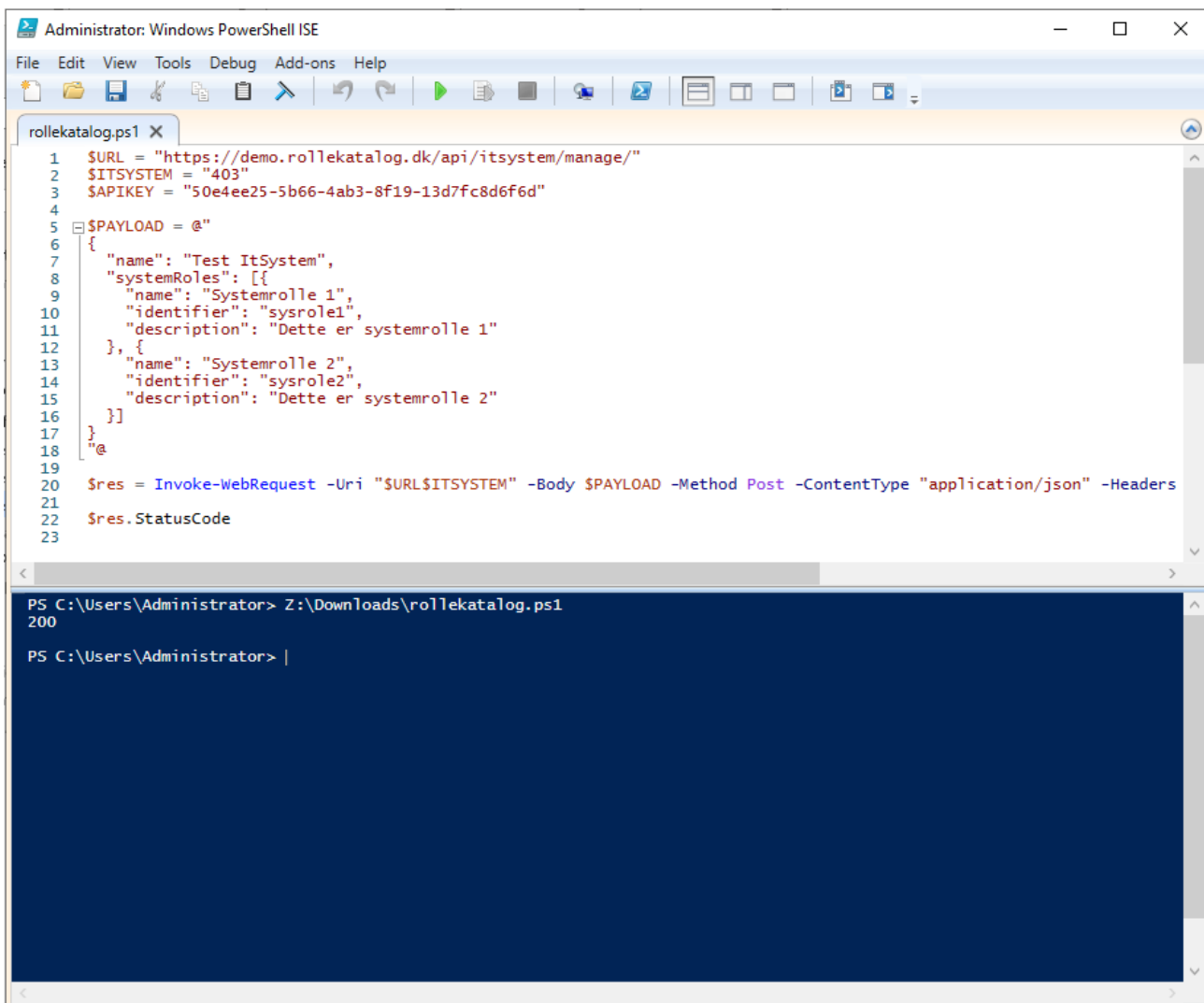
# ID på it-systemet vi ønsker at opdatere
$ITSYSTEM = "403"

# API nøglen
$APIKEY = "50e4ee25-5b66-4ab3-8f19-13d7fc8d6f6d"

# dan det payload der skal opdatere it-systemet med 2 systemroller og et navn
$PAYLOAD = @"
{
  "name": "Test ItSystem",
  "systemRoles": [{
    "name": "Systemrolle 1",
    "identifier": "sysrole1",
    "description": "Dette er systemrolle 1"
  }, {
    "name": "Systemrolle 2",
    "identifier": "sysrole2",
    "description": "Dette er systemrolle 2"
  }]
}
```

```
"@  
  
# foretag kaldet  
$res = Invoke-WebRequest -Uri "$URL$ITSYSTEM" -Body $PAYLOAD -Method Post -ContentType  
"application/json" -Headers @{"ApiKey"="$APIKEY"}  
  
# check returstatus  
$res.StatusCode
```

Og hvis vi forsøger at afvikle dette script i powershell ISE, får vi følgende output



The screenshot shows the Windows PowerShell ISE interface. The script file is named 'rollekatalog.ps1'. The script content is as follows:

```
1 $URL = "https://demo.rollekatalog.dk/api/itsystem/manage/"  
2 $ITSYSTEM = "403"  
3 $APIKEY = "50e4ee25-5b66-4ab3-8f19-13d7fc8d6f6d"  
4  
5 $PAYLOAD = @"  
6 {  
7   "name": "Test ItSystem",  
8   "systemRoles": [{  
9     "name": "Systemrolle 1",  
10    "identifier": "sysrole1",  
11    "description": "Dette er systemrolle 1"  
12   }, {  
13    "name": "Systemrolle 2",  
14    "identifier": "sysrole2",  
15    "description": "Dette er systemrolle 2"  
16   }]  
17 }  
18 @"  
19  
20 $res = Invoke-WebRequest -Uri "$URL$ITSYSTEM" -Body $PAYLOAD -Method Post -ContentType "application/json" -Headers  
21  
22 $res.StatusCode  
23
```

The output pane shows the command execution result:

```
PS C:\Users\Administrator> Z:\Downloads\rollekatalog.ps1  
200  
  
PS C:\Users\Administrator> |
```

5 Forretningsmæssigt API overblik

De udstiller API'er er indelt i afsnit. Hvert afsnit har en række operationer, hvor alle operationer i et afsnit kræver samme rolle for at anvende.

Nedenfor er de enkelte afsnit beskrevet, samt de forretningsmæssige operationer der findes i hvert afsnit. For konkrete detaljer omkring API operationerne skal man tilgå API dokumentationen.

5.1 AD FS API

Disse API'er kræver alle Læseadgang, og ved en fremtidig omstrukturering af API'er vil disse blive grupperet sammen med alle "læse-adgangs" API'erne. Alle de opslag som en AD FS (Identity Provider) kan finde på at foretage mod rollekataloget findes i dette afsnit.

Formelt er det

- Opslag på en brugers rettigheder (generelt opslag til fagsystemer der ikke har noget med KOMBIT at gøre)
- Opslag på en brugers rettigheder, formateret i OIO-BPP format til KOMBIT
- Opslag på en brugers identitet, formateret i X509SubjectName format til KOMBIT

5.2 Titles API

Disse API'er kræver Organisations-adgang, og ved en fremtidig omstrukturering af API'er vil disse blive grupperet sammen med de andre API operationer til organisations-administration.

Disse bruges til at vedligeholde rollekatalogets titel-katalog, og der er almindelige liste-, læse-, oprette-, rette-, slette- operationer til rådighed.

5.3 ItSystem API

Disse API'er kræver Rolleadministrator rollen, og anvendes til at vedligeholde stamdata på it-systemer i rollekataloget.

Der er API operationer til at udlæse it-systemer samt overskrive eksisterende it-systemers stamdata. Der er ingen delta-operationer eller operationer til at slette/oprette it-systemer. Disse ting skal gøres fra brugergrænsefladen.

5.4 Read API

Disse API'er kræver alle Læseadgang rollen, og her er alle operationer til at udlæse data fra rollekataloget samlet. Der er operationer til at udlæse

- Se tildelte rettigheder på en enhed
- Se tildelte rettigheder på en bruger
- Udtrække liste over alle Jobfunktionsroller
- Udtrække liste over alle Jobfunktionsroller for et givent it-system
- Udtrække liste over alle Rollebuketter
- Udtrække stamdata på én Jobfunktionsrolle
- Udtrække stamdata på én Rollebuket
- Se alle brugere der har jobfunktionsroller til et givent it-system, inkl stamdata om rolle-tildelingene
- Se alle brugere der er tildelt én bestemt Jobfunktionsrolle

5.5 Role Assignment API

Disse API'er kræver alle Rolleadministrator rollen, og anvendes til at foretage tildelinger (og fratagelser) af rettigheder. Følgende operationer er tilgængelige

- Tildel jobfunktionsrolle til bruger

- Tildelt jobfunktionsrolle til enhed
- Fratag jobfunktionsrolle fra bruger
- Fratag jobfunktionsrolle fra enhed

5.6 Organisation Management API

Disse API'er kræver alle Organisation rollen, og giver mulighed for at vedligeholde de indlæste stamdata om organisationen i rollekataloget. Følgende operationer er tilgængelige

- Udlæs organisations-hierarkiet (udlæser alle organisationsdata i én stor JSON struktur)
- Indlæs hele organisations-hierarkiet (fuldt load af alle enheder og brugere)
- Indlæs delta-load af brugere (liste af brugere som skal oprettes eller opdateres)

Den sidste operation er tiltænkt ad-hoc oprettelser eller opdateringer til brugere. Den bør altid suppleres med regelmæssige fulde loads, for at sikre at alle enheder er korrekt opdateret, samt at brugere bliver deaktiveret (sker som en side-effekt af det fulde load, hvis en bruger ikke længere er med i det fulde load).

Et typisk integrationsmønster er delta-loads af brugere hvert 5. minut og et fuldt load af hele organisationen hver time, for at sikre at data er korrekt opdateret i rollekataloget.

5.7 AD Sync API

Disse API'er er specifikt designet til ADSyncService, der er den integration der vedligeholder AD gruppemedlemsskaber på baggrund af data i rollekataloget. Det anbefales ikke at anvende disse API'er direkte, men i stedet at anvende ADSyncService til formålet.

API'erne kræver Rolleadministrators rollen, og følgende operationer er tilgængelige

- Hent liste over "beskidte" AD grupper
- Marker AD grupper som værende ajourført i AD
- Hent liste over AD gruppeoperationer (opret/nedlæg)
- Marker AD gruppeoperationer som ajourført i AD

Den første operation returnerer en liste over alle AD grupper som der er rettighedsændringer til. ADSyncService anvender så læse API'erne til at hente alle de relevante data for disse, og ajourføre AD, hvorefter den kalder den anden operation til at angive at den nu har udført opdateringen.

De to sidste operationer er til at understøtte opret/nedlæg AD grupper direkte fra rollekataloget.