The REST API

For CMS developers integrating apps and other systems, the REST interface that Exponential 6 ships. It covers the ezprestapi extension, the v1 and v2 routes, OAuth2 tokens and personal API keys, permissions, response…

Source file: `doc/eng-us/help/cms-developers/the-rest-api.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/cms-developers/the-rest-api.md

This page is for developers who connect a mobile app, a front end of another technology or another system to an Exponential site. Exponential 6 ships a REST interface for reading content and for creating and removing it; this page tells you what it covers today, how to authenticate, and where its limits are.

Contents

1. What serves the API
2. Addresses
3. The routes
4. Authentication
5. Permissions and answers
6. Examples
7. Settings
8. Limits to know
9. Routes of your own

What serves the API

The REST framework is in the kernel (`kernel/private/rest/`, `settings/rest.ini`). The content routes come from the extension `ezprestapi` (`se7enxweb/ezprestapi`, required by the root `composer.json`), whose provider `ezp7xRestApiProvider` serves the prefixes `ezp` and `ezpl`. The kernel's authentication provider serves `auth`, the OAuth2 token endpoint. The older `ezprestapiprovider` is deprecated (REST API v1 and v2).

Addresses

```
/api/ezp/v1/content/node/2
 |    |   |  |
 |    |   |  path
 |    |   version (v1 or v2)
 |    prefix (provider)
 ApiPrefix (rest.ini [System])
```

v1 answers as it always did, for existing clients; writes exist at v2 only.

The routes

Relative to `/api/ezp/<version>`:

Method	Path	v1	v2	What it does
GET	`/content/node/<id>`	yes	yes	one node
GET	`/content/node/<id>/list(/offset/<n>)(/limit/<n>)(/sort/<key>(/asc|desc))`	yes	yes	the children
GET	`/content/node/<id>/childrenCount`	yes	yes	the number of children
GET	`/content/node/<id>/fields`, `/field/<identifier>`	yes	yes	all fields, one field
GET	`/content/object/<id>` and its `fields`, `field/<identifier>`	yes	yes	the same by object ID
POST	`/content/node/create`	no	yes	create and publish below `parentNodeID`
POST	`/content/node/<id>`	no	yes	update: answers 501, not implemented yet
DELETE	`/content/node/delete/<id>`	no	yes	remove the node's object with its locations and subtree

Query parameters: `ResponseGroups=Metadata,Locations,Fields` chooses what an answer carries; `OutputFormat=xhtml` at v2 adds the rendered output of the node's view (The routes).

Authentication

`rest.ini` `[Authentication]` `RequireAuthentication=enabled` is the default: every content route needs a user.

Credential	For	How
Personal API key	a script of the user's own	the user makes it on **My account** > **API access** (needs the policy `apikey/create`); sent as `Authorization: Bearer expk_...`
OAuth2 token	an application that lets other users sign in and act for them	the application is registered under **Setup** > **oAuth admin**; authorization code flow; token endpoint `POST /api/auth/v1/oauth/token`

A request always runs as a user and never goes beyond that user's policies; a key is also limited to its scopes. Keys are shown once, can expire and can be revoked (Personal API keys, OAuth2 tokens).

Permissions and answers

Every call checks `content/read`, `content/create` or `content/remove` of its user, like the site does. Answers: `200`, `201` for a create, `400` for a missing or unknown field, `401` without valid credentials, `403` `access_denied` when the user may not, `404` for an unknown node. Build API access by giving a dedicated account a role with exactly the rights the client needs (Sections, roles and policies).

Examples

Keep the key in an environment variable, never in a file you commit:

Shell

```
read -s EXP_API_KEY && export EXP_API_KEY
curl -s -H "Authorization: Bearer $EXP_API_KEY" https://www.example.com/api/ezp/v1/content/node/2
curl -s -H "Authorization: Bearer $EXP_API_KEY" \
  "https://www.example.com/api/ezp/v1/content/node/2/list/offset/0/limit/20?ResponseGroups=Fields
"
curl -s -H "Authorization: Bearer $EXP_API_KEY" https://www.example.com/api/ezp/v2/content/node/c
reate \
  -d parentNodeID=<node id> -d classIdentifier=folder -d languageLocale=eng-US -d name='From a sc
ript'
```

A create answers `{"message":"Created","objectId":...,"nodeId":...}`. More in Examples with curl and the lesson Calling the API with a personal key.

Settings

Setting (`rest.ini`)	**Meaning**
`[System] ApiPrefix`	the path below which the REST interface answers (`/api`)
`[ApiProvider] ProviderClass[<prefix>]`	the provider class of each prefix
`[Authentication] RequireAuthentication`, `AuthenticationStyle`, `RequireHTTPS`	who may call, how, and only over HTTPS
`[ApiKeySettings] ApiKeys`	personal keys on or off (on by default)
`[OAuthSettings] TokenTTL`, `RefreshTokenTTL`	the life of an OAuth2 token and of its refresh token
`[OAuthSettings] AllowTokenInQuery`, `AllowTokenInBody`	where else than the `Authorization` header a token is read

Read them as the site sees them: `./console exp:ini list rest.ini/Authentication`. Turn `RequireHTTPS` on for any installation reachable by others.

Limits to know

- **No update route yet.** `POST /api/ezp/v2/content/node/<id>` answers 501. Update through the GraphQL mutation `updateContent` (The GraphQL API), look for an update service in the `expservices` catalog, or run a script on the server (Content from PHP).
- `listAtom` **answers 500** at v1 and v2: use the `list` route or the site's RSS export (Known issue: listAtom).
- **This is not the REST API of the Symfony platforms.** Clients written for `/api/ibexa/v2` need changes (Coming from eZ Platform or Ibexa).
- **GraphQL is a separate API.** Exponential 6.0.15 answers GraphQL at `/api/graphql`, off until it is switched on (The GraphQL API).

Routes of your own

A REST provider in your extension adds routes under a prefix of its own, with controllers that use the kernel API (A REST provider of your own). For administration functions and simple JSON calls there are also `expservices` and ezjscore server functions (Headless delivery).

Related pages

- Remote services and apps
- API and protocol standards
- Module 11: The REST API
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

Become an Exponential Partner

Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.

Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.