



User Guide: Access Control Layer

What is the Access Control Layer?	2
How does it work high-level?	2
Migrating to the Access Control Layer	3
Migrating from Integrations to Collections	3
Migrating to the Access token	3
Just show me the code!	5
Frequently asked questions	8

Last updated: 23/01/2024



What is the Access Control Layer?

The Access Control Layer is a new and simplified approach to managing user access when embedding Luzmo assets in your product. It provides more flexibility in controlling which content is accessible to specific individuals.

How does it work high-level?

The Access Control Layer encapsulates different concepts ensuring a simple and flexible embedding process.

At the core of the layer there is an Embed token, which manages the different accesses to Luzmo assets.

You can give the users of your product access to:

- Dashboards
- Datasets
- Collections

You can mix and match different types of assets to better align with your user access requirements and your user management system.

We are also introducing Embed Filters. These filters used at the dataset level will only apply when your content is embedded. They can be configured through the UI and overridden through the Embed token request.



Migrating to the Access Control Layer

Migrating from Integrations to Collections

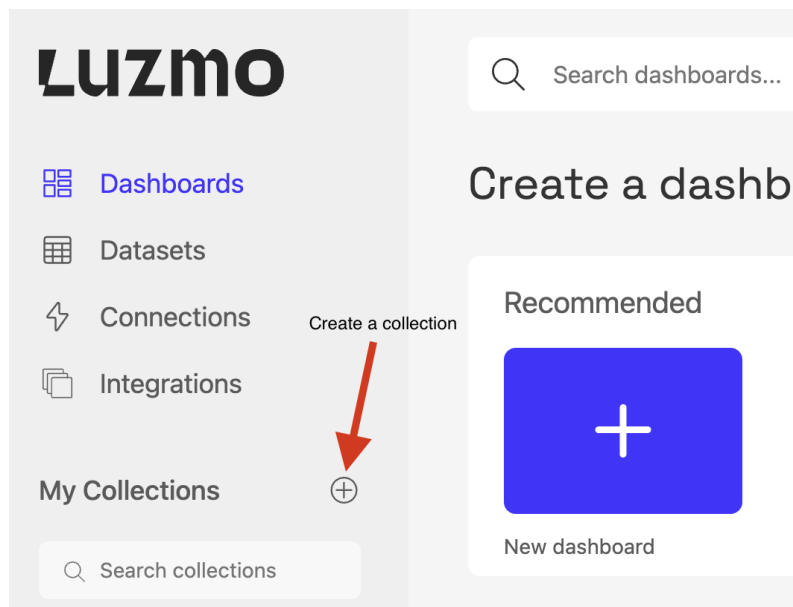
1. Create Collections based on your integrations. You can add dashboards AND datasets to the same Collection. Then you can give access to this Collection using the collection_id
2. Add access to specific dashboards or datasets outside of a Collection

Note: The user whose API token is used for token creation must have access to the items within the Collection. If not, the token creation will work but it will trigger a warning.

Migrating to the Access token

1. **Create Collections (optional but recommended):**

Grant access to datasets, dashboards and Collections:



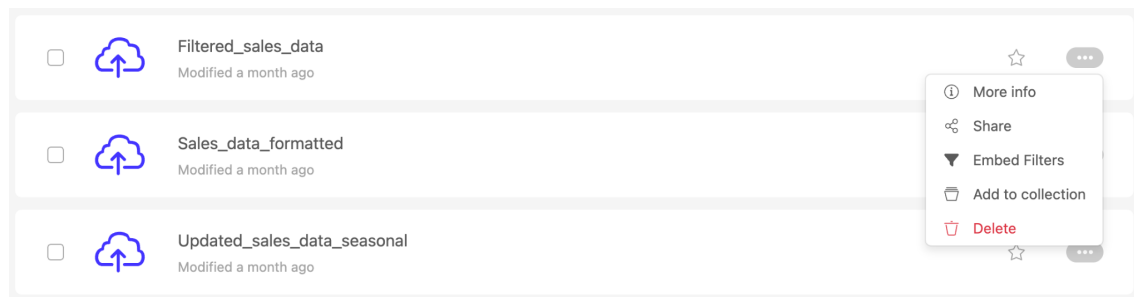
access: use this property to grant access to your assets. Specify the Collection(s), dashboard(s) and/or dataset(s) id(s) you want to give access to, as well as the corresponding rights associated with the asset.



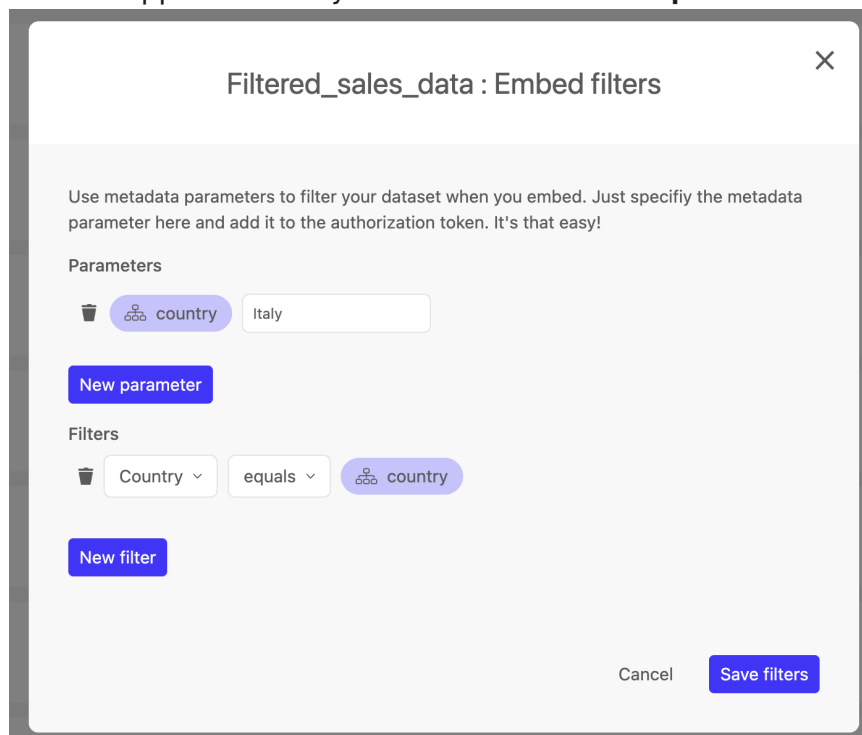
2. Create Embed Filters through the UI

Note: To have access to Embed Filters you need to have Organisation Owner rights

In your Datasets list menu, you can choose **Embed Filters**:



A modal appears where you can **create or edit a parameter and a filter**.



Override parameters through the Embed token

parameter_overrides: Specify the parameter name you want to override as well as the new value

3. Create the Embed token following the steps below



Just show me the code!

Note: examples within Luzmo UI coming soon!

How do I create an Embed token and grant access to a dashboard, dataset or collection?

Unset

```
curl https://api.cumul.io/0.1.0/authorization \
-H "Content-Type: application/json" \
-d @- << EOF
{
  "action": "create",
  "version": "0.1.0",
  "key": "< your API key >",
  "token": "< your API token >",
  "properties": {
    "type": "embed",
    "username": "randomName",
    "name": "randomName",
    "email": "test-random@luzmo.com",
    "access": {
      "collections": [
        {
          "id": "<collection_id>",
          "inheritRights": "use"
        }
      ],
      "datasets": [
        {
          "id": "<dataset_id>",
          "rights": "use"
        }
      ],
      "dashboards": [
        {
          "id": "<dashboard_id>",
          "rights": "use"
        }
      ]
    },
    "filters": [
      {
```



```
"clause": "where",
"origin": "global",
"securable_id": "< dataset id of a dataset used in the dashboard >",
"column_id": "< column id of a column in a dataset used in the dashboard >",
"expression": "? = ?",
"value": "< value to be applied to the expression >"
}
]
}
}
EOF
```

How inheritRights works?

When setting inheritRights you are providing access to all the items in a collection to the user. The user has to already have access to the items otherwise the embed user will not have access to the items and you will get a warning in the response.

How to override or clear a parameter?

```
Unset
curl https://api.cumul.io/0.1.0/authorization \
-H "Content-Type: application/json" \
-d@- << EOF
{
  "action": "create",
  "version": "0.1.0",
  "key": "< your API key >",
  "token": "< your API token >",
  "properties": {
    "type": "embed",
    "username": "randomName",
    "name": "randomName",
    "email": "test-random@luzmo.com",
    "parameter_overrides": {
      "id": [10],
```



```
"region": { "clear": true }  
},  
"access": {  
  "datasets": [  
    {  
      "id": "<dataset_id>",  
      "rights": "use"  
    }  
  ]  
}  
}  
}  
EOF
```

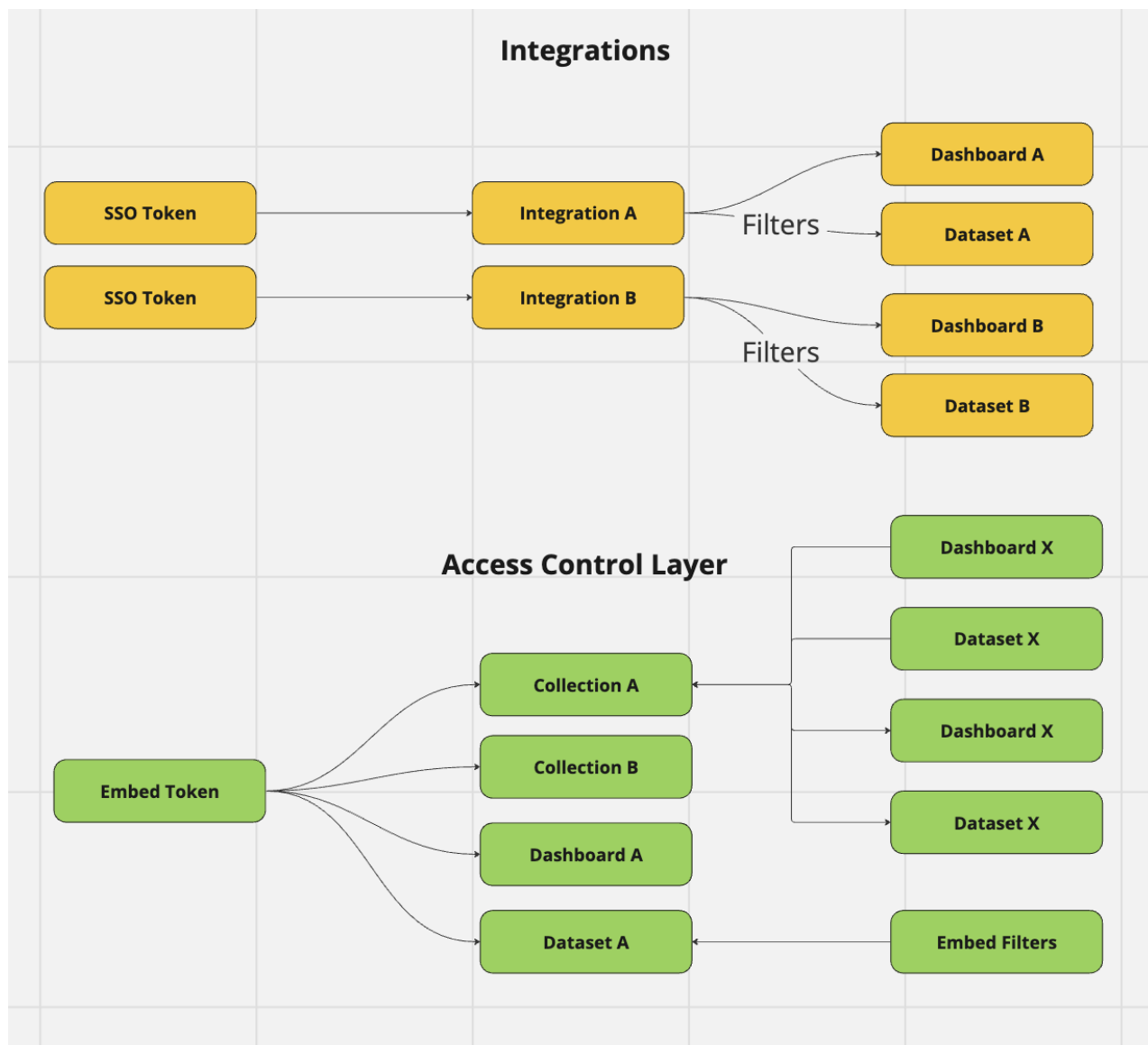


Frequently asked questions

Q: What is the difference between Integrations (SSO Token) and Access Control Layer (Embed token)?

A: By using the Embed token, you can grant access to multiple items including Collections, Dashboards and Datasets with more flexibility compared to the SSO Token.

While an SSO Token is limited to one integration, the Access Control Layer allows you to mix and match the types of content you want to give access to.





Q: Is it compulsory to change my integration?

A: Yes. Over the coming months, we will stop improving those features and they will be deprecated eventually. We recommend migrating to the Access Layer Control and updating your token to ensure you will get the latest updates.

Your existing integrations and/or "temporary" tokens will continue functioning for the time being. However, we strongly recommend opting in for the Access Layer Control to take advantage of the flexibility offered by this new approach.

Q: What are all the properties of the Embed token?

A:

- type // required
- access // required
- expiry // default 24 hours, UTC timestamps dropped support for '1 Day'
- username // required
- suborganization // default to username
- role // default to viewer
- name // required
- email // required
- inactivity_interval // integers, dropped support for '1 Day' syntax
- parameter_overrides
- environment
- ip
- screenmode
- currency_id
- width
- height
- metadata
- theme
- css
- account_overrides
- feature_overrides
- filters
- user_id
- theme_id
- timezone_id
- locale_id

Q: What happens if a dashboard or a dataset is added to a Collection after the Embed token creation?

A: There will be no impact on the token created. The rights are assigned at the time of the token creation which means that if an item is added to a collection or removed from a collection the access rights will stay the same.



Example:

- Collection A contains Dashboard A, Dataset A, Dashboard B and Dataset B
- Embed token is created with access to Collection A
- Token has access to Dashboard A, Dataset A, Dashboard B and Dataset B through the Collection A
- Dashboard X is added to the Collection A
- Token is not impacted and doesn't have access to Dashboard X

To update the access rights including Dashboard X a new embed token should be created.