Errors and status codes

Every HTTP status the REST API of Exponential answers, with its JSON body and what an app should do - 200, 201, 400, 401 with and without a body, 403 access_denied and insufficient_scope, 404, 405 with Allow, 429 with...

Source file: `doc/eng-us/help/mobile-app-developers/rest-api/errors-and-status-codes.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/mobile-app-developers/rest-api/errors-and-status-codes.md

This page is for app developers who write the error handling of their API client. It lists every status the REST API answers, from the controller code and the Personal API keys guide, with what the app should do.

Contents

1. The table
2. The error body
3. What the app should do
4. A retry policy
5. Errors of the remote services

The table

Status	Body	Meaning
200	the content	a read or a remove succeeded
201	`{"message":"Created","objectId":...,"nodeId":...}`	a create was published
400	`{"error":"invalid_request","error_message":...}`	a write field is missing, or names a class or language that does not exist
401	empty, with `WWW-Authenticate: OAuth realm='Exponential REST'`	no credential reached the server
401	`{"error":"invalid_token"}`	an unknown, wrong or revoked token or key, a disabled owner, or a key sent in the URL
401	`{"error":"expired_token"}`	a personal API key past its end date
403	`{"error":"access_denied","error_message":...}`	the user may not read or write this content
403	`{"error":"insufficient_scope"}`	a personal API key lacks the scope of the route, or its owner lost the permission behind it
404	`{"error":"not_found",...}`	no such node or object
405	with an `Allow` header	no route takes this method for this path
429	`{"error":"rate_limited",...}` with `Retry-After`	a personal API key sent more requests this minute than `RateLimitPerMinute` (120 by default)

Status	Body	Meaning
500	{"error":"failed",...}	a create or remove failed on the server; also every listAtom call (a known issue)
501	{"error":"not_implemented",...}	the update route, after the permission check

The 401 without a body was observed on the demonstration server for a request with no Authorization header. The statuses with error and error_message come from errorResult() of ezp7xRestContentController.

The error body

Content errors have the shape:

JSON

```
{"error": "access_denied", "error_message": "You may not create 'article' below node 60 in eng-US
 (content/create)."}
```

error is a stable machine-readable reason (invalid_request, access_denied, not_found, failed, not_implemented); error_message is English text for logs and for experienced users. Branch on error and on the status, never on the text.

What the app should do

Status	App behavior
401 (OAuth token)	try one refresh with the refresh token; if that fails, send the user to sign in again
401 (other)	treat the credential as dead; do not retry with it
403	show "not allowed", do not retry; for insufficient_scope during development, check the key's scopes
404	the item was removed or unpublished: drop it from the app's cache and go back
400	show the message; it names the field
429	wait for Retry-After seconds
500, 502, 503, 504, timeouts	retry with backoff for reads; for writes, check first whether the write happened
501	a feature gap: use the route named on What the APIs do not do

A retry policy

- Retry only idempotent reads automatically: two or three times, with growing delays and some randomness.
- Never retry a create blindly: it may have succeeded before the connection dropped (Creating and removing content).
- Give up on 401 and 403; they do not change by retrying.
- Log the status, the route and the time; the server keeps a request id in its logs (the response header X-Exp-Request-Id was present on the demonstration server's ezjscore answers), which helps the site's administrator find the request (Finding an error by its

reference).

Errors of the remote services

The remote services answer differently: a call that reaches the service answers HTTP 200, and the result is in `content.ok`, with `content.error.code` (400, 401, 403, 404, 409, 422, 500) when it failed. A non-200 status or a body that is not JSON is a transport fault (Calling expservices from an app).

Related pages

- Personal API keys: troubleshooting
- The Authorization header and proxies
- Routes reference
- 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.