Case Integration Guide

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.

Integration guide

Fetching a case

Fetching a single case is simply done using the case ID

```
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2/case
/123456
```

If successful, the above invocation will return the case basic model:

```
{
  "data": {
    "id": 123456,
    "subject": "My testcase",
    "description": "This is the description of the case",
    "customer": { "id":1, "shortName":"mnemonic", ...},
    "service": { "id":6, "shortName":"support", ...},
    "type":"operationalIncident",
    "status":"pendingCustomer",
    "priority":"medium",
    ...
  }
}
```

All endpoints for fetching, searching/listing, updating and deleting a case return the same datamodel.

See Swagger API documentation for details on the returned data model.

Creating a case

To create a case, you need to specify the service, case type, subject and description:

```
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case -d '{
  "service": "support",
  "type": "operationalIncident",
  "subject": "My testcase",
  "description": "This is the description of the case"
}'
```

The `description` field may contain formatted HTML.

By default, the case is created for the customer bound to the current user. To specify a different customer, use the "customer" parameter.

```
"customer":"mycustomer"  # ID or shortname of the customer to use
```

Note that the "service" parameter used must be valid for the selected customer. See Fetching subscriptions below to list which services are valid for a customer.

See Swagger API documentation for details on valid request parameters, and a detailed description of the returned data model.

Creating a restricted case

To create a case which is restricted from the time it is created, the create request can specify the `accessMode` variable, and optionally add users/groups with explicit access to the ACL members:

```
  "accessMode": "explicit",
  "aclMembers": [ { "subjectID":45, "level":"write" } ]
```

See Managing case access for details on access mode and ACL members.

Updating a case

Updating the basic fields of a case is done with a PUT request to the case resource.
If no parameters are provided, no changes are performed. Similarly, for any parameter to this endpoint, a `null` value will cause no change to the current value.

The example below will increase the priority to *high*, and change the status of the case to *pendingSoc*.

```
curl -X PUT -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/123456 -d '{
  "priority": "high",
  "status": "pendingSoc"
}'
```

See Swagger API documentation for details on valid request parameters.

Related endpoints

Other endpoints related to updating/modifying a case

- Adding a comment
- Adding an attachment
- Closing a case
- Moving the case to another service, case type and/or customer
- Managing case access
- Managing case tags

Restricted fields

Some fields only permitted to update by users which are granted the TECH role for the case:

- assignedTech
- reporter
- subject (can be changed by case owner)
- description (can be changed by case owner)

Attempts to update restricted fields will result in a 403 error code, with a FIELD_ERROR message explaining the error.

See Understanding case access for more details on access controls.

Update with comment

Adding a comment is a separate endpoint, but can also be added as part of a case update by setting the `comment` parameter.

Closing a case

Closing a case is a separate transition, which also triggers other notifications. When closing the case, an optional `comment` can be added to the case.

```
curl -X PUT -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/123456/close -d '{
  "comment": "Closing this case"
}'
```

See Swagger API documentation for details on valid request parameters.

Searching for cases

Searching for cases can be done using the *simple search* GET endpoint or the *advanced search* POST endpoint.

Please read the General Integration Guide to learn about general concepts for search endpoints.

Simple search

For simple search, the valid filtering parameters can be added as query parameters, which will ANDed together for each parameter.
If any parameter name is repeated, all the values for that parameter name will be combined into one disjunction (OR-statement)

```
# search for cases with service "ids", customer "mycustomer" and status
pendingSoc
curl -X GET -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2
/case?service=ids&customer=mycustomer&status=pendingSoc
```

```
# search for cases with service "ids", which are bound to either customer
ID 1 or 2
curl -X GET -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2
/case?service=ids&customer=1&customer=2
```

Parameter	Valid values	Example

customer	Customer ID or shortname	customer=1 # search for cases for customer ID 1
		customer=mnemonic # search for cases for customer mnemonic (will be resolved to this customers customerID)
		customer=1&customer=2 # search for cases for customerIDs 1 or 2

service	Numeric service ID
	
	Service shortname
		service=support # search for cases for support service
		service=6 # search for cases for support service (by supportservice numeric ID)
		service=support&service=ids # search for cases for services support or ids

status	pendingSoc pendingCustomer waitingSoc waitingCustomer pendingVendor pendingClose closed	status=closed # search for closed cases status=pendingSoc&status=waitingSoc # search for cases with status "pending soc" or "waiting soc"

type	change informational securityIncident operationalIncident	type=change # only cases of type change type=securityIncident&type=operationalIncident # cases of type security or operational incident
keywords	any keywords	keywords=test # search for cases with the word "test" keywords=test&keywords=malware # search for cases with both words "test" and "malware" (default match strategy is "ALL")
limit	0..100000 **Note: default is 25**	limit=0 # unlimited, up to system limit limit=1 # at most 1 result limit=25 # at most 25 results (which is default) limit=100000000 # invalid, system query limit is 100000
offset	0..100000	offset=0 # do not skip any records, this is default offset=25 # skip first 25, return next 25

Advanced search

Advanced search has access to all possible filtering parameters for case, and follow the general advanced search structure as described in the General integration guide.

As described there, multiple parameters in one criteria object are ANDed together. Multiple values for a single parameter are ORed together.

```
# search for cases with service "ids", customer ID 1 or 2, status
"pendingCustomer" or "waitingCustomer" and some keyword match for the word
"test"
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/search -d '{
  "service":["ids"],
  "customer":[1,2],
  "status":["pendingCustomer","waitingCustomer"],
  "keywords":["test"]
}'
```

See Swagger API documentation for more details on valid request parameters.

Subcriteria

Subcriteria are discussed in detail in the General integration guide. We provide some examples related to the Case API here, but the concepts for subcriteria are described more in detail there.

Using subcriteria allows you to fetch several different dimentions of data in one query, or express which data to exclude. By default, subqueries will be combined with an "OR" logic.

```
# search for cases that either have status "pendingSoc" OR have priority
"high". The customer criteria applies to both the subcriteria.
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/search -d '{
  "customer":["mycustomer"],
  "subCriteria": [
     {"status":["pendingSoc"]},
     {"priority":["high"]},
   ]
}'
```

Exclude subcriteria

Subqueries with *exclude=true*, defines a set of criteria for cases to exclude.

```
# search for cases for customer mnemonic, and exclude those with status
"pendingSoc" or "pendingCustomer"
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/search -d '{
  "customer":["mnemonic"],
  "subCriteria": [
     {"exclude":true,"status":["pendingSoc","pendingCustomer"]}
   ]
}'
```

Use a exclude subquery to easily exclude closed cases, if you want to only fetch open cases.

```
# search for cases for customer ID 1, and exclude those with status
"pendingSoc" or "pendingCustomer"

curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type:
application/json" https://api.mnemonic.no/cases/v2/case/search -d '{

  "customer":[1],

  ...

  "subCriteria": [

     {"exclude":true,"status":["closed"}

   ]

}'
```

Searching for cases by user

Each case has a number of user-related fields:

- reporter
- assigned user
- assigned tech
- creator (generally equal to reporter)
- publisher (generally equal to reporter)
- last updated by user
- closed by user
- all users who have added comments

To search for cases across these fields, use the "userID" search parameter. By default, it will search across all these fields for cases where the userID parameter contains a user listed in one of these fields.

```
# search for cases where userID 1, 2 or 3 are listed in any of the user
fields
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/search -d '{
  "userID":[1,2,3],
  ...
}'
```

To search for cases by specific users in specific fields, use the parameter userFieldStrategy, which determines which field(s) to search.

```
# search for cases which were created by userID 1, 2 or 3
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/search -d '{
  "userID":[1,2,3],
  "userFieldStrategy": ["createdByUser"]
  ...
}'
```

See Swagger API documentation for more details on valid search parameters.

Searching for cases by time

Please see the General integration guide for examples and details on use of the `startTimestamp`, `endTimestamp` and `timeFieldStrategy` fields.

Searching for cases by keywords

Please see the General integration guide for examples and details on use of the `keywords`, `keywordMatchStrategy` and `keywordFieldStrategy` fields.

Managing comments

Listing comments

Comments on a case can be listed using the comments endpoint:

```
#fetch comments, default limit of 25
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2/case/123456/comments
#fetch all comments
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2/case/123456/comments?limit=0
```

Adding a comment

Simply add a comment to a case:

```
#fetch comments, default limit of 25
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application/json" https://api.mnemonic.no/cases/v2/case/123456/comments -d '{
  "comment":"My comment"
}'
```

To update status/priority while adding a comment, use the update endpoint with parameter `comment`.

Fetching events

To fetch events for a case, use the Events endpoint `https://api.mnemonic.no/events/v1/case/<caseid>`

See Event Integration Guide

Managing attachments

Listing attachments

Attachments on a case can be listed using the attachments endpoint. This will return metadata about the attachments.

```
#fetch metadata about attachments, default limit of 25
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2/case/123456/attachments
#fetch metadata about all attachments
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2/case/123456/attachments?limit=0
```

Downloading an attachment

To download the contents of an attachment, use the attachment download endpoint. This will return the raw attachment, with the same content-type as the attachment originally uploaded.

```
#fetch raw attachment
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2/case
/123456/attachments/12345678-1234-ABCD-123456789ABC/download > /tmp
/attachmentfile
```

Adding an attachment

To upload an attachment, the attachment must be added to a base64-encoded POST request:

```
#upload attachment to case
curl -XPOST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json"  https://api.mnemonic.no/cases/v2/case/123456/attachments -d '{
  "name": "filename.log",
  "mimeType": "text/plain",
  "data": "YWJjZGVm"
}'
```

The `data` parameter is a base64-encoding of the binary attachment file.

Understanding case access

Access to cases is by default `roleBased`, meaning that users with rolebased access to the customer and service of the case, has access to it. There are different roles for `read` and `write` access, and a separate role for "tech users", generally meaning users representing the service provider.

A case can be given `restricted` access mode, meaning that access to customer users are explicit, and not rolebased. Tech users will still have normal access to the case. This use case is for cases which are handled by the service provider as normal, but which are sensitive for the customer.

The explicit access mode means that only users/user groups which are on the explicit access list have access to the case. This also applies to tech users.
Note: System administrators may still access the case without explicit access.

The explicit access list may also extend normal role based access, by giving explicit users/user groups access to a case, which do not have role based access to the case at all.

Access levels

For each individual case, a user may be granted one of the following access levels, depending on role-based and/or explicit access settings:

- **Read** privileges
  - permits fetching all case details
- **Write** privileges
  - Read privileges
  - updating case priority, status, reference, category, assignedUser
  - adding comments
  - adding/removing tags, links and attachments.
- **Owner** privileges
  - Write privileges
  - changing access mode
  - granting/revoking access
  - changing the reporter of a case (implicit owner)
  - changing *subject* and *description* on the case
  - changing watcher settings for other users

This table maps out which users get these roles:

Access mode	Read access	Write access	Owner access
roleBased	Case reporter Users with service read role All ACL members	Case reporter Users with service write role ACL members with accesslevel `write`	Case reporter Administrators
writeRestricted	Case reporter Users with service read role All ACL members	Case reporter **Users with service tech role** ACL members with accesslevel `write`	Case reporter Administrators

readRestricted	Case reporter **Users with service tech role** All ACL members	Case reporter Users with service tech role ACL members with accesslevel `write`	Case reporter Administrators
explicit	Case reporter All ACL members	Case reporter All ACL members	Case reporter Administrators

Access roles

In addition to access levels, there are also a set of access roles: **user, tech** and **admin**.

User role

All users with read and/or write access to a case, assume the "user" role.

Tech role

Users with the role "tech" have additional privileges, which ordinary users do not:

- update *restricted fields*
- view or create *internal comments*
- update case workflows
- create an "unpublished" case, which are not visible to non-tech users
- publish an "unpublished" case
- change watcher settings for other users.
- delete comments and see deleted objects

Any change to the case also requires access level "write". A tech user with only "read" access level, can only view additional information.

Admin role

Users with the admin role have additional privileges, which users and techs do not

- assume "owner" role for the case, allowing override of case access settings and explicit access grants.
- users with "admin" role always have full write access level (implied by "owner"), as well as "tech" role.

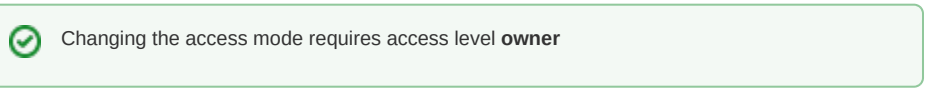
Determining access level and role

To see which access level the current user has on a particular case, each case contains a field "currentUserAccess":

```
"currentUserAccess": {
  "level": "write",
  "role": "user"
}
```

- The field "level" may return "read", "write" or "owner". Write implies read. Owner implies both read and write.
    - If the user has no read access, the case will not be returned from the API
- The field "role" may return "user", "tech" or "admin". Tech implies user. Admin implies both tech and user.

Changing case access mode



To change the access mode of a case, use the access PUT endpoint:

```
curl -X PUT -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json"  https://api.mnemonic.no/cases/v2/case/123456/access -d '{
  "accessMode": "readRestricted"
}'
```

Granting case access

> Granting case access requires access level **owner**

To grant access to a case for a user or user group, POST to the access endpoint with the subjectID of the user or group. The `level` parameter will determine the level of access granted to that user or group.

```
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json"  https://api.mnemonic.no/cases/v2/case/123456/access -d '{
  "subjectID": <userID>,
  "level": "read"
}'
```

List case access

Depending on the case `accessMode`, different users have *role-based* access to the case, according to the table above. Use the Get case endpoint to see the `accessMode` of a case.

Explicit access granted to single users or user groups can be listed using the access endpoint:

```
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2/case
/123456/access
```

Revoking case access

> Revoking case access requires access level **owner**

To revoke access from a user on the case ACL, use the DELETE access endpoint with the ID of the ACL entry to delete:

```
curl -X DELETE -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases
/v2/case/123456/access/c2134bd3-9d88-4d6c-a395-d8d2241b4cbd
```

> Please note, that users with role based access to a case cannot be explicitly revoked. To limit users with role based access from accessing a case, you need to change to a stricter `access Mode` on the case.

Managing case tags

Tags are a kind of labels to add structured keywords to cases. Each tag has a key and a value. A case may have multiple tags, and even multiple tags with the same key.

Listing tags

```
#fetch tags, default limit of 25
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2/case
/123456/tags
#fetch all tags
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2/case
/123456/tags?limit=0
```

Tags are returned with some metadata:

```
{
...
"data": [
  {
    "id": "c2134bd3-9d88-4d6c-a395-d8d2241b4cbd",
    "addedTimestamp": 1520800381632,
    "addedByUser": {...},
    "key": "mykey",
    "value": "myvalue1",
    "flags": []
  },
...
]
}
```

Adding a tag

```
#adds two tags, key=value1 and key2=value2
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/123456/tags -d '{
  "tags":["mykey/value1", "mykey2/value2"]
}'
#equivalent, using the full tag encoding
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/123456/tags -d '{
  "tags":[
    {"key":"mykey", "value":"value1"},
    {"key":"mykey2", "value":"value2"},
  ]
}'
```

Removing a tag

A tag can be removed by key/value, or by the ID of the tag itself.

```
curl -X DELETE -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases
/v2/case/123456/tags/mykey/value1
#equivalent, using the tags ID
curl -X DELETE -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases
/v2/case/123456/tags/c2134bd3-9d88-4d6c-a395-d8d2241b4cbd
```

Moving a case

> Moving a case requires access role **tech** for the service subscription, in addition to the special privileges moveCase
>
> If moving the case to another service and/or customer, the operation requires **tech** access role also for the target service subscription.

This endpoint is used to change the case type, service or customer of a service.

This example moves the case to the service `ids` for customer `"newcustomer"`:

```
curl -X PUT -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/123456/move -d '{
  "customer": "newcustomer",
  "service": "ids",
  "type": "securityIncident"
}'
```

⚠ When moving to another service, the `caseType` must be valid for the target service.

See Fetching services below to list valid services and their case types.

⚠ If the case is assigned a category, that category must also be valid for the target case type and /or service.

If not, the request must also unassign the category (set `category: null`) or assign a new category which is valid for the target case type/service.

See Fetching categories below to list valid services and their case types.

Required permissions

See Swagger API documentation for details on valid request parameters.

Fetching services

To list possible services to submit to, and which case types they support, use the services endpoint:

```
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2
/service
# will return
{
  "data": [
    "id": 6,
    "shortName": "support",
    "caseTypes": ["operationalIncident", "securityIncident", "change",
"informational"],
    ...
  ],
  ...
}
```

To fetch only a specific service, you can use the service GET endpoint `https://api.mnemonic.no/cases/v2/service/ID` where ID can be the service numeric ID or shortname.

Fetching service subscription

To use a specific service, a customer must have a valid *service subscription*. To check which service subscriptions a customer has, use the servicesubscription endpoint with a "customer" query parameter:

```
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2
/servicesubscription?customer=mnemonic
# will return
{
  "data": [
    "id": "010afb43-323e-463a-a3fc-e93336488798",
    "service": {
        "id": 2,
        "name": "Security Monitoring",
        ...
    },
    "customer": {
        "id": 1,
        "name": "mnemonic",
        ...
    },
    ...
    "currentUserAccess": {
        "level": "write",
        "role": "user"
    }
    ...
  ],
  ...
}
```

The currentUserAccess field of the service subscription object provides information about the role based access level for the specified service and customer.

To create a new case, the current user must have at least access level "write".

Fetching categories

To list available categories, use the category endpoint:

```
curl -H "Argus-API-Key: my/api/key" https://api.mnemonic.no/cases/v2
/category
# will return a list of categories
{
  "data": [
    {
      "id": 61,
      "name": "Firewall operational incidents",
      "shortName": "firewall-operational",
      ...

      "bindings": [
        {
          ...
          "service": {
            "id": 6,
            "name": "Support",
            "shortName": "support"
          },
          "caseTypes": [
            "operationalIncident"
          ]
        }
      ],
      ...
    }
  ],
  ...
}
```

The category listed specify valid bindings to services and case types. In the example above, the category "firewall-operational" is bound to the service `support`, for case type `operationalIncident`. This means that it is valid to use for cases with this service and caseType.

One category may specify multiple bindings, and possibly multiple case types per binding.

To fetch only a specific category, you can use the category GET endpoint `https://api.mnemonic.no/cases/v2/category/ID` where ID can be the category numeric ID or shortname.