magmodules



REST API

FAQ Manager for Magento 2

The FAQ Manager exposes its questions and topics through Magento's REST API, so an external system (a support dashboard, a PIM, a content tool) can create and publish FAQs and put them on the right product and category pages without touching the Magento admin. All endpoints require an admin bearer token and the `Magmodules_Faq::faq_manage` ACL resource; writing topics requires `Magmodules_Faq::topic_manage`.

Authentication

```
# Get an admin token
curl -X POST https://your-store.com/rest/V1/integration/admin/token \
  -H "Content-Type: application/json" \
  -d '{"username": "admin", "password": "admin123"}'

# Use it on every request
curl -H "Authorization: Bearer <token>" \
  https://your-store.com/rest/all/V1/faq/questions?searchCriteria=
```

An integration token (System > Extensions > Integrations) works the same way and is the better choice for a system that calls the API unattended.

Store views

Questions, answers and topic names are stored per store view. The store view of a request is the store code in the URL:

URL	Scope
`/rest/all/V1/...`	the default scope (store 0), which every store view falls back to
`/rest/default/V1/...`	the store view with code `default`; replace with the code of the store view you want
`/rest/V1/...`	the default store view of the website, **not** the default scope

Reading through a store code returns that store view's own text, or the default-scope text when the store view has none. Writing through a store code stores the text for that store view only. Create your FAQs through `/rest/all/V1/` and add store-view translations through `/rest/<code>/V1/` afterwards; a FAQ that exists for one store view only is invisible everywhere else.

`store_id` in a payload overrides the store code of the URL.

Questions

Fields

Field	Type	Write	Description
`faq_id`	int	read-only	Identifier, assigned on create
`topic_id`	int or null	yes	The topic the FAQ belongs to; `null` for none
`question`	string	required on create	The question, for the store view of the request
`short_answer`	string	yes	The short answer shown in listings
`answer`	string	yes	The full answer, HTML allowed
`is_active`	int	yes	`1` published, `0` hidden
`sort_order`	int	yes	Position within the topic, lowest first
`product_ids`	int[]	yes	The products the FAQ is shown on, by Magento product id
`product_skus`	string[]	yes	The same products by SKU; combined with `product_ids` when both are sent
`category_ids`	int[]	yes	The categories the FAQ is shown on
`store_id`	int	yes	The store view the text belongs to; defaults to the store code of the URL
`view_count`	int	read-only	How often the accordion was opened on the storefront
`created_at`, `updated_at`	string	read-only	Timestamps

A write only touches the fields it carries. Sending `is_active` alone publishes or unpublishes a FAQ and leaves its text and links alone. `product_ids`, `product_skus` and `category_ids` replace the links when present: an empty list unlinks everything, a missing key changes nothing.

Create a question and put it on a product and a category

```
POST /rest/all/V1/faq/questions
```

```
curl -X POST -H "Authorization: Bearer <token>" -H "Content-Type: application/json" \
  https://your-store.com/rest/all/V1/faq/questions \
  -d '{
    "entity": {
      "topic_id": 1,
      "question": "How long does delivery take?",
      "short_answer": "1-2 working days.",
      "answer": "<p>Orders placed before 16:00 ship the same day.</p>",
      "is_active": 1,
      "product_skus": ["24-MB01", "24-MB02"],
      "category_ids": [3]
    }
  }'
```

The response is the stored FAQ, including its new `faq_id` and the resolved `product_ids`. An unknown SKU, product id, category id or topic id is refused with a `400` that names it; nothing is stored in that case.

Publish or unpublish

```
PUT /rest/all/V1/faq/questions/:faqId
```

```
curl -X PUT -H "Authorization: Bearer <token>" -H "Content-Type: application/json" \
  https://your-store.com/rest/all/V1/faq/questions/5 \
  -d '{"entity": {"faq_id": 5, "is_active": 1}}'
```

Change the links

```
# Replace the product links, keep the categories
curl -X PUT -H "Authorization: Bearer <token>" -H "Content-Type: application/json" \
  https://your-store.com/rest/all/V1/faq/questions/5 \
  -d '{"entity": {"faq_id": 5, "product_skus": ["24-MB01"]}}'

# Remove the FAQ from every category
curl -X PUT -H "Authorization: Bearer <token>" -H "Content-Type: application/json" \
  https://your-store.com/rest/all/V1/faq/questions/5 \
  -d '{"entity": {"faq_id": 5, "category_ids": []}}'
```

Translate for a store view

```
curl -X PUT -H "Authorization: Bearer <token>" -H "Content-Type: application/json" \
  https://your-store.com/rest/de/V1/faq/questions/5 \
```

```
  -d '{
    "entity": {
      "faq_id": 5,
      "question": "Wie lange dauert die Lieferung?",
      "short_answer": "1-2 Werktage.",
      "answer": "<p>Bestellungen vor 16:00 Uhr werden am selben Tag versandt.</p>"
    }
  }'
```

Fields left out start from the default-scope text, so a store view can override the answer alone.

Read a question

```
GET /rest/all/V1/faq/questions/:faqId
GET /rest/<code>/V1/faq/questions/:faqId
```

```
curl -H "Authorization: Bearer <token>" https://your-store.com/rest/all/V1/faq/questions/5
```

```
{
  "faq_id": 5,
  "topic_id": 1,
  "sort_order": 0,
  "is_active": 1,
  "view_count": 12,
  "created_at": "2026-11-01 10:09:38",
  "updated_at": "2026-11-01 10:09:39",
  "store_id": 0,
  "question": "How long does delivery take?",
  "short_answer": "1-2 working days.",
  "answer": "<p>Orders placed before 16:00 ship the same day.</p>",
  "product_ids": [5, 6],
  "product_skus": ["24-MB01", "24-MB02"],
  "category_ids": [3]
}
```

List and search questions

```
GET /rest/all/V1/faq/questions
```

Standard Magento `searchCriteria` : filters, sorting and paging. `question` , `short_answer` and `answer` are filterable and sortable, and a `store_id` filter selects the store view the text is read for.

```
# Every question in topic 1, by sort order
curl -H "Authorization: Bearer <token>" \
  "https://your-store.com/rest/all/V1/faq/questions?searchCriteria[filterGroups][0][filters][0][field]=top

# Questions whose text mentions delivery
curl -H "Authorization: Bearer <token>" \
  "https://your-store.com/rest/all/V1/faq/questions?searchCriteria[filterGroups][0][filters][0][field]=que

# Page 2 of 20
curl -H "Authorization: Bearer <token>" \
  "https://your-store.com/rest/all/V1/faq/questions?searchCriteria[pageSize]=20&searchCriteria[currentPage
```

Delete a question

```
DELETE /rest/all/V1/faq/questions/:faqId
```

Removes the FAQ with its translations and links. Returns `true`.

Topics

Fields

Field	Type	Write	Description
`topic_id`	int	read-only	Identifier
`name`	string	required on create	The name, for the store view of the request
`url_key`	string	yes	URL slug on FAQ pages; left empty, it is derived from the name (`Shipping & Returns` becomes `shipping-returns`, a taken key gets `-2`, `-3`, ...)
`icon`	string or null	yes	File name of an icon uploaded through the admin
`position`	int	yes	Order on FAQ pages, lowest first
`is_active`	int	yes	`1` visible, `0` hidden
`store_id`	int	yes	The store view the name belongs to; defaults to the store code of the URL

Field	Type	Write	Description
`created_at`, `updated_at`	string	read-only	Timestamps

Endpoints

```
GET    /rest/all/V1/faq/topics           list, with searchCriteria (name is filterable and sortable)
GET    /rest/all/V1/faq/topics/:topicId  one topic
POST   /rest/all/V1/faq/topics           create
PUT    /rest/all/V1/faq/topics/:topicId  update; fields left out keep their value
DELETE /rest/all/V1/faq/topics/:topicId  delete; the topic's FAQs stay and lose their topic
```

```
curl -X POST -H "Authorization: Bearer <token>" -H "Content-Type: application/json" \
  https://your-store.com/rest/all/V1/faq/topics \
  -d '{"entity": {"name": "Shipping", "url_key": "shipping", "is_active": 1}}'
```

Pages

A FAQ page is a storefront page on its own URL that lists topics with their questions.

Fields

Field	Type	Write	Description
`page_id`	int	read-only	Identifier
`url_key`	string	required on create	The path the page answers on, without a leading slash; must be unique
`title`	string	required on create	The page title, for the store view of the request
`intro`	string or null	yes	Text above the topics, HTML allowed
`meta_title`, `meta_description`, `meta_keywords`	string or null	yes	Head meta tags; the title is used when `meta_title` is empty
`page_layout`	string or null	yes	A Magento page layout such as `1column` (the default)

Field	Type	Write	Description
`design`	string or null	yes	`luma_simple` (default), `luma_icons`, `hyva_simple` or `hyva_icons`
`is_active`	int	yes	`1` published, `0` hidden (the URL answers 404)
`topic_ids`	int[]	yes	The topics the page lists, in display order; replaces the list when present, an empty list clears it
`store_id`	int	yes	The store view the texts belong to; defaults to the store code of the URL
`created_at`, `updated_at`	string	read-only	Timestamps

Endpoints

```
GET    /rest/all/V1/faq/pages            list, with searchCriteria (title is filterable and sortable)
GET    /rest/all/V1/faq/pages/:pageId    one page
POST   /rest/all/V1/faq/pages            create
PUT    /rest/all/V1/faq/pages/:pageId    update; fields left out keep their value
DELETE /rest/all/V1/faq/pages/:pageId    delete
```

```
curl -X POST -H "Authorization: Bearer <token>" -H "Content-Type: application/json" \
  https://your-store.com/rest/all/V1/faq/pages \
  -d '{
    "entity": {
      "url_key": "faq",
      "title": "Frequently asked questions",
      "intro": "<p>Everything about ordering, shipping and returns.</p>",
      "is_active": 1,
      "topic_ids": [1, 2]
    }
  }'
```

The page is live on `https://your-store.com/faq` as soon as it is created with `is_active: 1`. An unknown topic id or a URL key that is already taken is refused with a `400`.

Errors

HTTP code	Meaning
200	Success; the body is the stored record
400	Refused: a missing question, an unknown SKU, product, category or topic. `message` and `parameters` say which
401	Missing or invalid token
404	No FAQ or topic with that id
500	Server error, see `var/log/exception.log`

A refused write stores nothing: the record, its text and its links are written in one transaction.

Need More Help?

Documentation:

- All Help Articles - Complete documentation overview

Support:

- Contact Support - Get help from our team

For a complete overview of features and configuration options, see the FAQ Manager extension on magmodules.eu

All articles for FAQ Manager

Installation

1	Installation using Composer (recommended)
2	Installation using the Adobe Marketplace
3	Install through FTP and SSH

Configuration

1	Configuration Guide
2	Quick Start Guide
3	FAQ Manager for Magento 2

Troubleshooting

1	Troubleshooting

Other

1	Import & Export
2	**REST API (current)**

Usage Guide

1	Managing FAQs
2	Managing Pages
3	Managing Topics

Background

1	Best Practices
2	CLI Commands

