ID 및 액세스

Astra Automation

NetApp
December 01, 2023

목차

ID 및 액세스

사용자를 나열합니다

특정 Astra 계정에 대해 정의된 사용자를 나열할 수 있습니다.

사용자를 나열합니다

다음과 같은 REST API 호출을 수행한다.

HTTP 메소드	경로
가져오기	/accounts/{account_id}/core/v1/users

추가 입력 매개변수

모든 REST API 호출에서 일반적으로 사용되는 매개 변수 외에도 이 단계의 curl 예제에도 다음 매개 변수가 사용됩니다.

매개 변수	유형	필수 요소입니다	설명
포함	쿼리	아니요	필요에 따라 응답에서 반환할 값을 선택합니다.

curl 예: 모든 사용자의 모든 데이터를 반환합니다

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/users' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

curl 예: 모든 사용자의 이름, 성 및 **ID**를 반환합니다

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/users?include=first
Name,lastName,id' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

JSON 출력 예

```
{
  "items": [
    [
    "David",
    "Anderson",
    "844ec6234-11e0-49ea-8434-a992a6270ec1"
    ],
    [
      "Jane",
      "Cohen",
      "2a3e227c-fda7-4145-a86c-ed9aa0183a6c"
    ]
  ],
  "metadata": {}
}
```

사용자를 생성합니다

특정 자격 증명과 사전 정의된 역할을 가진 사용자를 생성할 수 있습니다. 선택적으로 사용자의 특정 네임스페이스에 대한 액세스를 제한할 수도 있습니다.

사용자 이름을 선택합니다

워크플로우를 수행합니다 "사용자를 나열합니다" 를 클릭하고 현재 사용 중이 아닌 사용 가능한 이름을 선택합니다.

2.사용자를 생성합니다

다음 REST API 호출을 수행하여 사용자를 생성합니다. 통화가 성공적으로 완료된 후 새 사용자는 아직 사용할 수 없습니다.

HTTP 메소드	경로
게시	/accounts/{account_id}/core/v1/users

JSON 입력 예

```
{
  "type" :  "application/astra-user",
  "version" : "1.1",
  "firstName" : "John",
  "lastName" : "West",
  "email" : "jwest@example.com"
}
```

컬의 예

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/users' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data
@JSONinput
```

JSON 출력 예

```
{
    "metadata": {
        "creationTimestamp": "2022-11-20T17:23:15Z",
        "modificationTimestamp": "2022-11-20T17:23:15Z",
        "createdBy": "a20e91f3-2c49-443b-b240-615d940ec5f3",
        "labels": []
    },
    "type": "application/astra-user",
    "version": "1.2",
    "id": "d07dac0a-a328-4840-a216-12de16bbd484",
    "authProvider": "local",
    "authID": "jwest@example.com",
    "firstName": "John",
    "lastName": "West",
    "companyName": "",
    "email": "jwest@example.com",
    "postalAddress": {
        "addressCountry": "",
        "addressLocality": "",
        "addressRegion": "",
        "streetAddress1": "",
        "streetAddress2": "",
        "postalCode": ""
    },
    "state": "active",
    "sendWelcomeEmail": "false",
    "isEnabled": "true",
    "isInviteAccepted": "true",
    "enableTimestamp": "2022-11-20T17:23:15Z",
    "lastActTimestamp": ""
}
```

3.허용되는 네임스페이스를 선택적으로 선택합니다

워크플로우를 수행합니다 "네임스페이스를 나열합니다" 액세스를 제한할 네임스페이스를 선택합니다.

역할에 사용자를 바인딩합니다

사용자를 역할에 바인딩하려면 다음 REST API 호출을 수행합니다. 아래 예제에서는 네임스페이스 액세스에 제한이 없습니다. 을 참조하십시오 "네임스페이스 세분화를 통해 RBAC 강화" 를 참조하십시오.

HTTP 메소드	경로
게시	/accounts/{account_id}/core/v1/roleBindings

JSON 입력 예

```
{
  "type" :  "application/astra-roleBinding",
  "version" : "1.1",
  "userID" : "d07dac0a-a328-4840-a216-12de16bbd484",
  "accountID" : "29e1f39f-2bf4-44ba-a191-5b84ef414c95",
  "role" : "viewer",
  "roleConstraints": [ "*" ]
}
```

컬의 예

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/roleBindings'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data
@JSONinput
```

자격 증명을 생성합니다

다음 REST API 호출을 수행하여 자격 증명을 생성하여 사용자와 연결합니다. 이 예제에서는 base64 값으로 제공된 암호를 사용합니다. 를 클릭합니다 `name` 속성에는 이전 단계에서 반환된 사용자의 ID가 포함되어야 합니다. 입력 속성 `change` 또한 base64로 인코딩되어야 하며 사용자가 처음 로그인할 때 암호를 변경해야 하는지 여부를 결정합니다 (`true` 또는 `false`)를 클릭합니다.

> 이 단계는 로컬 인증을 사용하여 Astra Control Center를 구축하는 경우에만 필요합니다. LDAP 또는 Astra Control Service를 구축한 Astra Control Center 구축 환경에서는 필요하지 않습니다.

HTTP 메소드	경로
게시	/accounts/{account_id}/core/v1/credentials

JSON 입력 예

```
{
  "type" :  "application/astra-credential",
  "version" : "1.1",
  "name" : "d07dac0a-a328-4840-a216-12de16bbd484",
  "keyType" : "passwordHash",
  "keyStore" : {
      "cleartext" : "TmV0QXBwMTIz",
      "change" : "ZmFsc2U="
  },
  "valid" : "true"
}
```

컬의 예

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/credentials'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data
@JSONinput
```