Headless delivery

For CMS developers who deliver Exponential content to apps and front ends of other technologies. It sets out what exists, namely the REST interface, expservices, ezjscore server functions, JSON views of your own and...

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

This page is for developers who want content managed in Exponential and shown elsewhere: a mobile app, a single page application, a kiosk, another website. Exponential 6 is a full CMS with its own templates, and it can serve content as data next to its pages, or instead of them. This page lists the ways that exist and how to choose.

Contents

1. The options
2. The REST interface
3. expservices
4. ezjscore server functions
5. A JSON view of your own
6. Feeds
7. Choosing
8. Patterns that work
9. What is not there

The options

Way	Address	Authentication	Best for
REST interface (`ezprestapi`)	`/api/ezp/v1/...`, `/api/ezp/v2/...`	personal API key, OAuth2 token	reading nodes, children and fields; creating and removing at v2
`expservices`	`/ezjscore/call/exp<domain>::<method>`	session cookie, or a personal token (`expt_...`)	the administration's functions: content, users, shop, media, layouts and more
ezjscore server functions of your own	`/ezjscore/call/<class>::<function>`	the site's session and policies	small JSON calls for your own front end
A module view that answers JSON	`/<your module>/<view>/...`	the site's session and policies	a custom endpoint with its own shape
Feeds	RSS export, and feed output from `expservices`	public, or as the site allows	lists of content for readers and aggregators

The REST interface

The documented content API: nodes, children with offset, limit and sort, fields, objects; writes at v2. At v2 `OutputFormat=xhtml` adds the rendered output of the node's view, so a client can show rich text as HTML drawn by the site's templates. Details, authentication and limits: The REST API.

expservices

The `expservices` extension, shipped in the kernel's `extension/` directory, offers the administration's functions as named HTTP services with a JSON envelope (`{"ok":true,"data":...,"meta":{...}}`), paged lists, a shell client and Python examples. A signed-in user creates a personal token with `expsession::tokenCreate`; it is shown once and sent as `Authorization: Bearer <token>` (Remote services and apps, Remote services: expservices). Discover what your installation offers with the catalog service rather than from memory:

Shell

```
curl -s -H "Authorization: Bearer $EXP_SERVICES_TOKEN" \
  'https://www.example.com/ezjscore/call/expservices::catalog::system?ContentType=json'
```

ezjscore server functions

For a front end of your own on the same site (a search-as-you-type box, a "load more" button), an ezjscore server function is a static PHP function registered in `ezjscore.ini`, called at `/ezjscore/call/...` and answered as JSON with the site's session and policies (Ajax with ezjscore server functions, A server function by hand).

A JSON view of your own

When a client needs a shape no existing interface gives, a module view in your extension can answer with JSON instead of a page (no pagelayout, a JSON body, the right content type) and declare a policy function for access (Answering with JSON or a file, Modules and views). Build the data with fetch functions or the kernel API, so permissions apply (Content from PHP).

Feeds

The kernel's RSS export publishes a subtree as a feed (**Setup** > **RSS**), and `expservices` can output RSS, Atom or JSON Feed for any subtree (RSS feeds). Use feeds for news lists that other systems pull on a schedule.

Choosing

Need	Choose
An app reads published content	REST v1 or v2 with a personal key of a read-only account
An app lets its users sign in and write as themselves	REST with OAuth2 tokens

Need	Choose
A tool automates editorial or administrative tasks	`expservices` with a personal token
Your own site's JavaScript needs a little data	an ezjscore server function
A partner needs one specific JSON shape	a module view of your own

Patterns that work

- **One account per client**, with a role that allows exactly what the client needs, and keys that expire.
- **Ask for fields, not pages.** Use `ResponseGroups` and `field/<identifier>` routes to keep answers small.
- **Keep rich text rendering on the server** when the client should look like the site: v2 `OutputFormat=xhtml`.
- **HTTPS only**: `rest.ini` `[Authentication]` `RequireHTTPS=enabled` on any reachable installation.
- **Cache on the client** and avoid polling every few seconds; content changes are published, not streamed.

What is not there

The REST interface has no working update route yet (it answers 501). Exponential 6.0.15 has a GraphQL endpoint of its own, `/api/graphql`, which reads and changes content and is off until it is switched on (The GraphQL API). Projects that need the Symfony platform's REST v2 and GraphQL use the Exponential Platform line, which can run next to this kernel through the legacy bridge (Coming from eZ Platform or Ibexa, Legacy bridge).

Related pages

- The REST API
- Application creators: start here
- API and protocol standards
- 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.