



# Finding and using undocumented AWS APIs

———— Maurice Borgmeier ————

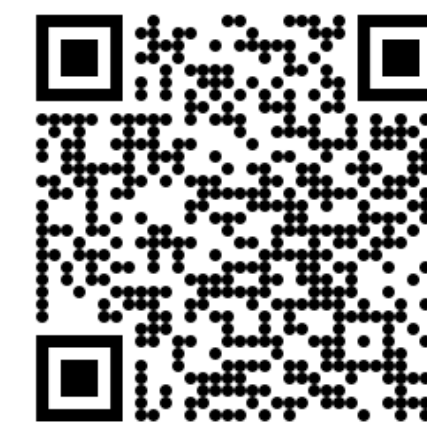
# Who am I

**Maurice Borgmeier**  
(Mopuc)

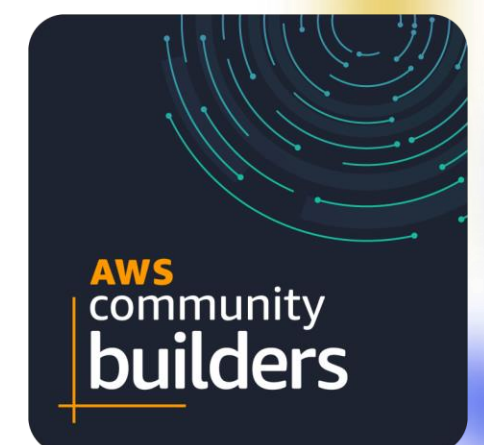
*Senior Cloud Consultant & Trainer @ tecRacer*

- Serverless & Data Analytics
- Collector of AWS Certifications
- AWS Ambassador
- AWS Community Builder

**tecRACER**  
*Cloud Enabling Your Business*



LinkedIn



What can be done in AWS

What the **SDKs** can do

What the **AWS Console** can do

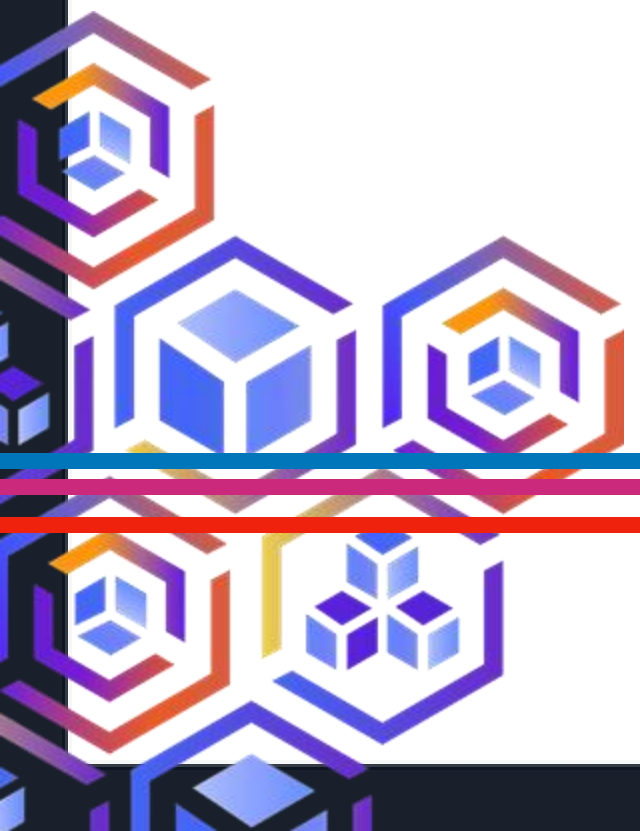


can be done in AWS

the **SDKs** can do

**AWS Console** can do

# Undocumented APIs

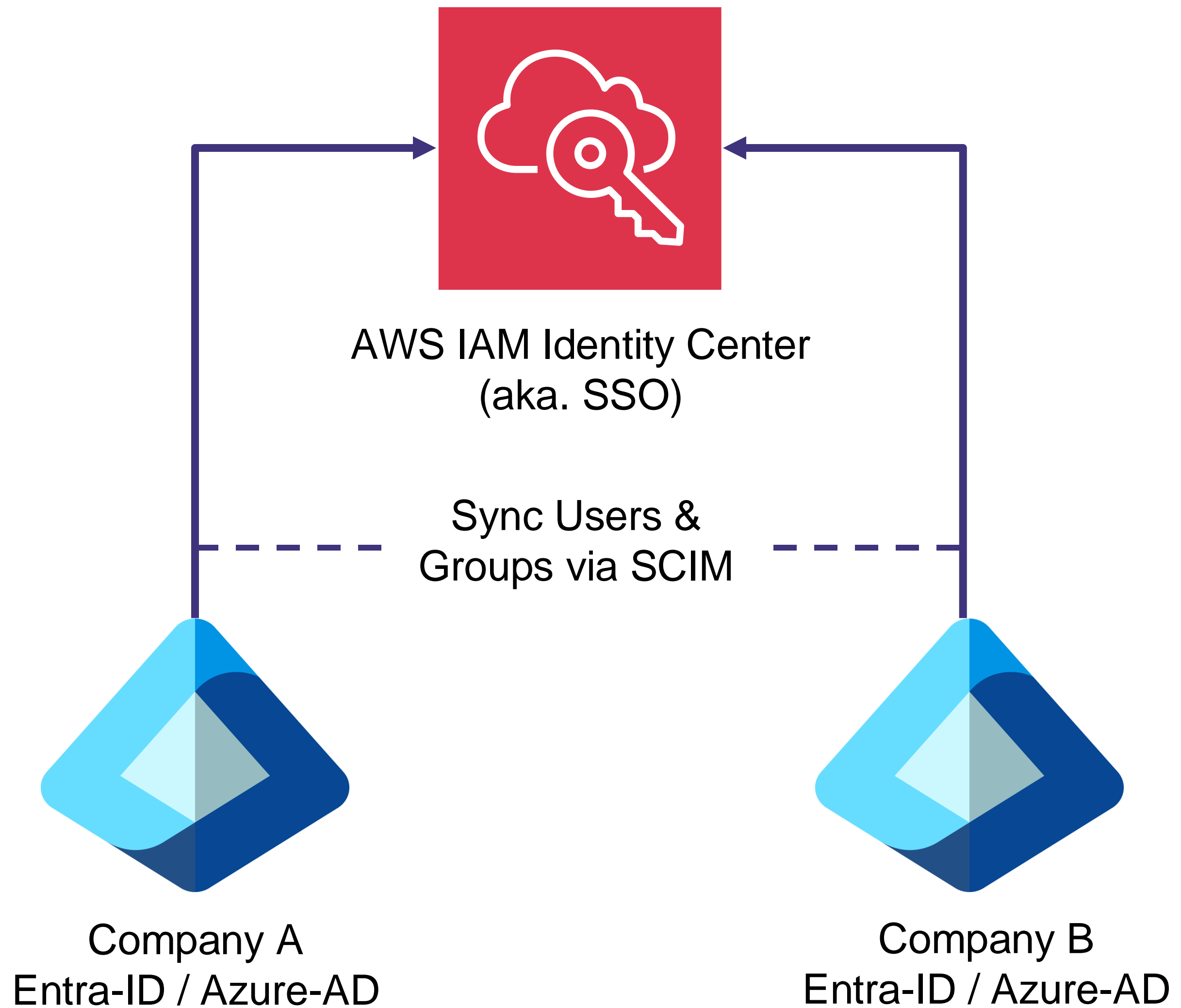


# Identity Provider Migration



AWS IAM Identity Center  
(aka. SSO)

# Identity Provider Migration



# No SDK Support :(

[IAM Identity Center](#) > [Settings](#) > Automatic provisioning


## Automatic provisioning

Automatic provisioning relies on the protocol System for Cross-domain Identity Management (SCIM) to provision users from your identity provider (IdP), and it requires that your IdP support a compatible SCIM profile. [Learn more](#) 

### Configuration

Disable

SCIM endpoint

 <https://scim.eu-central-1.amazonaws.com/kE370acb0d7-97b3-4383-8d96-example/scim/v2/>

Status

 Enabled

The Identity Center directory includes an OAuth bearer access token. This token is issued from the directory with each request. For security reasons, existing access tokens can't be displayed again, but you can generate new ones. The Identity Center directory supports up to two access tokens.

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms
200	POST	eu-central-1.console.aws.am...	presign	awsc-head.js:2 (xhr)	json	2,80 kB	2,06 kB	179
200	POST	eu-central-1.prod.pr.analytic...	panoramamaroute	c0631c552b404bec88a3db78f...	json	667 B	76 B	150
200	GET	global.console.aws.amazon....	metadata?packageName=@amzn/aws-unified-search&majorVersion=1&accountId=7673	fetch	json	1,12 kB	518 B	89
200	GET	global.console.aws.amazon....	metadata?packageName=@amzn/awsc-tools-experience-module&majorVersion=1&acc	fetch	json	1,13 kB	531 B	93
200	POST	eu-central-1.prod.pr.analytic...	panoramamaroute	c0631c552b404bec88a3db78f...	json	667 B	76 B	179
200	POST	up.sso.eu-central-1.amazon...	/	main.js:2 (xhr)	x-amz-json-1.1	399 B	86 B	46
200	POST	eu-central-1.console.aws.am...	evaluate	apertureWidget.js:2 (fetch)	json	1,39 kB	880 B	156
200	POST	eu-central-1.console.aws.am...	presign	index.js:78 (xhr)	json	2,59 kB	1,85 kB	156
200	POST	telemetry.cell-0.eu-central-1...	telemetry	home:16 (fetch)	json	483 B	2 B	107
200	POST	eu-central-1.console.aws.am...	/p/pref/1/	778.js:2 (fetch)	x-amz-json-1.1	526 B	37 B	102
200	POST	eu-central-1.console.aws.am...	/p/log/1/singlesignon/1/OP/	awsc-head.js:2 (xhr)	gif	361 B	43 B	88
200	POST	eu-central-1.prod.pr.analytic...	panoramamaroute	c0631c552b404bec88a3db78f...	json	667 B	76 B	215
200	POST	eu-central-1.console.aws.am...	evaluate	utils.js:22 (fetch)	json	2,09 kB	1,58 kB	159
200	POST	eu-central-1.prod.pr.analytic...	panoramamaroute	c0631c552b404bec88a3db78f...	json	667 B	76 B	216
200	POST	eu-central-1.console.aws.am...	evaluate	utils.js:22 (fetch)	json	895 B	387 B	163
200	GET	global.ccs.amazonaws.com	DiscoverEndpoint	module-metadata.js:4 (fetch)	json	cached	74 B	0 m
200	GET	global.ccs.amazonaws.com	DiscoverEndpoint	module-metadata.js:4 (fetch)	json	cached	74 B	0 m



Inspector Console Debugger **Network** Style Editor Performance Memory Storage Accessibility Application 11

domain:sso

All HTML CSS JS **XHR** Fonts Images Media WS Other  Disable Cache No Throttling

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	358 B	45 B	23
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	564 B	250 B	284
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	397 B	84 B	319
200	POST	sso.eu-central-1.amazonaws...	/	main.js:2 (xhr)	x-amz-json-1.1	706 B	392 B	265
200	POST	sso.eu-central-1.amazonaws...	/	main.js:2 (xhr)	x-amz-json-1.1	706 B	392 B	117
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	345 B	32 B	113
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	564 B	250 B	136
200	POST	up.sso.eu-central-1.amazon...	/	main.js:2 (xhr)	x-amz-json-1.1	399 B	86 B	46
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	682 B	368 B	97
200	POST	up.sso.eu-central-1.amazon...	/identitystore/	main.js:2 (xhr)	x-amz-json-1.1	532 B	218 B	74
200	POST	up.sso.eu-central-1.amazon...	/identitystore/	main.js:2 (xhr)	x-amz-json-1.1	449 B	135 B	85

# We need

Endpoint + Method

API Operation

Request Headers

Parameters

Output Format

Performance Memory Storage Accessibility Application

All HTML CSS JS XHR Fonts Images Media WS Other

Type	Transferred	Size	Headers Cookies Request Response Timings
r) x-amz...	358 B	45 B	Filter properties
r) x-amz...	564 B	250 B	JSON
r) x-amz...	397 B	84 B	ProvisioningTenants: [ {...} ]
r) x-amz...	706 B	392 B	0: Object { CreationTime: 1713363979.436, ScimEndpoint: "https://s0acb0d7-97b3-4383-8d96[redacted].scim/v2/", TenantId: "kE370acb0d7-97b3-4383-8d96[redacted]"; CreationTime: 1713363979.436 ScimEndpoint: "https://scim.eu-central-1.amazonaws.com/kE37031/scim/v2/" TenantId: "kE370acb0d7-97b3-4383-8d96[redacted]"; }
r) x-amz...	706 B	392 B	
r) x-amz...	345 B	32 B	
r) x-amz...	564 B	250 B	
r) x-amz...	399 B	86 B	
r) x-amz...	682 B	368 B	
r) x-amz...	532 B	218 B	
r) x-amz...	449 B	135 B	

Endpoint + Method

API Operation

Request Headers

Parameters

Output Format

Headers Cookies Request Response Timings Stack Trace Security

Filter Headers Block Resend

▶ **POST** https://up.sso.eu-central-1.amazonaws.com/identitystore/ **HTTP Method and URL**

Status **200** ⓘ  
Version HTTP/2  
Transferred 532 B (218 B size)  
Referrer Policy strict-origin-when-cross-origin  
DNS Resolution System

▶ Response Headers (314 B) Raw

▼ Request Headers (2,139 kB) Raw

ⓘ Accept: \*/\*  
ⓘ Accept-Encoding: gzip, deflate, br  
ⓘ Accept-Language: en-US,en;q=0.5  
ⓘ Authorization: AWS4-HMAC-SHA256 Credential=ASIA3FLD[REDACTED]/202404[REDACTED]/eu-central-1/identitystore/ aws4\_request, SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-amz-security-token;x-amz-target;x-amz-user-agent, Signature=3bb59325cdd711551de1bfc881d4c39[REDACTED]19450e2ad8fd6 **Region + Signature Namespace**

ⓘ Connection: keep-alive  
ⓘ Content-Length: 34  
ⓘ **Content-Type: application/x-amz-json-1.1** **Content Type**  
ⓘ DNT: 1  
ⓘ Host: up.sso.eu-central-1.amazonaws.com  
ⓘ Origin: https://eu-central-1.console.aws.amazon.com  
ⓘ Sec-Fetch-Dest: empty  
ⓘ Sec-Fetch-Mode: cors  
ⓘ Sec-Fetch-Site: cross-site  
ⓘ Sec-GPC: 1  
ⓘ TE: trailers  
ⓘ User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:124.0) Gecko/20100101 Firefox/124.0  
X-Amz-Content-Sha256: 37861720bd60f4053de284a101d803a8699251be84d95fb635543a2750cbc5c6

- ✓ Endpoint + Method
- ✓ API Operation
- ✓ Request Headers
- Parameters
- ✓ Output Format

tIA/qb1Z/HMDMTnAUAgZ15qKagF0iegLQpb1ty9qM3pj/YD/mY1IHBYouzy5vZgPBTRUGXSno2qgiqXZZXb2K  
EobRbHCJ/vMERDnuH8aF5Egztscw7/Kg1zlpWtq3Vzx4462meEFi7jI6CHtFOQYpMEShfXSZnnBWw==  
**X-Amz-Target: AWSIdentityStoreService.ListProvisioningTenants**  
X-Amz-User-Agent: aws-sdk-js/2.1467.0 promise

II HTML CSS JS XHR Fonts Images Media WS

▶ Headers Cookies Request Response

⌵ Filter Request Parameters

Request payload

1	<pre>{"IdentityStoreId": "d-996[REDACTED]"}</pre>
---	---------------------------------------------------

- ✓ Endpoint + Method
- ✓ API Operation
- ✓ Request Headers
- ✓ Parameters
- ✓ Output Format

via: sso-admin>ListInstances

# Putting it together

```
import json

import boto3
import requests

from requests_aws4auth import AWS4Auth

# Where we plan to get the info from
region = "eu-central-1"
identity_store_id = "d-99677XXXXX"

# Customize this to select the right credentials
boto_session = boto3.Session()
credentials = boto_session.get_credentials()

# Provide SignatureV4 request signing
auth = AWS4Auth(
    region=region,
    service="identitystore",
    refreshable_credentials=credentials
)

# Make the API call based on the collected information
response = requests.post(
    f"https://up.sso.{region}.amazonaws.com/identitystore/",
    headers={
        "Content-Type": "application/x-amz-json-1.1",
        "X-Amz-Target": "AWSIdentityStoreService.ListProvisioningTenants",
    },
    auth=auth,
    json={"IdentityStoreId": identity_store_id},
)

# Output the response
print(json.dumps(response.json(), indent=2))
```

# Putting it together

Using requests-aws4auth lib to properly sign our API call with credentials.

```
Request Headers (2,139 kB) Raw  
? Accept: /*/  
? Accept-Encoding: gzip, deflate, br  
? Accept-Language: en-US,en;q=0.5  
? Authorization: AWS4-HMAC-SHA256 Credential=ASIA3FLD[REDACTED]/20240418/eu-central-1/identitystore/aws4_request, SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-amz-security-token;x-amz-target;x-amz-user-agent, Signature=3bb59325cdd711551de1bfc881d4c398[REDACTED]9450e2ad8fd6
```

Region + Signature Namespace

```
import boto3  
import requests  
  
from requests_aws4auth import AWS4Auth  
  
# Where we plan to get the info from  
region = "eu-central-1"  
identity_store_id = "d-99677XXXXX"  
  
# Customize this to select the right credentials  
boto_session = boto3.Session()  
credentials = boto_session.get_credentials()  
  
# Provide SignatureV4 request signing  
auth = AWS4Auth(  
    region=region,  
    service="identitystore",  
    refreshable_credentials=credentials  
)
```

# Putting it together

Combining endpoints, headers, and parameters to call the undocumented API

```
# Make the API call based on the collected information
response = requests.post(
    f"https://up.sso.{region}.amazonaws.com/identitystore/",
    headers={
        "Content-Type": "application/x-amz-json-1.1",
        "X-Amz-Target": "AWSIdentityStoreService.ListProvisioningTenants",
    },
    auth=auth,
    json={"IdentityStoreId": identity_store_id},
)

# Output the response
print(json.dumps(response.json(), indent=2))
```



# Putting it together

```
import json

import boto3
import requests

from requests_aws4auth import AWS4Auth

# Where we plan to get the info from
region = "eu-central-1"
identity_store_id = "d-99677XXXXX"

# Customize this to select the right credentials
boto_session = boto3.Session()
credentials = boto_session.get_credentials()

# Provide SignatureV4 request signing
auth = AWS4Auth(
    region=region,
    service="identitystore",
    refreshable_credentials=credentials
)

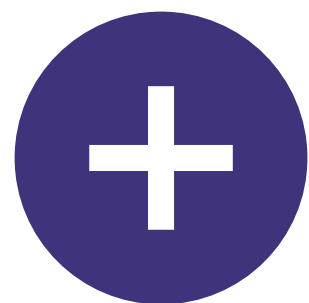
# Make the API call based on the collected information
response = requests.post(
    f"https://up.sso.{region}.amazonaws.com/identitystore/",
    headers={
        "Content-Type": "application/x-amz-json-1.1",
        "X-Amz-Target": "AWSIdentityStoreService.ListProvisioningTenants",
    },
    auth=auth,
    json={"IdentityStoreId": identity_store_id},
)

# Output the response
print(json.dumps(response.json(), indent=2))
```

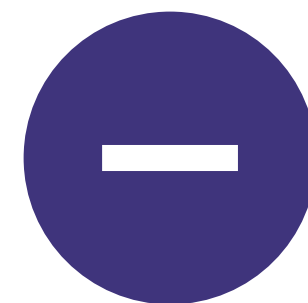
# Output

```
● ● ●  
$ python get_scim_endpoint.py  
{  
  "ProvisioningTenants": [  
    {  
      "CreationTime": 1713363979.436,  
      "ScimEndpoint": "https://scim.eu-central-1.amazonaws.com/kE370acb0d7-97b3-4383-8d96-example/scim/  
v2/",  
      "TenantId": "kE370acb0d7-97b3-4383-8d96-example"  
    }  
  ]  
}
```

# So... is this a good idea?



Automation  
Educational  
(Fun)

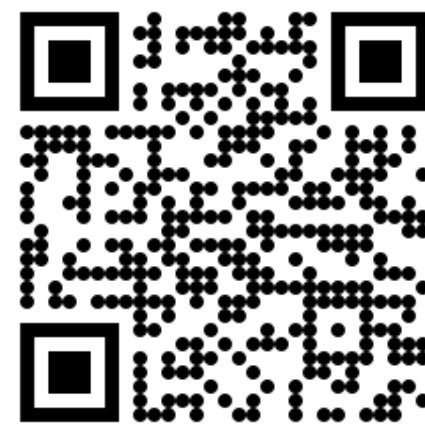


No support  
API may go away at any point  
Data structures may change without notice

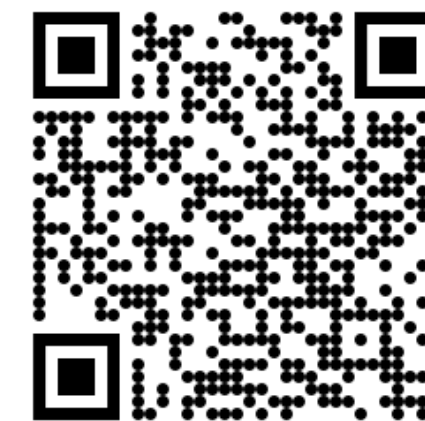
# Thank You

Maurice Borgmeier

Mail: [hi@mauricebrg.com](mailto:hi@mauricebrg.com)



[mauricebrg.com/community-day-bg-2024.html](https://mauricebrg.com/community-day-bg-2024.html)



LinkedIn

<end>