Independent Submission
Request for Comments: 0010
Category: Informational
ISSN: 2070-1721

M. Developer
RFC.EE Project

April 2026

API Reference for the RFC.EE Editor

Abstract

RFC.EE exposes a simple JSON API at /api.php. This document lists the available actions, their parameters, their return shapes, and the authentication rules that apply to each.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of an independent submission. It has been approved for publication by the RFC Editor, but is not a candidate for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc0010.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

Abstract 1
1. Introduction 4
2. Authentication 4
3. Read Actions 4
4. Write Actions 5
5. User-Management Actions 6
6. Error Responses 6
7. Security Considerations 6
8. References 6
9. Author's Address 6

1. Introduction

All API endpoints live at a single URL: /api.php. The endpoint is selected with a query-string parameter "action". Request bodies may be JSON or form-encoded; the response is always JSON with Content-Type: application/json.

2. Authentication

Read actions are public and require no authentication. Write actions require a valid session cookie obtained by logging in at /login. Calls to write actions without a session return HTTP 401 and an "Authentication required" JSON error.

User-management actions additionally require the admin role. All write actions rely on the session cookie's SameSite=Strict attribute for CSRF protection; callers outside the browser context must therefore use an authenticated session established in the same origin.

3. Read Actions

list_articles
: Returns all articles (published and unpublished) as an array of rows.

list_published
: Returns only published articles.

get_article (id=N)
: Returns one article with its full block list, blocks sorted by sort_order.

4. Write Actions

Write actions modify the database. They require authentication as an editor or admin.

create_article
: Creates a new article. Accepts rfc_number, title, short_title, author_name, author_org, category, status_stream, month_year, issn. Returns {id: N}.

update_article (id=N, ...)
: Updates an existing article with any of the above fields.

delete_article (id=N)
: Deletes an article and, by cascade, all of its blocks.

clone_article (id=N)
: Duplicates an article, assigning a fresh RFC number and copying every block.

toggle_publish (id=N)
: Flips the published flag on an article.

create_block (article_id=N, block_type=T, content=JSON)
: Appends a block to an article.

update_block (id=N, ...)
: Updates a block's type or content.

delete_block (id=N)
: Deletes a block.

duplicate_block (id=N)
: Inserts a copy of a block immediately after the original.

reorder_blocks (article_id=N, order=[id1,id2,...])
: Rewrites the sort_order column for the article's blocks.

seed_demo
: Inserts the RFC 9999 demonstration article.

seed_guides
: Inserts the ten meta-RFCs documented in this series (RFC 0001 through RFC 0010). Idempotent: existing documents are skipped.

5. User-Management Actions

User-management actions require the admin role.

list_users
: Returns all users.

create_user
: Creates a new user with a username, password, and role.

update_user (id=N, ...)
: Updates a user's username, password, or role.

delete_user (id=N)
: Deletes a user. Will refuse to delete the last admin or to let an admin delete themselves.

6. Error Responses

On error, the API returns a JSON object with an "error" key and an HTTP status code in the 4xx or 5xx range. Clients SHOULD check the HTTP status first and the response body second; the body is always valid JSON, even on errors.

7. Security Considerations

All actions use prepared statements for database access. Write actions rely on SameSite=Strict cookies for CSRF protection; operators who expose the API to non-browser callers MUST supply an explicit CSRF token. The API does not expose password hashes to any caller.

8. References

[RFC0005] M. Developer, "About the RFC.EE Editor", RFC 0005, February 2026

9. Author's Address

M. Developer
RFC.EE Project

Email: developer@example.org