

Data Product API

Created by [Yerneni, Bhaskar Teja](#), last modified by [MCDONALD, JP P](#) on [Oct 12, 2022](#)

Support

For questions or comments regarding lower environments please reach out directly via email edsplatformsupport@evernorth.com.

For production issues please open a service now ticket.

For more information about the Ingestions Framework see the [supporting documentation](#).

02. Prerequisites

See the [02. Prerequisites](#) page regarding global group access required to use the API's and manage your teams data.

This is also where you will make your **G_TEAM_{NAME}** global group which is referenced often in the following documentation.

03. Accessing the API via Postman

See the [03. Accessing the API via Postman](#) page regarding postman setup and authentication.

- [02. Prerequisites](#)
 - [03. Accessing the API via Postman](#)
- [Create your team's global group](#)
 - [Using your Team's global group](#)
 - [Example on using your Global Group with DataSets](#)
 - [Example on using your Global Group with Data Products](#)
 - [Minting a role for different purposes in the Platform](#)
- [DataProduct object](#)
 - [API:](#)
 - [Create data product:](#)
 - [Fetch data products:](#)
 - [Fetch single data product:](#)

- Add a source:
 - Add a pipeline:
 - Update data product serving:
- Example of making a Data Product
 - Making a data product
 - Getting write access to my data product
 - Creating a serving
 - Example Servings (to be refined)
 - Example Servings Access (to be refined)
 - Making a multi tenant data product
 - Using a Data Product that someone else published
 - Requesting access to a Data Product
 - Building & Serving your own Data Product

Create your team's global group

Teams will need to create a **G_TEAM_{name}** global group for their respective teams as part of their data ownership.

You do this through Aveksa (<https://accessmanagement.sys.cigna.com/aveksa/>)

This global group will be used to give access to registered datasets and data products.

What do it mean to be part of a team?

How do teams work for analytical teams?

Using your Team's global group

Individuals in these global groups will be able to do the following as we continue to build out the platform:

- Create and modify ingestion-datasets for their team
- Start and stop ingestions for ingestion-datasets for their team
- Read ingested data for their team's datasets in lower environments and using a service role for prod (primarily for writing ETL jobs)
- Create and modify data products with locations from which the data products data will reside (tables, views, topics ...)
- Read and write to the locations defined within data products their team owns in lower environments and using a service role for prod
- Read from the locations defined within data products that their team has been granted access in lower environments and using a service role for prod

Example on using your Global Group with DataSets

after you have created a global group, it will be used to register a dataset and create data products.

example:

- GG created = G_TEAM_PROVIDER

Provider team then registers dataset, like the following:

- datasetOwner = PROVIDER

```
{
  "sourceInfo": {
    "sourceType": "Kafka",
    "topicConfigs": [
      {
        "topic": "edpint.iodsint.IBORODS.INDIV_CVRG"
      },
      {
        "topic": "edpint.iodsint.IBORODS.INDIV_CVRG_G"
      },
      {
        "topic": "edpdev.pmrdev.PMRS.MD_INSTITUTIONAL_EXTERNAL"
      }
    ],
    "sourceFormat": "Avro"
  },
  "datasetName": "my_awesome_dataset_name",
  "datasetOwner": "PROVIDER",
  "datasetAcquisition": "STREAM",
  "targets": [
    "sandbox"
  ]
}
```

Example on using your Global Group with Data Products

See here: [Data Product API](#)

Minting a role for different purposes in the Platform

- ALL ROLES ARE READONLY

To support various components in the platform, such as ingestion data, you can 'mint' or create a role that is associated with your team.

Your team becomes your identity in the Platform.

DataProduct object

An example DataProduct object in EDP

Example DataProduct object

```
{
  "dataProductName": "normalized_name_client",
  "sources": {
    "client_account_entity": {
      "sourceType": "DATASET",
      "created": "2021-08-03T14:09:13.923512+00:00",
      "datasetName": "ESI_ACCOUNT_GROUP"
    },
    "client_contract_entity": {
      "sourceType": "DATASET",
      "created": "2021-07-30T20:00:51.261956+00:00",
      "datasetName": "ESI_CLIENT_CONTRACT"
    }
    ...
  },
  "pipelines": [
    {
      "pipelineType": "AIRFLOW"
      "pipelineId": "123" # airflow DAG ID
      "tasks": {
        "taskName": "gwy1_client_raw_to_delta"
        ...
      }
    }
  ]
}
```

```
    },
    ...
  ],
  "serving": {
    "gwy1_client_delta": {
      "servingType": "S3",
      "created": "2021-08-02T13:09:14.765953+00:00",
      "url": "s3://cds-data-product-dev/normalized_name_client/delta/stage/client",
      "schemaUrl": "s3://cds-data-product-dev/normalized_name_client/delta/databricks/delta/stage/gwy1_rz"
    }
    "gwy1_api": {
      "servingType": "api",
      "url": "https://bullseye.da-datastore-prod.aws.cignacloud.com/",
      "schemaId": "cb748763-121b-499a-8d40-829a0afa9c56"
    },
    "gwy1_kafka": {
      "servingType": "kafka",
      "kafkaUrl": "",
      "topic": "bullseye",
      "schemaId": "2a654e51-ddf5-45d1-96d5-2ff9dae77c4a"
    }
    ...
  },
  {
    "access": {
      "CONSUMER": ["gwy1_client_delta"]
      ...
    }
  },
  "owner": "BULLSEYEDEV",
  "slas": ...,
  "created": "2021-08-18T15:12:05.872500+00:00"
}
```

API:

Create data product:

 PUT /v2/internal/data-products/{dataProductName}


Fetch data products:

 GET /v2/internal/data-products/

Fetch single data product:


 GET /v2/internal/data-products/{dataProductName}

Add a source:

 POST /v2/internal/data-products/{dataProductName}/sources

```
{
  "sourceType": "DATASET",
  "sourceName": "source_one",
  "datasetName": "e2e_dataproduct_dataset_one"
}
```

Add a pipeline:

 POST /v2/internal/data-products/{dataProductName}/pipelines

```
{
  "pipelineName": "e2e-test-dag"
  "pipelineType": "AIRFLOW",
  "created": "2021-09-23T14:33:46.126341+00:00",
    "tasks": [
      {
        "taskName": "e2e_metric"
      }
    ],
  "slos": {}
}
```

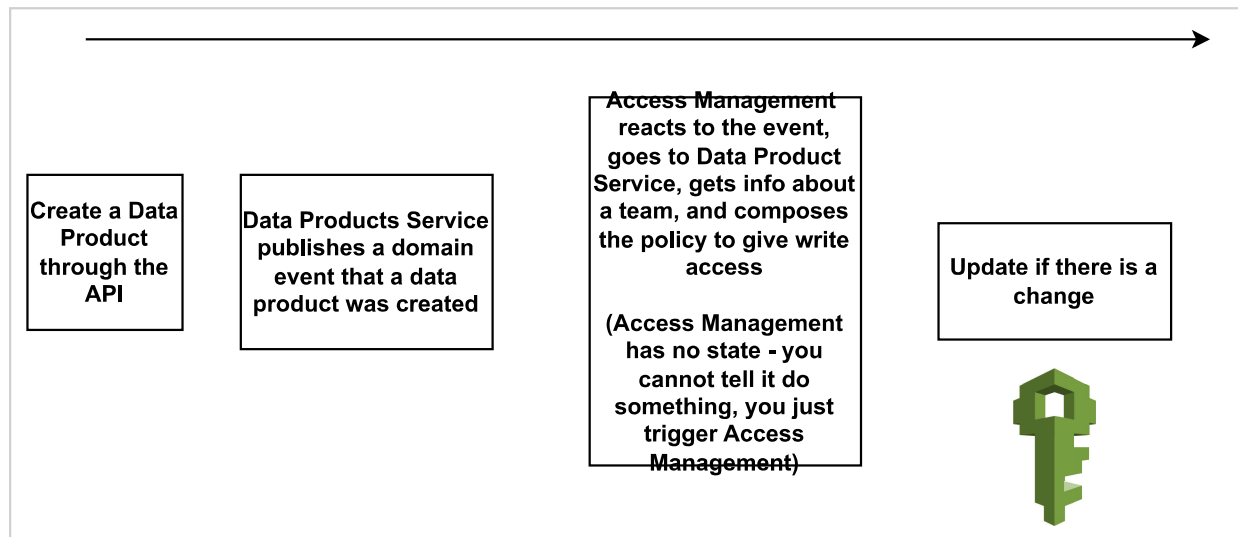
Update data product serving:

 POST /v2/internal/data-products/{dataProductName}/servings

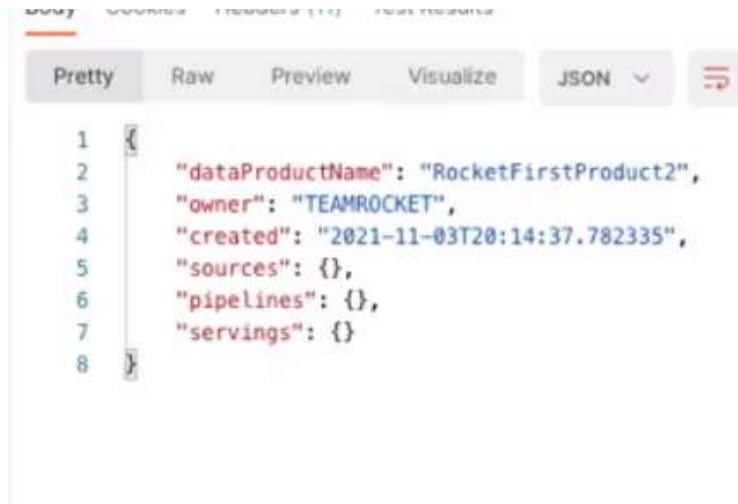
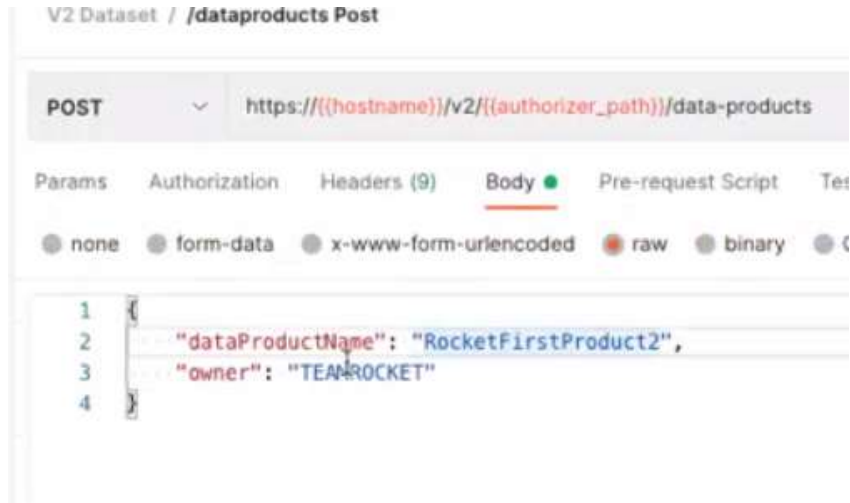
```
"serving": [
  {
    "type": "S3",
    "url": "s3://cds-data-product-dev/supphealthapp/bullseye_out",
    "schemaUrl": "s3://cds-data-product-dev/supphealthapp/bullseye_out_schema"
  },
  {
    "type": "api",
    "url": "https://bullseye.da-datastore-prod.aws.cignacloud.com/",
    "schemaId": "cb748763-121b-499a-8d40-829a0afa9c56"
  },
]
```

```
{  
  "type": "kafka",  
  "kafkaUrl": "",  
  "topic": "bullseye",  
  "schemaId": "2a654e51-ddf5-45d1-96d5-2ff9dae77c4a"  
}  
...  
],
```

Example of making a Data Product



Making a data product



Getting write access to my data product

[Roles](#) > TEAM_TEAMROCKET

Summary

Role ARN	arn:aws:iam::643101592424:role/team/TEAM_TEAMROCKET 
Role description	TEAMROCKET's role for access to datasets and data products Edit
Instance Profile ARNs	
Path	/team/
Creation time	2021-11-01 16:15 PDT
Last activity	2021-11-01 16:35 PDT (3 days ago)
Maximum session duration	1 hour Edit

Give this link to users who can switch roles in the console https://signin.aws.amazon.com/switchrole?roleName=team/TEAM_TEAMROCKET&account=cigna-us-da-datastore-dev 

Permissions

Trust relationships

Tags

Access Advisor


Revoke sessions

▼ Permissions policies (1 policy applied)


[Attach policies](#)

Policy name ▼

► [TEAM_TEAMROCKET_DROPZONE](#)

Role ARN arn:aws:iam::643101592424:role/team/TEAM_TEAMROCKET 

Role description TEAMROCKET's role for access to datasets and data products | [Edit](#)

Instance Profile ARNs 

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": "s3:ListAllMyBuckets",
    "Resource": "arn:aws:s3::*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:AbortMultipartUpload",
      "s3:DeleteObject*",
      "s3:GetObject*",
      "s3:ListBucket",
      "s3:ListMultipartUploadParts",
      "s3:PutObject*"
    ],
    "Resource": [
      "arn:aws:kms:us-east-1:643101592424:key/d7d2b274-b030-4424-b84b-0e21abef7e9d",
      "arn:aws:kms:us-east-1:643101592424:key/d7d2b274-b030-4424-b84b-0e21abef7e9d/TeamRocketFirstProduct/*"
    ],
    "Condition": {
      "StringEqualsIfExists": {
        "s3:x-amz-server-side-encryption-aws-kms-key-id": "arn:aws:s3:::cds-data-product-dev"
      },
      "StringLikeIfExists": {
        "s3:prefix": [
          "TeamRocketFirstProduct/*"
        ]
      }
    }
  }
]
}

```

Creating a serving

VZ dataproducts / /data-products - Create

POST https://api.da-datastore-{{env}}.aws.cignacloud.com/v2/internal/data-products/client_account/servings

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "servingName": "some_serving",
3   "servingType": "S3",
4   "url": "s3://da-datastore-client.dev-cignasplithorizon/databricks/delta/stage/client",
5   "schemaUrl": "s3://da-datastore-client.dev-cignasplithorizon/databricks/delta/stage/gwy1_r2"
6 }

```

Example Servings (to be refined)

As a producer of data products you can write to the full prefix under the /cds-data-product-{env}/{dataProductName}/ location and the following servings are ones that are exposed and can be requested for access.

```
{
  "dataProductName": "incentives",
  "servings": {
    "goals_CIS": [
      {
        "servingType": "S3",
        "address": "s3://cds-data-product-dev/incentives/customer_goals",
        "schemaUrl": "s3://cds-data-product-dev/incentives/databricks/delta/stage/gwy1_rz",
      },
      {
        "servingType": "S3",
        "address": "s3://cds-data-product-dev/incentives/customer_completions",
        "schemaUrl": "s3://cds-data-product-dev/incentives/databricks/delta/stage/gwy1_rz"
      },
      {
        "servingType": "kafka", (not implemented)
        "topic": "goalsAndCompletionsUpdated"
      }
    ],
    "goals_DataAnalysts": [
      {
        "servingType": "snowTable", (not implemented)
        "address": "acctid/incentives.customer-goals",
        "schemaUrl": "s3://da-datastore-client.dev-cignasplithorizon/databricks/delta/stage/gwy1_rz"
      },
      {
        "servingType": "kafka",
        "topic": "goalsAndCompletionsUpdated"
      }
    ]
  },
}
```

```
"access": {  
  "TEAMGALACTIC": ["goals_CIS"]  
}
```

Example Servings Access (to be refined)

The owner of a data product may grant access to other teams and the platform will facilitate the read access if the data product is written to the data product path within the data product bucket like so:

```
{  
  "dataProductName": "incentives",  
  "servings": {  
    "goals_CIS": [  
      {  
        "servingType": "S3",  
        "address": "s3://cds-data-product-dev/incentives/customer_goals",  
        "schemaUrl": "s3://cds-data-product-dev/incentives/databricks/delta/stage/gwy1_rz"  
      }  
    ],  
  },  
  "access": {  
    "TEAMGALACTIC": ["goals_CIS"]  
  }  
}
```

Making a multi tenant data product

If you tag that your data product is multi tenant, a policy such as below will be created. You will need to write object tags to your data product output for consumption as it checks what the user has access to.

Sharing multi tenant data products

Today, the data products api allows for teams to define data products with servings that grants them the ability to read and write data to a designated space within the data products bucket.

But what is data if it is not consumed? It is like the sound of a tree that falls with no one to hear it. Can it really be defined as a sound at that point?

Now, however has come that day in which that sound can be heard, AKA data product servings can be shared and multitenancy applied.

- * Defining a data product
- * Defining for that data product servings with no access restrictions, organizational restrictions and organizational+tenant restrictions.
- * Sharing those servings with another team
- * Demonstrating that the appropriate permissions are being applied to the team granted access

Creating a data product

* POST `/v2/internal/data-products`

```
``` json
{
 "dataProductName": "it",
 "owner": "TEAMROCKET"
}
```
```

Creating a data product serving with organization access restrictions

* POST `/v2/internal/data-products/it/servings`

```
``` json
{
 "servingName": "orglevel",
 "servingType": "S3",
 "url": "s3://cds-data-product-dev/it/orglevel",
 "schemaUrl": "TBD",
 "accessRestriction": "ORGANIZATION"
}
```
```

Creating a data product serving with tenant access restrictions

* POST `/v2/internal/data-products/it/servings`

```
``` json
{
 "servingName": "tenantlevel",
 "servingType": "S3",
 "url": "s3://cds-data-product-dev/it/tenantlevel",
 "schemaUrl": "TBD",
 "accessRestriction": "TENANT"
}
```

```
}
...
```

### Granting access to the data product servings

```
* PUT `/v2/internal/data-products/it/access`
```

```
``` json
{
  "grantee": "INTEGRATION-TESTS",
  "servings": ["orglevel", "tenantlevel"]
}
...`
```

INTEGRATION-TESTS updated policy:

```
``` json
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": "s3:GetObject*",
 "Resource": [
 "arn:aws:s3:::cds-data-product-dev/it/orglevel/*"
],
 "Effect": "Allow",
 "Condition": {
 "StringLike": {
 "aws:PrincipalTag/orgs": "*:${s3:ExistingObjectTag/org}:*"
 }
 }
 },
 {
 "Action": "s3:GetObject*",
 "Resource": [
 "arn:aws:s3:::cds-data-product-dev/it/tenantlevel/*"
],
 "Effect": "Allow",
 "Condition": {
 "StringLike": {
 "aws:PrincipalTag/orgs": "*:${s3:ExistingObjectTag/org}:*",
 "aws:PrincipalTag/tenantIds": "*:${s3:ExistingObjectTag/tenantId}:*"
 }
 }
 },
 {
 "Action": "s3:GetObject*",
```

```
"Resource": [
 "arn:aws:s3:::cds-data-product-dev/it/tenantlevel/*"
],
"Effect": "Allow",
"Condition": {
 "StringLike": {
 "aws:PrincipalTag/orgs": "*:${s3:ExistingObjectTag/org}:*",
 "aws:PrincipalTag/tenantGroupIds": "*:${s3:ExistingObjectTag/tenantGroupId}:*"
 }
},
{
 "Action": "s3:ListBucket",
 "Resource": "arn:aws:s3:::cds-data-product-dev",
 "Effect": "Allow",
 "Condition": {
 "StringLike": {
 "s3:prefix": [
 "it/orglevel*/org=USM/*",
 "it/tenantlevel*/org=USM/*"
]
 },
 "aws:PrincipalTag/orgs": "*:USM:*"
 }
},
{
 "Action": "s3:ListBucket",
 "Resource": "arn:aws:s3:::cds-data-product-dev",
 "Effect": "Allow",
 "Condition": {
 "StringLike": {
 "s3:prefix": [
 "it/orglevel*/org=EHS/*",
 "it/tenantlevel*/org=EHS/*"
]
 },
 "aws:PrincipalTag/orgs": "*:EHS:*"
 }
},
{
 "Action": "kms:Decrypt",
 "Effect": "Allow",
 "Resource": "arn:aws:kms:us-east-1:643101592424:key/d7d2b274-b030-4424-b84b-0e21abef7e9d"
}
]
```



```
}
...
```

## Access Granted to USM

aws s3 cp --quiet [s3://cds-data-product-dev/it/orglevel/org=USM/test.txt](#) /dev/stdout

## Access Not Granted to EHS

aws s3 cp [s3://cds-data-product-dev/it/orglevel/org=EHS/test.txt](#) /dev/stdout

## Access Granted to TenantId: 1

aws s3 cp --quiet [s3://cds-data-product-dev/it/tenantlevel/org=USM/test.txt](#) /dev/stdout

## Access Not Granted to TenantId: 2

aws s3 cp [s3://cds-data-product-dev/it/tenantlevel/org=USM/testfail.txt](#) /dev/stdout

## Access Granted to TenantGroupId: 3

aws s3 cp --quiet [s3://cds-data-product-dev/it/tenantlevel/org=USM/testgroup.txt](#) /dev/stdout

## Access Not Granted to TenantGroupId: 4

aws s3 cp [s3://cds-data-product-dev/it/tenantlevel/org=USM/testfailgroup.txt](#) /dev/stdout

### Example policy

```
{
 "Effect": "Allow",
 "Action": "s3:GetObject*",
 "Resource": ["arn:aws:s3:::cds-data-product-dev/dap_mvp_0/*"],
 "Condition": {
 "StringLike": {
 "aws:PrincipalTag/orgs": "*:${s3:ExistingObjectTag/org}:*",
 "aws:PrincipalTag/tenants": "*:${s3:ExistingObjectTag/tenant}:*"
 }
 }
},
{
 "Effect": "Allow",
 "Action": "s3:GetObject*",
 "Resource": ["arn:aws:s3:::cds-data-product-dev/dap_mvp_0/*"],
 "Condition": {
 "StringLike": {
```

```

 "aws:PrincipalTag/orgs": "*:${s3:ExistingObjectTag/org}:*",
 "s3:ExistingObjectTag/tenant": "unrestricted"
 }
}
}

```

### Example of writing out data to the data product bucket

<https://docs.databricks.com/security/credential-passthrough/iam-passthrough.html#read-and-write-s3-data-using-credential-passthrough>

```

dbutils.fs.mount(
 "s3a://cds-data-product-dev/CatchingPikachu/",
 "/mnt/cds-data-product-dev/CatchingPikachu/",
 extra_configs={"fs.s3a.credentialsType":
 "AssumeRole", "fs.s3a.stsAssumeRole.arn":
 "arn:aws:iam::643101592424:role/team/TEAMROCKET",
 "fs.s3a.acl.default": "BucketOwnerFullControl"
 }
)

my_data_product
 .write.partitionBy('org')
 .format("delta")
 .save("/mnt/cds-data-product-dev/CatchingPikachu/pokedex/")

```

### Example of adding the object tags

```

dbutils.notebook.run("/dap/parametrized_tagging", 300, {"BUCKET": "cds-data-product-dev", "PREFIX": "dap_mvp_0",
"ROLE_ARN": "arn:aws:iam::643101592424:role/team/TEAM-DAP", "SUFFIX": "parquet"})

```

## Tags (4)

Track storage cost of other criteria by tagging your objects. [Learn more](#) 

Key	Value
tenantGroupId	1000000000
org	EHS
tenantId	1000000000
tenantCode	EIGHT16 FOUNDATIONS:CRGV

## Using a Data Product that someone else published

A data product, once discovered, should have a unique address following a global convention that helps its users to programmatically access it.

The servings layer listed in a Data Product definition will help you locate where you can access that Data Product.

An example of a Data Product being served in S3:

```
{
 "dataProductName": "normalized_name_client",
 "organization": "USM|EHS",
 "product": "mdlive",
 "tenant": "cigna",
 "serving": "s3://dataproducs/MyFirstDataProduct/Table1"
}
```

## Requesting access to a Data Product

We want teams to be able to request access to Data Products that they discover.

An API is available for you to grant a team access to a Data Product.

### How do I know they should have access?

Can we automate something with passports on what data products people can have access to?

A @McCormick, Jeffrey R HHHH question

How does TBOR work with this or will curation teams need to know what to do?

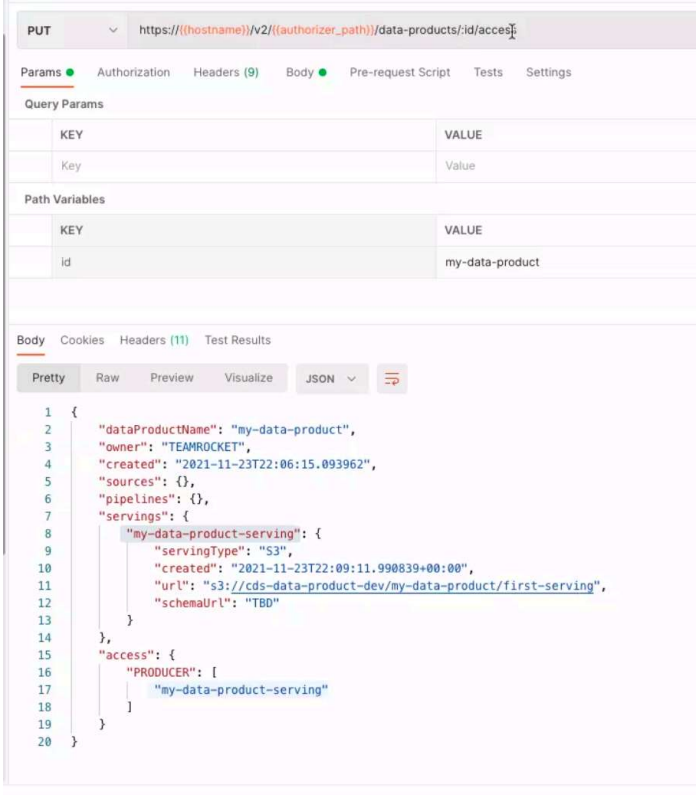
Will Tenant ID will need to be added to all data structures?

Getting access to a data product

Today, teams are able to grant access to a data product for you, but you must contact the owner of the data product to request they add your team.

Access at each specific serving, so if a data product serves at 3 s3 prefixes, you will need to get access to all 3 prefixes to see all the data.

Example adding access to a particular serving of a data product.



Removing access

You can also remove access to teams from a serving that you own as well.

## Building & Serving your own Data Product

When you register a Data Product, you get a location in S3 that is locked down to just your team for reading and writing.

Legacy 'domain buckets' are no longer needed but are supported for existing assets.

You then will serve up specific prefixes in that Data Product that you want to be discoverable by updating the definition of your Data Product.

**How do we ensure people who tokenize data not write it out again in a new data product?**

**How to ensure consistency across all data owners when creating data product definitions, metadata, etc?**

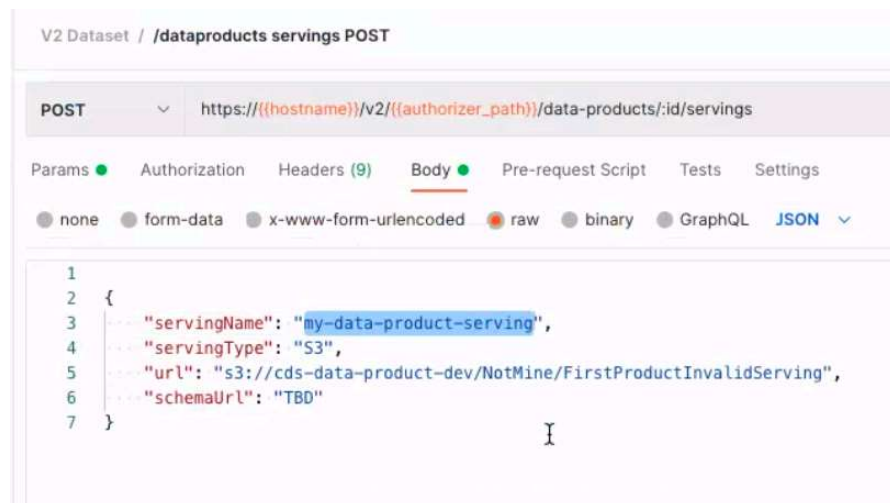
**What are our requirements?**

When producers publish data in a data mesh, they create an immutable data contract with these properties:

- Data type
- Physical schema
- Business characteristics
- Delivery frequency
- Data quality assertion
- Lifecycle policy
- ...

### Serving your data

To share your data, you must define a serving (which is currently required to be in the cds-data-product-env s3 bucket.)



You can define a serving outside the s3 bucket **but access is not able to be granted through our access management service.**

