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.

Uploading attachments before creating a case

Uploading attachments is a separate endpoint, to allow uploading potential large attachments, and to limit the size of the create request. However, sometimes you may want to add attachments to a case while creating it (as opposed to adding the attachments AFTER), for example to add images to the case description, the image src tag must point to a valid image URI.

To do this, you can use the *prepare case flow*:

- Prepare a new case
- Upload attachments to the new case
- Create the prepared case, which will contain the already uploaded attachments

Example:

```
# prepare a new case
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/prepare
{ "data": { "caseID": 123456 } }

# upload file to case
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/123456/attachments/upload
/myfile.txt --data-binary @myfile.txt

# Then actually create the case
curl -X POST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json" https://api.mnemonic.no/cases/v2/case/123456 -d
'{ "service": "support", "type": "operationalIncident", "subject": "My
testcase", "description": "This is the description of the case" }'
```

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.

Case status

A case can assume the following statuses:

Status	Description
pendingSoc	Waiting for SOC to work on the case.
pendingCustomer	Waiting for Customer to work on the case.
workingCustomer	In progress by Customer. **Note:** when setting this status, if the case has no assigned customer user, the ticket will automatically be assigned to the current user.
workingSoc	In progress by SOC. **Note**: this status can only be set by SOC users. **Note:** when setting this status, if the case has no assigned "tech user" (SOC user), the ticket will automatically be assigned to the current user.
pendingVendor	Waiting for 3rd party vendor.

pendingClose	Ready to be closed. **Note:** for mnemonic services, cases in this status will be automatically closed after 90 days of inactivity.
closed	Case is closed.

Description and comment markup

Both the `description` field of a case, as well as the `comment` of any comment, allows HTML markup.

The HTML content is *sanitized* upon submission, so any client should expect that the HTML content will change after submitted.
If the client expects to keep the HTML content in sync with a source state, it has to update its own state with the result of the submission (e.g. read the sanitized `description` or `comment` from the result).

Allowed HTML tags are

- a - anchor
- img - either remote URI or data URI
- h1 .. h6 header tags
- br, span, div, p
- ul, ol, li
- table, tr, td, th, colgroup, caption, col, thead, tbody, tfoot
- b, strong, i, em, del, s, ins, u
- pre, code, blockquote

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 workingSoc workingCustomer 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.

Or, use the streaming endpoint to upload a binary attachment:

```
#upload raw file 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/upload/myfile.txt?mimeType=text/plain --data-binary @myfile.txt
```

Case watchers

A *watcher* is a user who will be notified about changes to the case. There are three different type of watchers

- A *default watcher*, who will be automatically added to the case based on service, case type and case priority. This is managed by administrators as *customer contacts*.
- An explicit user watcher, where a specific user or user group is explicitly added as a watcher for a specific case
- An explicit mailbox watcher, where an explicit email address is added as a watcher for a specific case.

A watcher may be configured to send *email* or *sms* alerts. Users that have configured the Argus Mobile app, may also enable *push notifications* on case changes.

Adding an explicit user watcher

```
#upload raw file to case
curl -XPOST -H "Argus-API-Key: my/api/key" -H "Content-Type: application
/json"  https://api.mnemonic.no/cases/v2/case/123456/watchers  -d '{
  "userOrGroup": "username"
}'
```

By default, the watcher will be added as an *email watcher*. Use parameter `type=SMS` to request SMS updates.
Users with Argus Mobile Push notifications enabled will be notified regardless of type.

Verbose watchers

By default, the watcher will receive an email/message which only contains the ticket number, and a link to the ticket.
To enable actual contents in the notification, use the `verbose` option.

This option will be automatically enabled for users having verbose notifications as default in their user preferences.

Watchers and access control

Note that to be added as a watcher, the user/group must have access to the case, either role-based, or by explicit access.
When removing access to a case from a user/group, any watcher entries will also be removed.

When adding explicit access to a case, by default the granted user/group will also be added as a watcher. To disable this behaviour, use the option `addWatcher=false`.

See Understanding case access below.

Understanding case access

See Understanding Case Access Control

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.