

DataStore Integration Guide

About DataStores

Argus DataStores is a simple key/value database which is divided into **multiple named stores**, and for each store data is separated **per customer**. Some stores are marked as **global**, meaning that it cannot contain customer-specific data. The query API allows querying a single named store for data across multiple customers.

Datastores may be integrated with Argus Event Filters, allowing the Argus analysis system to distribute the datastores to processing nodes and event filters to use the data in event queries.

A datastore is configured by a **descriptor** which will control its behaviour and data retention. Datastores are either LOCAL or CENTRAL. LOCAL datastores have no data stored in the central Argus database and are only utilized in the distributed system. Only data of CENTRAL datastores can be added and queried through the API. A descriptor can configure a retention policy. Data which is older than the configured retention will be automatically deleted from the datastore.

Entries inside one datastore can be of type MAP or LIST. MAP entries are key/value pairs while entries of type LIST only consist of a key, i.e. a key exist inside a datastore or not. The values of MAP entries are unstructured and clients control their format. If a datastore is global entries are not bound to a customer. With customer-specific datastores the same key can exist inside the same datastore for different customers, i.e. a key is only unique for the same customer.

Argus controls access to datastores on a role-based model which allows granting read/write or read-only access to datastores per customer. Inside one datastore users can only access entries for the customer(s) they have been granted access to. Additionally, a user needs special permissions to access a global datastore.



Granular access per datastore is not supported. When granting users access to datastores for a customer, this grants the user access to ALL datastores for that customer.



Please read the [General Integration Guide](#) to learn the general concepts and common data structures used throughout the Argus API.

Detailed API documentation

The [Swagger API documentation](#) is always up-to-date and lets you try out any query with your user session or an API key.

- [About DataStores](#)
- [Detailed API documentation](#)
- [Integration guide](#)
 - [Listing available datastores](#)
 - [Querying a datastore](#)
 - [Fetching a datastore entry](#)
 - [Advanced search](#)
 - [Searching for specific keys](#)
 - [Search for range of keys](#)
 - [Search for all updates](#)
 - [Updating a datastore entry](#)
 - [Deleting a datastore entry](#)
 - [Updating multiple datastore entries](#)

Integration guide

Listing available datastores

Listing available datastores is done by querying the *descriptor* endpoint:

```
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/datastores/v1/descriptor
```

The endpoint will return metadata about available datastores:

```
{
  "data": [
    {
      "name": "myStore",
      "description": "Description of store",
      "dataType": "MAP",
      "behaviourType": "CENTRAL",
      "lastUpdatedTimestamp": 1501662018874,
      "lastUpdatedByUser": {...},
      "lifeTime": 259200000,
      "deleted": false,
      "globalData": false,
      "expireData": true
    }
  ], ...
}
```



Only datastores with `behaviourType: CENTRAL` can be queried and/or updated via the API.



Datastores with `globalData: true` cannot be queried or updated per customer. For these datastores data only exists for customer 0 (global).



Before interacting with a datastore it must be defined in Argus by creating a new descriptor. This is an administrative operation and only users with special permissions are allowed to manage datastore descriptors.

Querying a datastore

To query a datastore for entries use the `store` endpoint:

```
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/datastores/v1/store/myStore
```

By default, this will query the store for all keys and across all customers the current user has access to. The default result limit is 25.

To query for only a specific customer use the `customerID` parameter:

```
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/datastores/v1/store/myStore?customerID=1
```



To paginate a large result please see the [General integration guide](#).

Fetching a datastore entry

To fetch a specific key from a datastore use the key on the `store` endpoint. **Note** that unless otherwise specified, this will fetch the value for the key belonging to the current user's customer.

```
# fetch entry for "myKey" from "myStore" for the customer of the active user
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/datastores/v1/store/myStore/myKey

# fetch entry for "myKey" from "myStore" for customer 1
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/datastores/v1/store/myStore/myKey?customerID=1
```

Advanced search

Advanced search has access to all possible filtering parameters for datastores and follows the general advanced search structure as described in the [General integration guide](#).

Searching for specific keys

```
# search for all values on keys "key1" or "key2" across all available customers
curl -XPOST -H "Argus-API-Key: my/api/key" -H "Content-Type: application/json" https://api.mnemonic.no/datastores/v1/store/myStore/search -d '{
  "key":["key1","key2"]
}'
```

Search for range of keys

```
# search for all keys set in the range "key-a" to "key-h" (in lexical
ordering)
curl -XPOST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/datastores/v1/store/myStore/search -d '{
  "fromKey": "key-a",
  "toKey": "key-h"
}'
```

Search for all updates

```
# search for all updated or deleted keys since time 1521496196000 (in
milliseconds since epoch)
curl -XPOST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/datastores/v1/store/myStore/search -d '{
  "includeDeleted": true,
  "startTimestamp": 1521496196000
}'
```

See [Swagger API documentation](#) for more details on valid search parameters.

Updating a datastore entry

Update a single entry using the store *PUT* endpoint:

```
# put new value "myValue" for entry "myKey" for store "myStore" and the
customer of the active user
curl -XPUT -H "Argus-API-Key: my/api/key" https://api.mnemonic.no
/datastores/v1/store/myStore/myKey/myValue

# put new value "myValue" for entry "myKey" for store "myStore" and
customer 1
curl -XPUT -H "Argus-API-Key: my/api/key" https://api.mnemonic.no
/datastores/v1/store/myStore/myKey/myValue?customerID=1
```

Deleting a datastore entry

To delete datastore entries use the *DELETE* store endpoint. Each key is listed as a query parameter.

```
# delete entry "myKey" for store "myStore" and the customer of the active
user
curl -XDELETE -H "Argus-API-Key: my/api/key" https://api.mnemonic.no
/datastores/v1/store/myStore?key=myKey

# delete entry "myKey" for store "myStore" and customer 1
curl -XDELETE -H "Argus-API-Key: my/api/key" https://api.mnemonic.no
/datastores/v1/store/myStore?key=myKey&customerID=1

# delete entries "myKey" and "otherKey"
curl -XDELETE -H "Argus-API-Key: my/api/key" https://api.mnemonic.no
/datastores/v1/store/myStore?key=myKey&key=otherKey
```

Updating multiple datastore entries

To update multiple entries in a single operation use the bulk *PUT* endpoint. The customerID is optional. If omitted it will default to the current user's customer.

```
# put values for keys "myKey" and "otherKey" for the store "myStore" and
customer 1
curl -XPUT -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/datastores/v1/store/myStore -d '{
  "customerID":1,
  "entries":{
    "myKey": "myValue",
    "otherKey": "otherValue"
  }
}'
```