



ID 및 액세스

Astra Automation

NetApp
December 01, 2023

목차

ID 및 액세스	1
사용자를 나열합니다	1
사용자를 생성합니다	2

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 ID를 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
```

저작권 정보

Copyright © 2023 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.