

DataStore Integration Guide

About DataStores

Argus DataStores is a simple key/value data store which is divided into **multiple named stores**, and for each store data is separated **per customer**. Some stores are marked as **global**, meaning that there is no customer-specific data.

The query API allows querying a named store across multiple customers.

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

DataStores are either LOCAL or CENTRAL. LOCAL data stores have no data stored in the central database, and are only relevant in the distributed system. The datastore descriptor will control the data retention and scope.

Argus controls access to data stores on a role based model, which allows granting read/write or read-only access to datastores per customer.



Granular access per data store is not supported. When granting users access to datastores for a customer, this grants the user access to ALL data stores 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 endpoints:

```
{
  "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).

Querying a datastore

To query a datastore, 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, 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 resultset, please see the [General integration guide](#)

Fetching a datastore entry

To fetch a specific key for a datastore, use the key on the `store` endpoint. **Note** that unless otherwise specified, this will fetch the value for the key set for the current users 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 follow 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
millis 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 users 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"
  }
}'
```