Independent Submission
Request for Comments: 0005
Category: Informational
ISSN: 2070-1721

M. Developer
RFC.EE Project

February 2026

About the RFC.EE Editor

Abstract

RFC.EE is a small web application that lets authors compose documents in a block-based content model and render them as RFC-style documents. This document describes the application: its design goals, its architecture, and its limitations.

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/rfc0005.

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. Design Goals 4
3. Architecture Overview 5
4. Data Model 5
5. Technology Stack 5
6. Limitations 6
7. Security Considerations 6
8. References 6
9. Author's Address 6

1. Introduction

The RFC Series is a joy to read and a chore to write. Authors who want to produce an RFC-shaped document face a choice between the official toolchain, which is powerful but aimed at experienced contributors, and ad-hoc text editing, which provides no structure at all. RFC.EE occupies the middle ground: a browser-based editor that enforces structure while preserving the plain-text look of the output.

2. Design Goals

- Faithful output: documents rendered by the application SHOULD be indistinguishable in layout from genuine RFCs, within the limits of the block types supported.
- Zero build step: the application runs on a commodity PHP and MySQL installation, with no asset pipeline, no transpiler, and no external front-end framework.
- Block-first authoring: authors edit typed blocks rather than raw text, so structure is preserved across export formats.
- Three outputs from one source: every document is available as HTML, as RFC-style plain text, and as a print-ready PDF.

3. Architecture Overview

The application is intentionally small. The entire server-side codebase fits in ten PHP files totaling fewer than seven thousand lines. There is no framework; HTTP requests land in PHP files, which use prepared statements to talk to MySQL and emit HTML, JSON, or plain text.

A single renderer, render.php, walks the block list of an article and produces HTML that mirrors the 72-column plain-text layout. The HTML is authoritative: the plain-text and PDF formats are derived from it by extraction and rewriting, not re-rendered from scratch.

4. Data Model

The database has three tables.

articles
: One row per document. Stores the RFC number, title, short title, author name and organization, category, stream, month/year, ISSN, and a published flag.

blocks
: One row per block. Stores the article id (foreign key), the block type, a JSON content envelope, and a sort order.

users
: One row per editor account. Stores a username, a password hash, and a role (admin or editor).

5. Technology Stack

- PHP 8 with the mysqli driver.
- MySQL or MariaDB for storage.
- dompdf for PDF rendering.
- Inline HTML, CSS, and JavaScript -- no external front-end assets.

6. Limitations

The application makes no attempt to support every feature of the official RFC toolchain. It does not produce XML in the RFC v3 schema, it does not cross-link to external RFC registries, and it does not perform automated validation of references. It is intended as a low-friction editor for RFC-style documentation, not as a replacement for the IETF's tooling.

7. Security Considerations

The application uses PHP sessions for authentication, CSRF tokens on admin POST forms, and prepared statements for every database query. All user-supplied text is escaped before being written into HTML output. Operators SHOULD change the default admin password after installation. See [RFC0009] for the public-facing security model.

8. References

[RFC0009] M. Developer, "Security Model of the RFC.EE Editor", RFC 0009, March 2026

[RFC7322] Flanagan, H. and S. Ginoza, "RFC Style Guide", RFC 7322, September 2014

9. Author's Address

M. Developer
RFC.EE Project

Email: developer@example.org
URI: https://rfc.ee/