Rollekataloget

AuditLog API

Date: 27.07.2022
Author: BSG

Indhold

1 Indledning

Dette dokument beskriver de API snitflader der findes på OS2rollekatalog til at udlæse auditlog records fra systemet.

1.1 Forudsætninger

Der skal være oprettet en klient inde i OS2rollekatalog med AuditlogReader rollen, som anvendes til at kalde API'et. Her oprettes ligeledes API nøglen, som skal angives i en HTTP header ved navn ApiKey når man kalder servicen.

2 Operationer

2.1 HTTP GET /api/auditlog/head

Dette endpoint returnerer ID'et på den nyeste auditlog-record, og formålet med endpointet er at have en hurtig måde at checke om der er kommet nye auditlogs siden man kaldte sidste gang. Denne operation er billig og hurtig at kalde, og det anbefales at man anvender den periodisk til at checke om der er nye data.

Eksempel output

```
{
  "head": 7192
}
```

Hvis man gemmer ID'erne på de auditlog records man har synkroniseret ud lokalt, så kan man nemt verificere om der er nye records i OS2faktor der skal læses ud. ID'erne er stigende, men der er ingen garanti for at de stiger med 1 for hver record (de kan hoppe med 1-3 værdipoint for hver record, afhængig af hvilken node i clusteret der danner recorden, men de er altid stigende).

2.2 HTTP GET /api/auditlog/read?offset=xxx

Dette API endpoint henter op til 250 auditlog records (hvis der er færre end 250 records, hentes kun det antal der er tilgængelige).

Som argument anvendes offset, der er ID'et på den seneste auditlog record man har udlæst. Hvis man fx har udlæst alle auditlog records til og med den med ID 7192, så kan man kalde med

GET /api/auditlog/read?offset=7192

Og så henter den de næste 100 auditlog records der har et ID større end 7192 (fortløbende). Man kan kalde endpointet fortløbende med stigende offsets for at få læst alle nye auditlog records ud.

Da der kan være mange millioner auditlog records, anbefales det at man ikke starter fra 0 og udlæser alle data hver gang, men at man holder en lokal kopi af auditlog records fx i en database.

Eksempel output


```
[
  {
    "id": 3649,
    "timestamp": "2022-03-17T08:40:37.000+00:00",
    "ipAddress": "34.255.4.85",
    "username": "system",
    "entityType": "USER",
    "entityId": "7e088bf3-1cf8-4fe2-a0b5-e3f278d7cb99",
    "entityName": "Brian St Graversen (bsg)",
    "eventType": "LOGIN_EXTERNAL",
    "secondaryEntityType": "ITSYSTEM",
    "secondaryEntityId": "375",
    "secondaryEntityName": "SOFD Core",
    "description": null
  }
]
```


Output er altid er array af auditlog records, og de enkelte felter er beskrevet nedenfor

Felt	Type	Beskrivelse
id	Integer	Er det unikke ID på denne auditlog record, og kan fx anvendes til at identificere næste offset i senere udlæsninger
timestamp	Timestamp (som streng)	Timestamp på hvornår hændelsen indtraf
ipAddress	Streng	IP adressen på den der udførte handlingen
entityType	Streng	Den primære entitet som handlingen er udført på – se nedenfor for en liste over mulige entitets-typer
entityId	Integer	Det unikke (interne) ID på den entiteten
entityName	Streng	Navnet på entiteten
eventType	Streng	Typen af handling der er udført. En løbende voksende liste af handlinger (undlad at hardkode en liste i fortolkningen, da der kan komme yderligere til over tid)
secondaryEntityType	Integer	Typen på et evt sekundært objekt som er involveret (fx den rolle som er tildelt en bruger, hvor brugerne så er den primære entitet)
secondaryEntityId	Streng	Unikt ID på den sekundære entitet
secondaryEntityName	Streng (enum)	Navnet på den sekundære entitet
Description	Streng	En ekstra beskrivelse, hvis en er tilgængelig for denne auditlog record

Mulige entitetstyper

- ORGUNIT (Enhed)
- POSITION (Stilling, gammel kode)

- ROLEGROUP (Rollebuket)
- USER (Bruger)
- TITLE (Stilling, ny kode)
- USERROLE (Jobfunktionsrolle)
- ITSYSTEM (IT-system)
- SYSTEMROLE (Brugersystemrolle)
- KLE_PERFORMING (KLE opgaveansvar)
- KLE_INTEREST (KLE indsigt)
- REQUEST_APPROVE (Anmod/godkend flow)