Finding and using undocumented AWS APIs

Maurice Borgmeier

Who am I

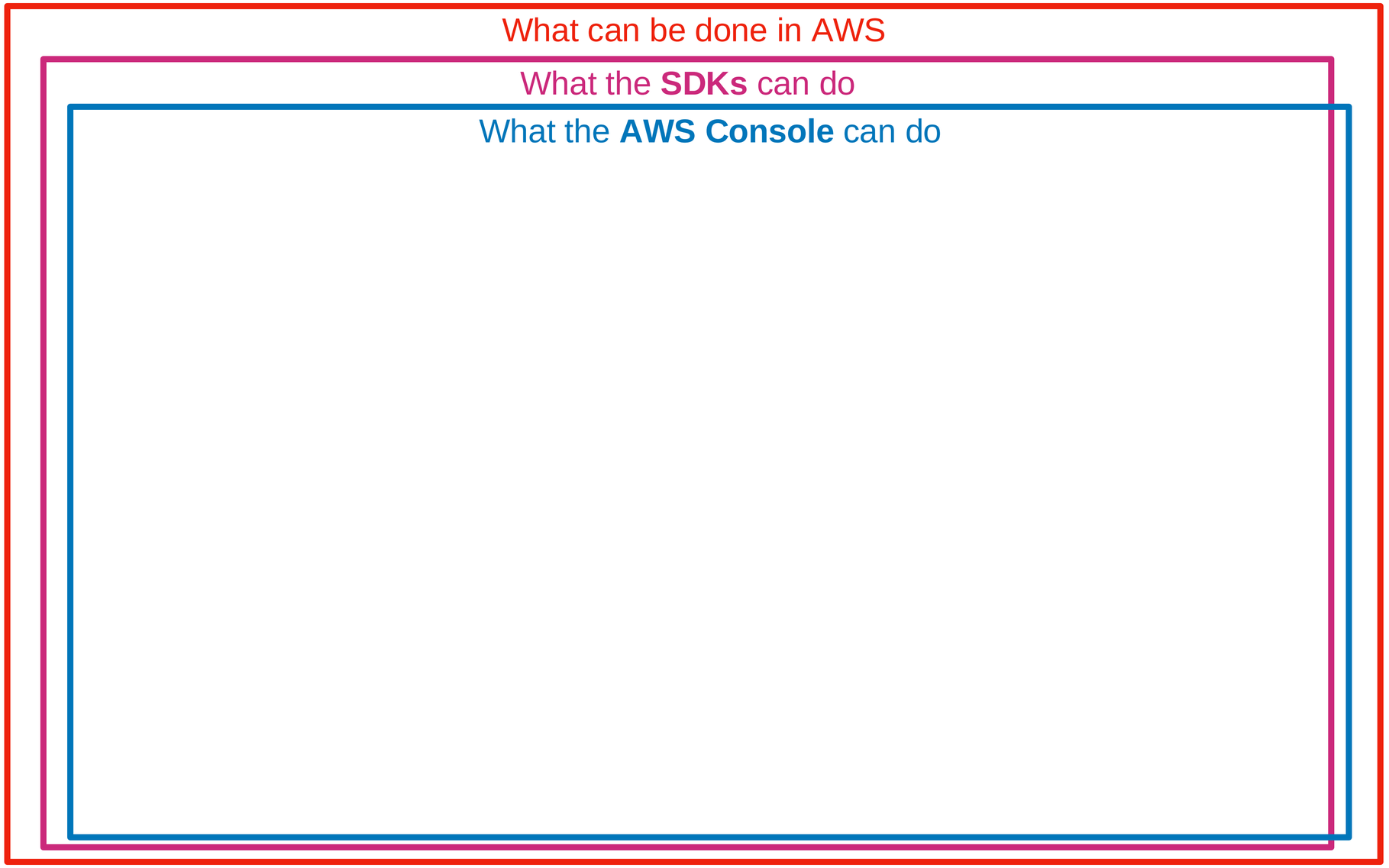
Maurice Borgmeier
(*Mopuc*)

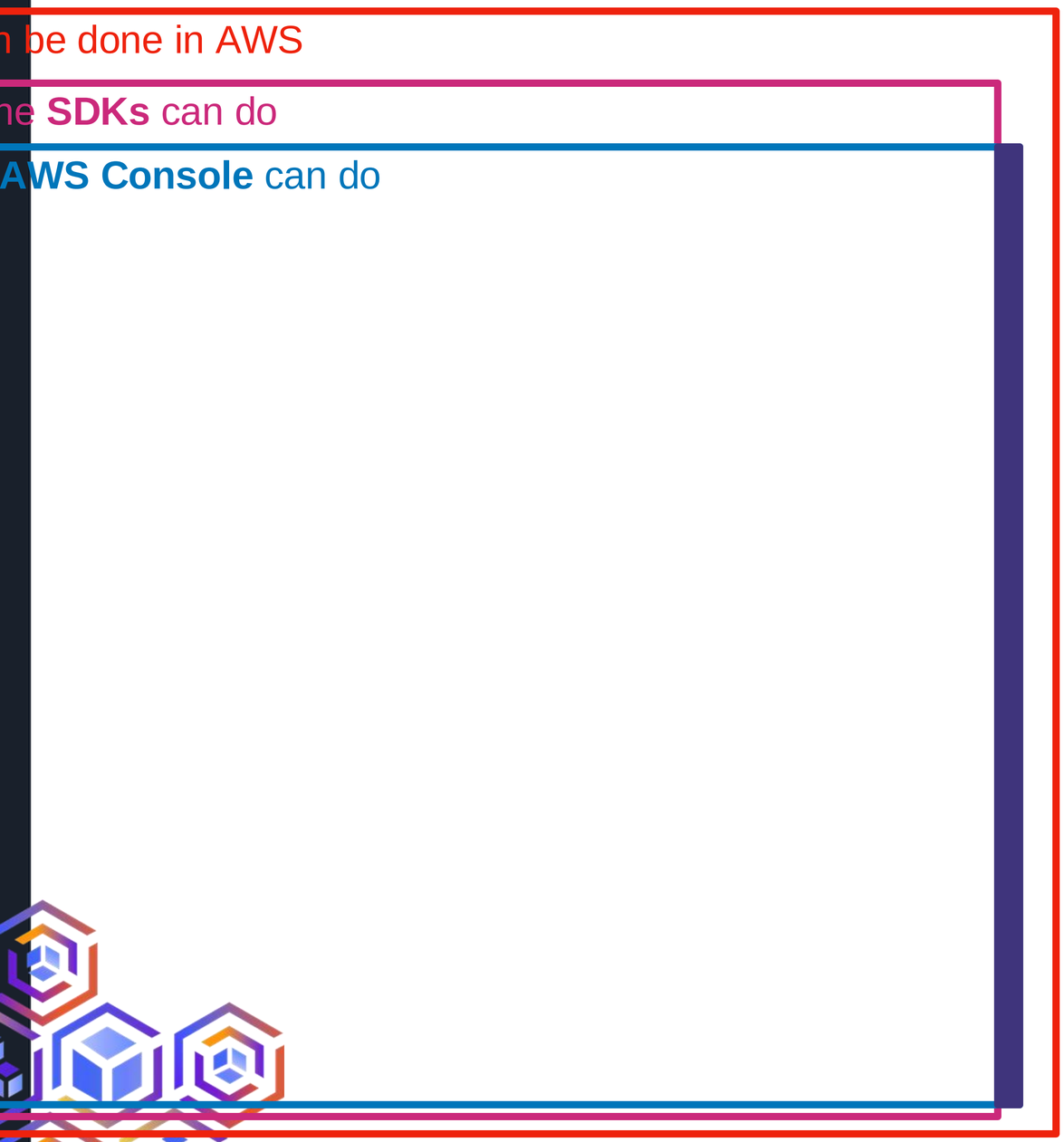
Senior Cloud Consultant & Trainer @ tecRacer

- 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

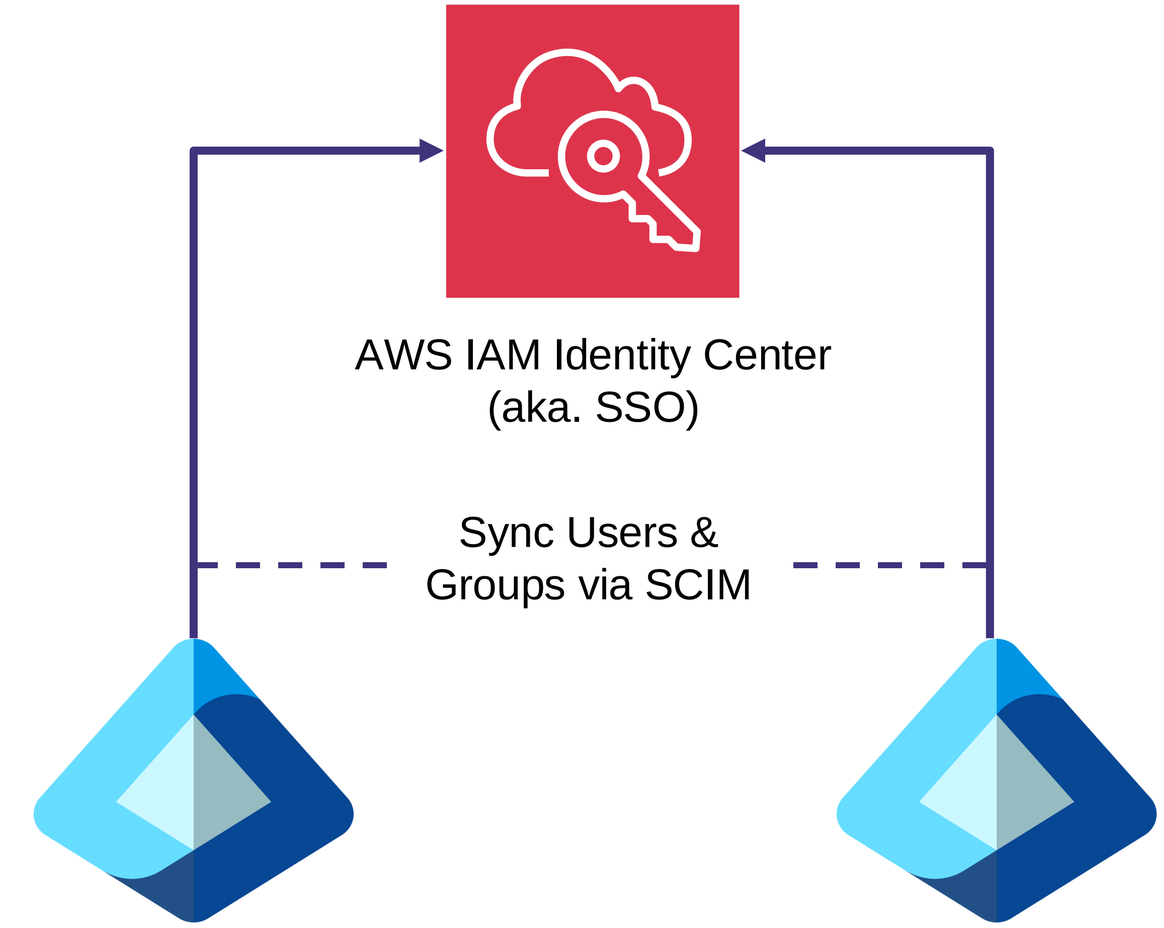


Undocumented APIs

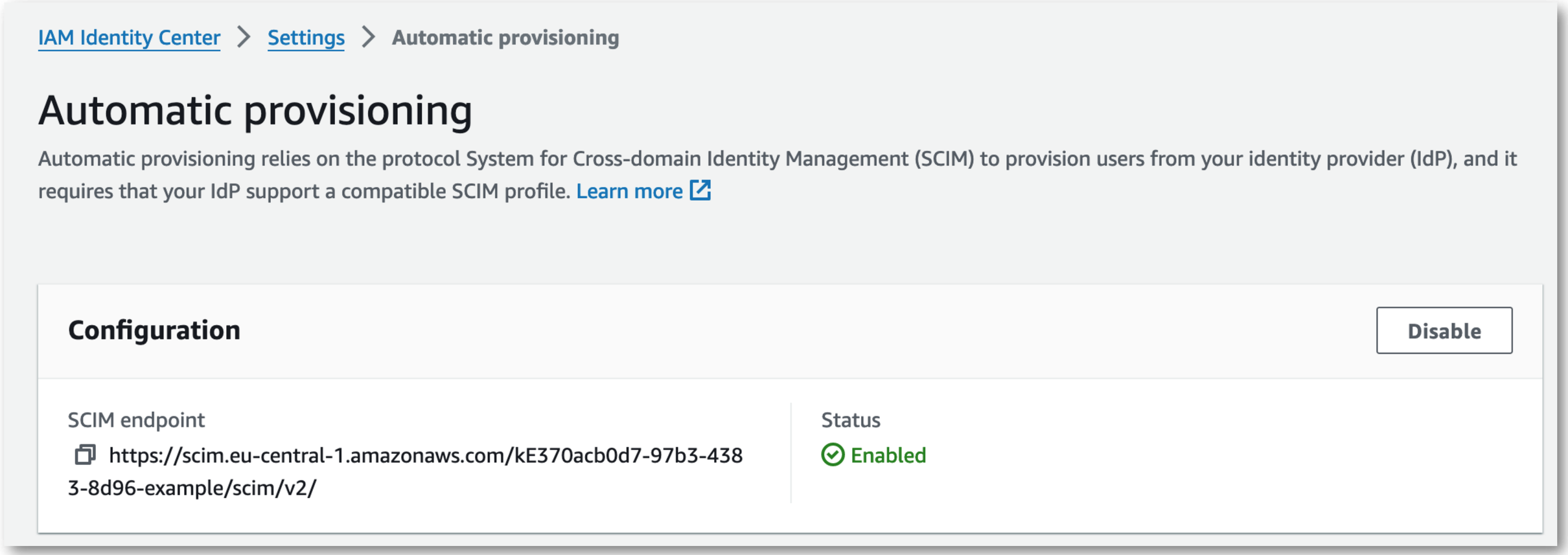
Identity Provider Migration

AWS IAM Identity Center
(aka. SSO)

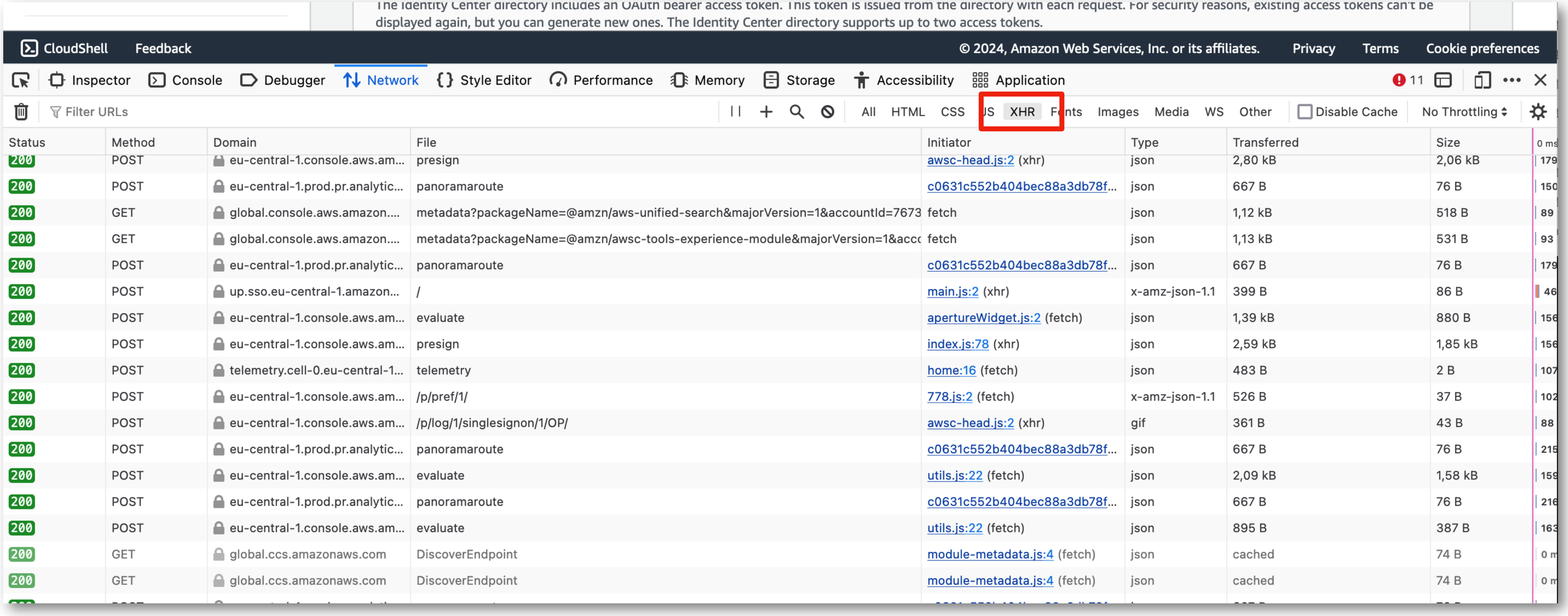
Identity Provider Migration

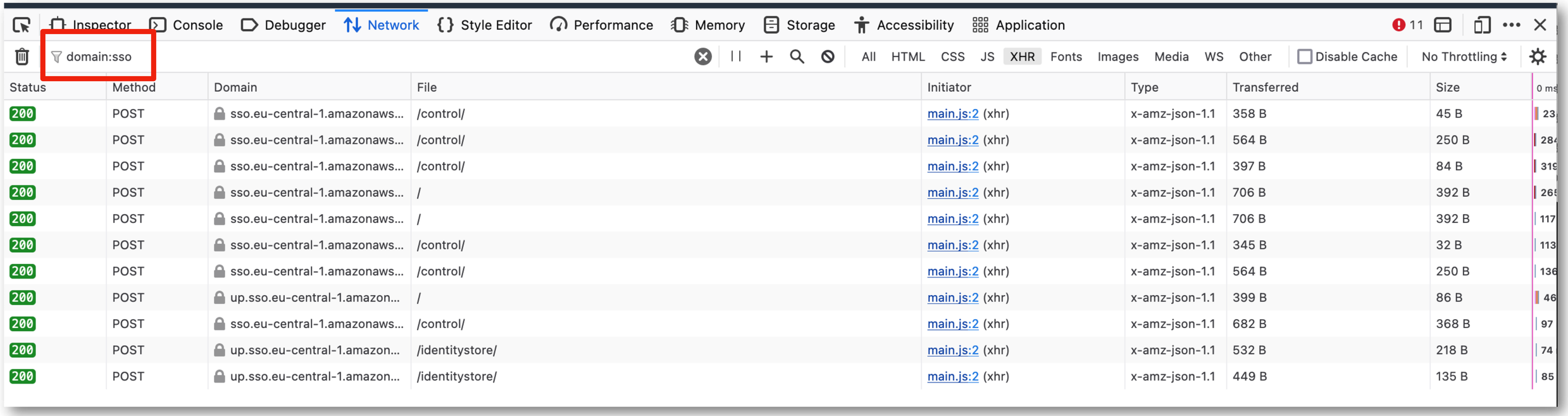


No SDK Support :(



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
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	358 B	45 B
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	564 B	250 B
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	397 B	84 B
200	POST	sso.eu-central-1.amazonaws...	/	main.js:2 (xhr)	x-amz-json-1.1	706 B	392 B
200	POST	sso.eu-central-1.amazonaws...	/	main.js:2 (xhr)	x-amz-json-1.1	706 B	392 B
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	345 B	32 B
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	564 B	250 B
200	POST	up.sso.eu-central-1.amazon...	/	main.js:2 (xhr)	x-amz-json-1.1	399 B	86 B
200	POST	sso.eu-central-1.amazonaws...	/control/	main.js:2 (xhr)	x-amz-json-1.1	682 B	368 B
200	POST	up.sso.eu-central-1.amazon...	/identitystore/	main.js:2 (xhr)	x-amz-json-1.1	532 B	218 B
200	POST	up.sso.eu-central-1.amazon...	/identitystore/	main.js:2 (xhr)	x-amz-json-1.1	449 B	135 B

We need

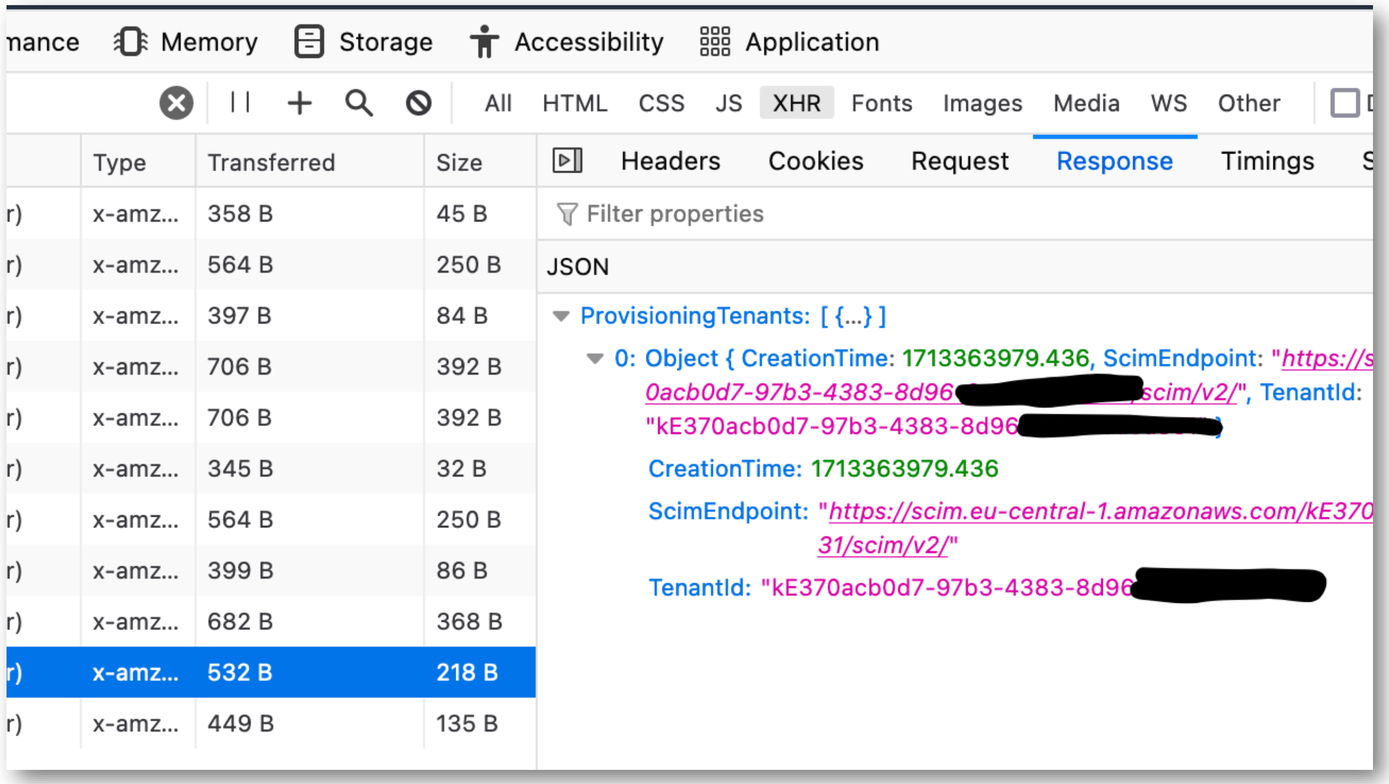
Endpoint + Method

API Operation

Request Headers

Parameters

Output Format



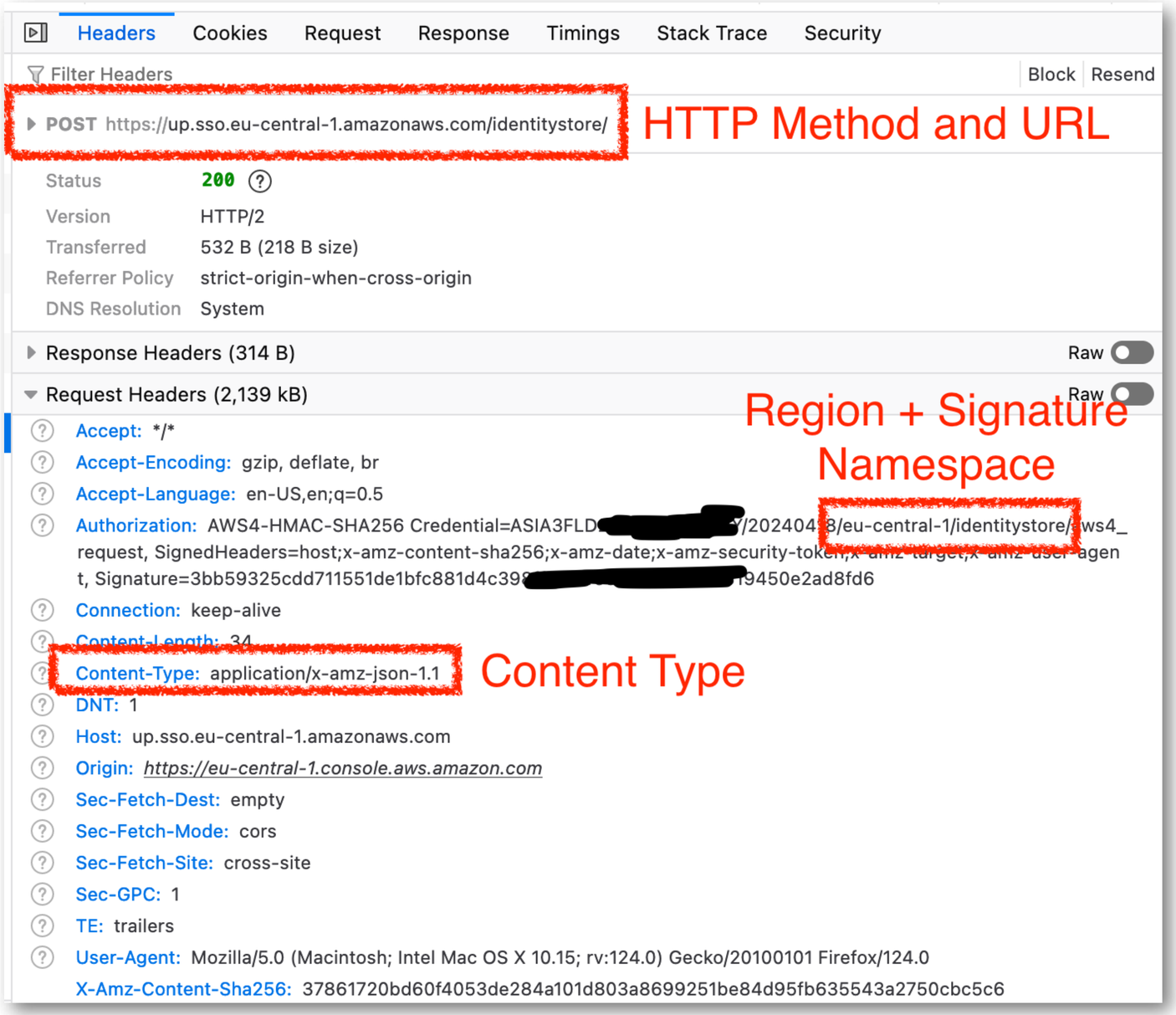
Endpoint + Method

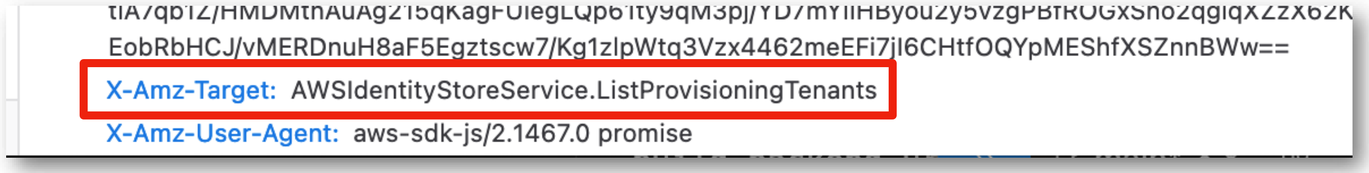
API Operation

Request Headers

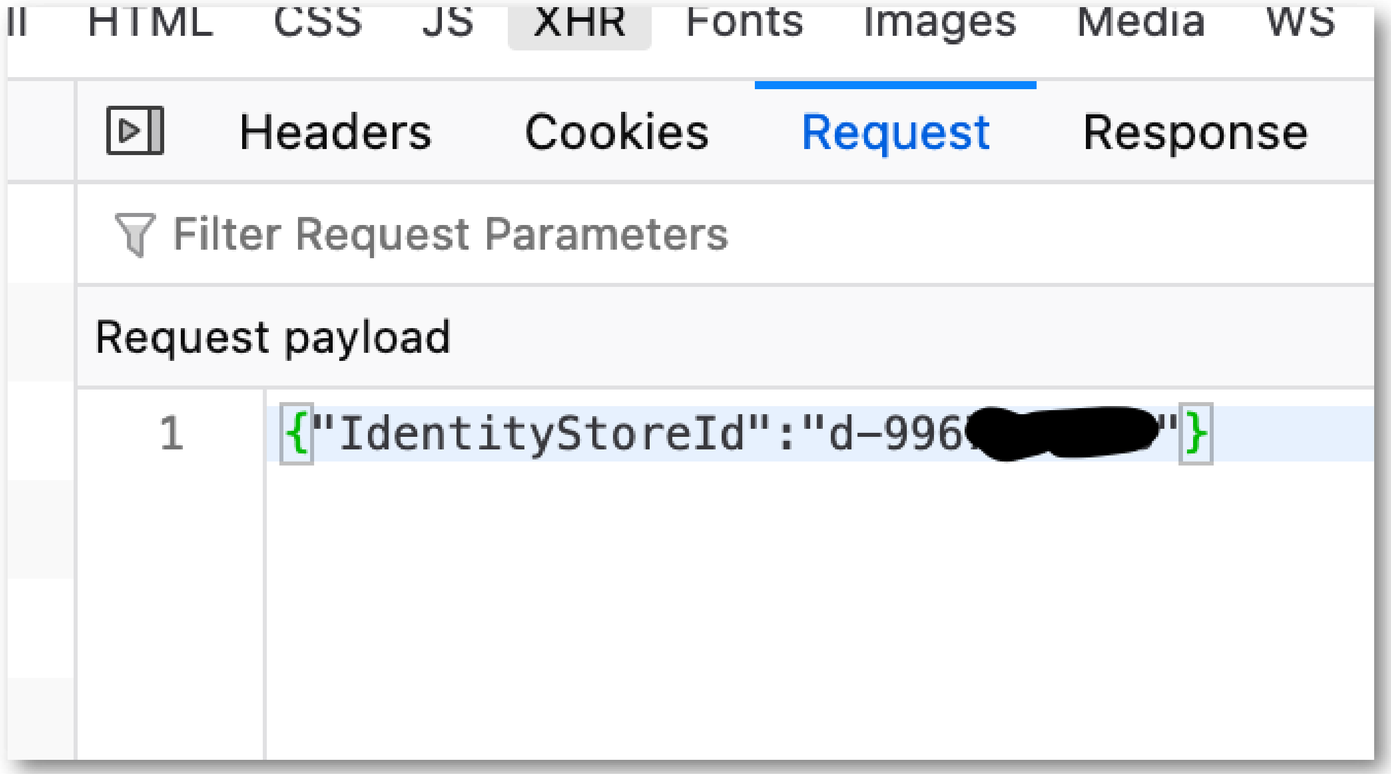
Parameters

☑ Output Format





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



- [x] Endpoint + Method
- [x] API Operation
- [x] Request Headers
- [x] Parameters
- [x] Output Format

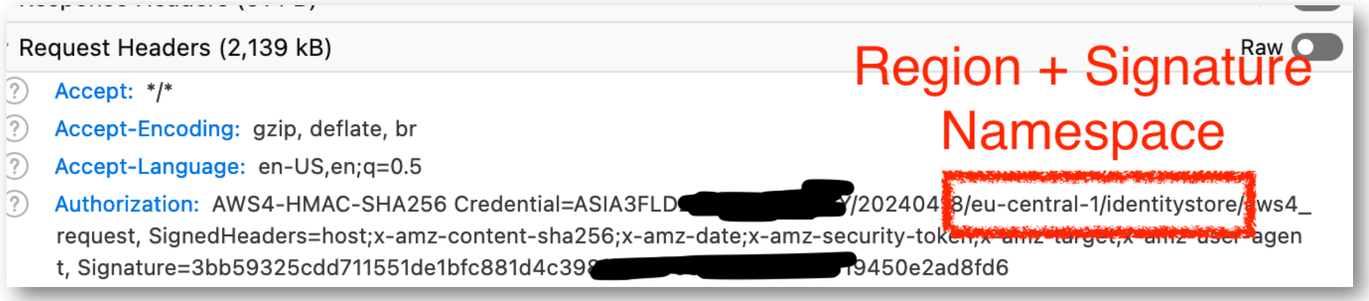
via: `sso-admin:ListInstances`

Putting it together

```python
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.




```python
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

```python
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

mauricebrg.com/community-day-bg-2024.html

LinkedIn

<end>