

Template API

Allows the creation and management of **Storage Unit** configuration templates.

On this page:

A configuration template accepts settings for:

- Content Processing Modules
- Content Processing flag
- Encryption flag
- Compression flag
- Reprocessing Queue flag

- [Add Template](#)
- [Get Template](#)
- [Delete Template](#)
- [List Templates](#)

When defining a template, a regex pattern is assigned to the template. When creating a new storage unit, if the name of the storage unit matches the pattern of the template, it will be created with the template configuration settings. Templates will be sorted by a user-defined priority. In case multiple patterns match the storage unit name, the template with the highest priority will be applied (0 is the highest priority). Priorities can be reassigned, by saving again the template with a different priority number.

Add Template

Adds or updates a configuration template. The template name is the identifier for update operations.

Request

The add template PUT/POST request requires the template name in the URL and a body with the fields:

- **pattern**: a regex for matching storage unit names.
- **priority**: a number with the template priority in the template list. 0 is the highest priority. Calling **template/list** will return all templates sorted by priority.
- **settings**: the content processing modules configuration as described in [Configure Storage Unit](#).
- **contentProcessing**: flag to enable/disable content processing triggers. If enabled, modules configured in settings will execute for the configured scopes.
- **encryptContent**: flag to enable/disable content encryption.
- **compressContent**: flag to enable/disable content compression.
- **reprocessingQueue**: flag to enable/disable background reprocessing.

```
PUT template/add/<template-name>
{
  "pattern": ".*_test",
  "priority": 0,
  "settings": {
    "modules": [
      {
        "connector": [
          {
            "module": "FieldMapping"
          },
          {
            "settings": {
              "elasticsearch-index": "srindex",
              "elasticsearch-type": "srdoc"
            },
            "module": "HTTPEPublisher"
          }
        ]
      }
    ],
    "settings": {
      "elasticsearch-hosts": "localhost:9200",
      "elasticsearch-use-ssl": false,
      "elasticsearch-bulk-size": "1mb"
    }
  },
  "contentProcessing": true,
  "encryptContent": true,
  "compressContent": true,
  "reprocessingQueue": true
}
```

Response

If the template was created successfully, a 200 response code with an OK message is returned.

```
{ "message": "OK" }
```

Get Template

Returns the template configuration.

Request

The get template GET request requires the template name in the URL.

```
GET template/get/<template-name>
```

Response

If the template exists, a 200 response code with the template configuration is returned.

```
{
  "pattern": "*_test",
  "priority": 0,
  "settings": {
    "modules": [
      {
        "connector": [
          {
            "module": "FieldMapping"
          },
          {
            "settings": {
              "elasticsearch-index": "srindex",
              "elasticsearch-type": "srdoc"
            },
            "module": "HTTPEPublisher"
          }
        ]
      }
    ],
    "settings": {
      "elasticsearch-hosts": "localhost:9200",
      "elasticsearch-use-ssl": false,
      "elasticsearch-bulk-size": "1mb"
    }
  },
  "contentProcessing": true,
  "encryptContent": true,
  "compressContent": true,
  "reprocessingQueue": true
}
```

If the template doesn't exist, a 400 response is returned.

```
{ "message": "TEMPLATE_DOESNT_EXIST" }
```

Delete Template

Silently deletes a template by name.

Request

The delete template DELETE/POST request requires the template name in the URL.

```
DELETE template/delete/<template-name>
```

Response

A 200 response code with an OK message is returned when the operation completes.

```
{ "message": "OK" }
```

List Templates

Gets the list of available templates sorted by priority. For each template, the name, pattern and priority are returned.

Request

No parameters are required for the list GET request.

```
GET template/list
```

Response

A 200 response code with the list of available templates is returned sorted by priority (0 is highest). An empty list is returned if there are no templates.

```
[
  {
    "name": "test8",
    "pattern": ".*_test8",
    "priority": 0
  },
  {
    "name": "test9",
    "pattern": ".*_test9",
    "priority": 1
  }
]
```