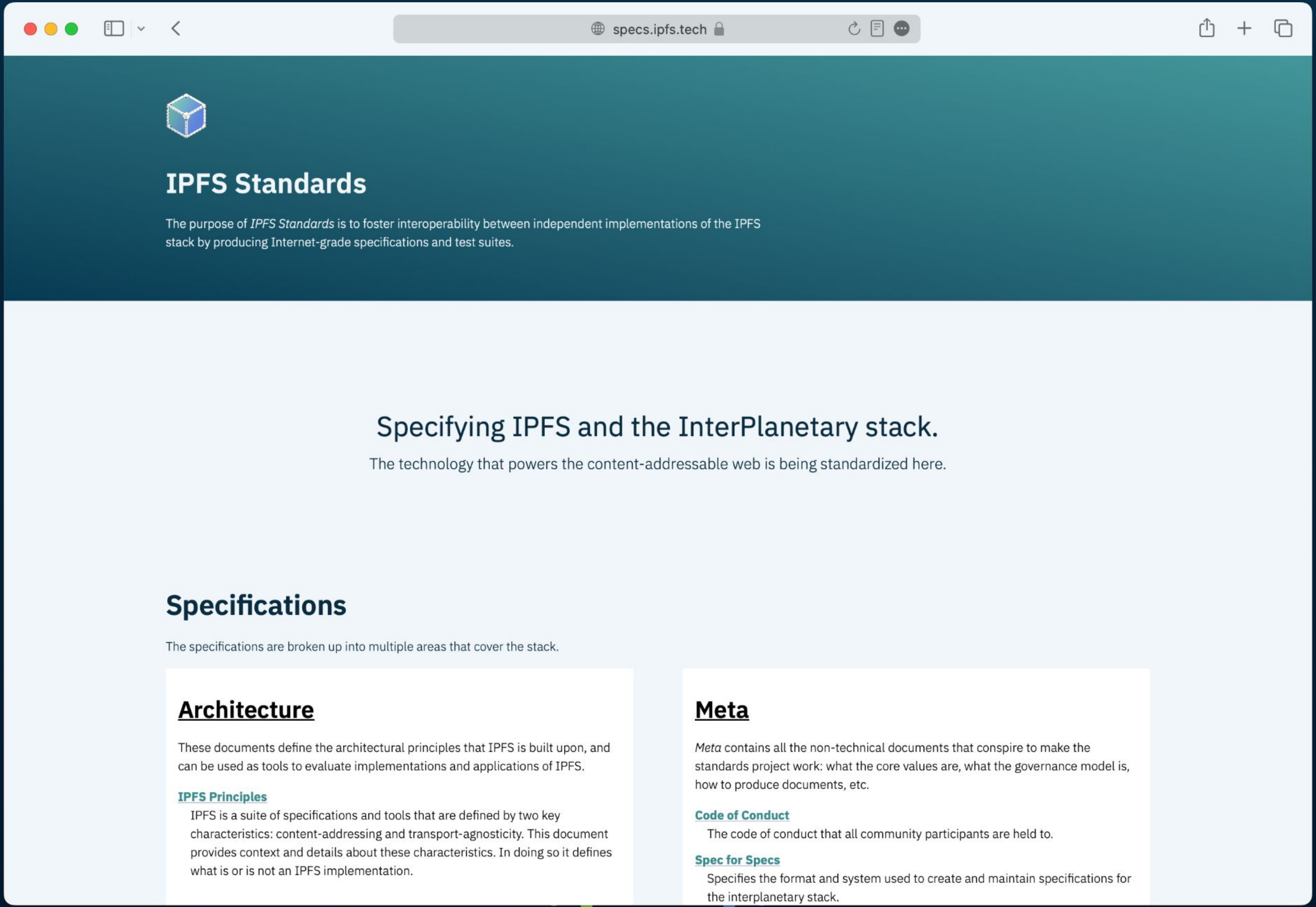
IPFS ÞING
Brussels, Belgium
2023

InterPlanetary Specifications

Henrique Dias (@hacdias)
Protocol Labs

✨ specs.ipfs.tech ✨



IPFS Standards

The purpose of *IPFS Standards* is to foster interoperability between independent implementations of the IPFS stack by producing Internet-grade specifications and test suites.

Specifying IPFS and the InterPlanetary stack.

The technology that powers the content-addressable web is being standardized here.

Specifications

The specifications are broken up into multiple areas that cover the stack.

Architecture

These documents define the architectural principles that IPFS is built upon, and can be used as tools to evaluate implementations and applications of IPFS.

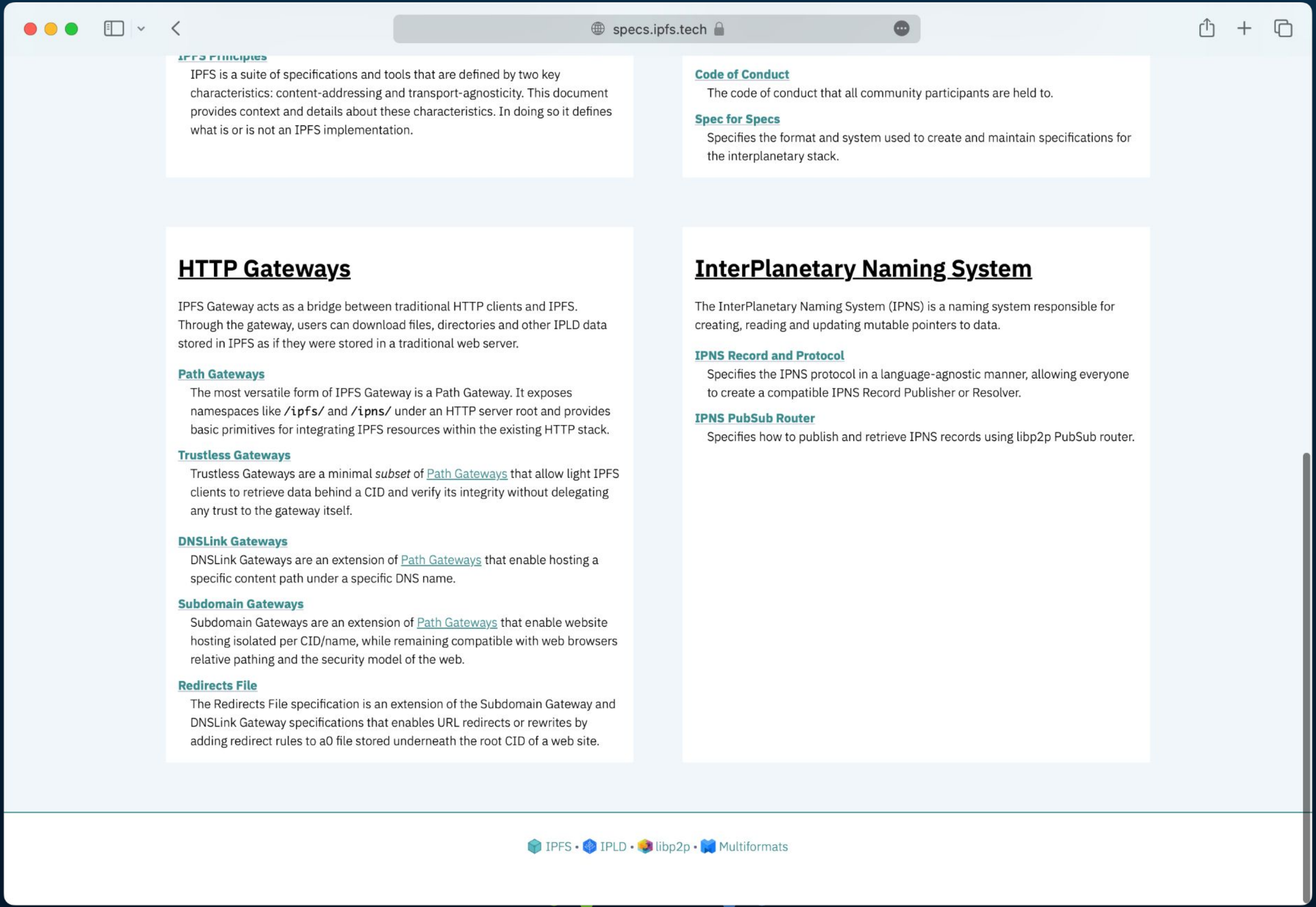
IPFS Principles
IPFS is a suite of specifications and tools that are defined by two key characteristics: content-addressing and transport-agnosticity. This document provides context and details about these characteristics. In doing so it defines what is or is not an IPFS implementation.

Meta

Meta contains all the non-technical documents that conspire to make the standards project work: what the core values are, what the governance model is, how to produce documents, etc.

Code of Conduct
The code of conduct that all community participants are held to.

Spec for Specs
Specifies the format and system used to create and maintain specifications for the interplanetary stack.



specs.ipfs.tech

IPFS Principles

IPFS is a suite of specifications and tools that are defined by two key characteristics: content-addressing and transport-agnosticity. This document provides context and details about these characteristics. In doing so it defines what is or is not an IPFS implementation.

Code of Conduct

The code of conduct that all community participants are held to.

Spec for Specs

Specifies the format and system used to create and maintain specifications for the interplanetary stack.

HTTP Gateways

IPFS Gateway acts as a bridge between traditional HTTP clients and IPFS. Through the gateway, users can download files, directories and other IPLD data stored in IPFS as if they were stored in a traditional web server.

Path Gateways

The most versatile form of IPFS Gateway is a Path Gateway. It exposes namespaces like `/ipfs/` and `/ipns/` under an HTTP server root and provides basic primitives for integrating IPFS resources within the existing HTTP stack.

Trustless Gateways

Trustless Gateways are a minimal *subset* of Path Gateways that allow light IPFS clients to retrieve data behind a CID and verify its integrity without delegating any trust to the gateway itself.

DNSLink Gateways

DNSLink Gateways are an extension of Path Gateways that enable hosting a specific content path under a specific DNS name.

Subdomain Gateways

Subdomain Gateways are an extension of Path Gateways that enable website hosting isolated per CID/name, while remaining compatible with web browsers relative pathing and the security model of the web.

Redirects File

The Redirects File specification is an extension of the Subdomain Gateway and DNSLink Gateway specifications that enables URL redirects or rewrites by adding redirect rules to a0 file stored underneath the root CID of a web site.

InterPlanetary Naming System

The InterPlanetary Naming System (IPNS) is a naming system responsible for creating, reading and updating mutable pointers to data.

IPNS Record and Protocol

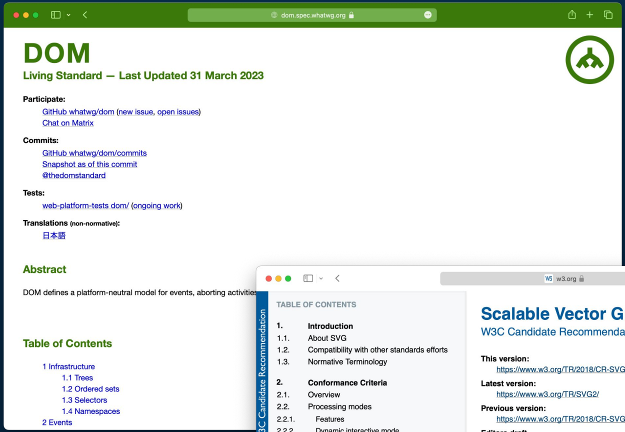
Specifies the IPNS protocol in a language-agnostic manner, allowing everyone to create a compatible IPNS Record Publisher or Resolver.

IPNS PubSub Router

Specifies how to publish and retrieve IPNS records using libp2p PubSub router.

Why standards? 🤔

- It should be possible to implement without looking at source code.
- Long-term, **documentation > code**.
- Maintain a clear track record of community consensus.
- Support better test writing (*testing is coming too*)!
- Be clear so as to create a bridge to other systems.



DOM
Living Standard — Last Updated 31 March 2023

Participate:
GitHub whatwg/dom (new issue, open issues)
Chat on Matrix

Commits:
GitHub whatwg/dom/commits
Snapshot as of this commit
@thedomstandard

Tests:
web-platform-tests dom/ (ongoing work)

Translations (non-normative):
日本語

Abstract

DOM defines a platform-neutral model for events, aborting activities

Table of Contents

1 Infrastructure
1.1 Trees
1.2 Ordered sets
1.3 Selectors
1.4 Namespaces
2 Events

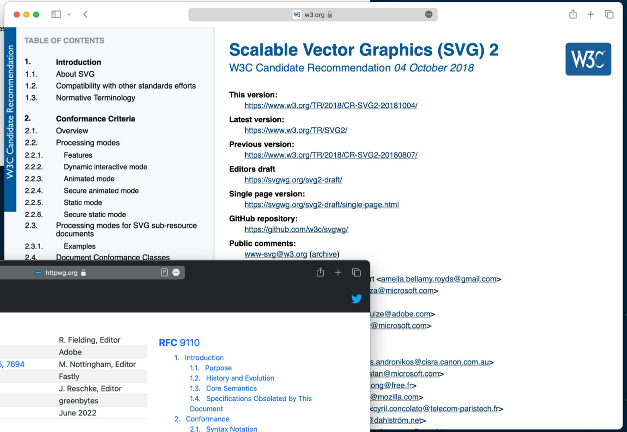


TABLE OF CONTENTS

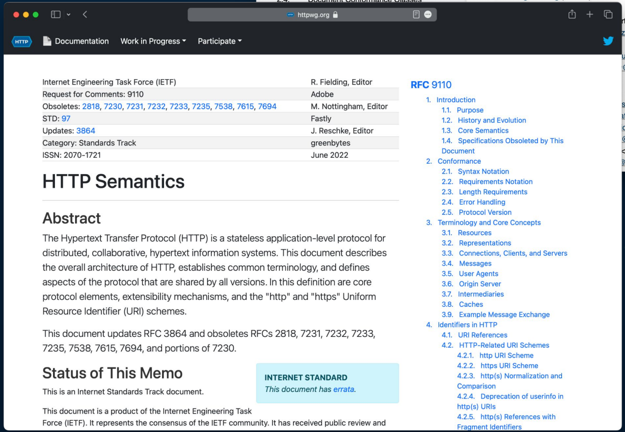
1. Introduction
1.1. About SVG
1.2. Compatibility with other standards efforts
1.3. Normative Terminology

2. Conformance Criteria
2.1. Overview
2.2. Processing modes
2.2.1. Features
2.2.2. Dynamic interactive mode
2.2.3. Animated mode
2.2.4. Secure animated mode
2.2.5. Static mode
2.2.6. Secure static mode
2.3. Processing modes for SVG sub-resource documents
2.3.1. Examples
2.4. Document Conformance Classes

Scalable Vector Graphics (SVG) 2
W3C Candidate Recommendation *04 October 2018*

This version:
https://www.w3.org/TR/2018/CR-SVG2-20181004/
Latest version:
https://www.w3.org/TR/SVG2/
Previous version:
https://www.w3.org/TR/2018/CR-SVG2-20180807/
Editors draft
https://svgwg.org/svg2-draft/
Single page version:
https://svgwg.org/svg2-draft/single-page.html
GitHub repository:
https://github.com/w3c/svgwg/
Public comments:
www-svg@w3.org (archive)

rt <amelia.bellamy.royds@gmail.com>
za@microsoft.com>
ulze@adobe.com>
@microsoft.com>
s.andronikos@cisra.canon.com.au>
atan@microsoft.com>
ong@free.fr>
@mozilla.com>
<cyril.concolato@telecom-paristech.fr>
@dahlström.net>



Internet Engineering Task Force (IETF)	R. Fielding, Editor
Request for Comments: 9110	Adobe
Obsoletes: 2818, 7230, 7231, 7232, 7233, 7235, 7538, 7615, 7694	M. Nottingham, Editor
STD: 97	Fastly
Updates: 3864	J. Reschke, Editor
Category: Standards Track	greenbytes
ISSN: 2070-1721	June 2022

HTTP Semantics

Abstract

The Hypertext Transfer Protocol (HTTP) is a stateless application-level protocol for distributed, collaborative, hypertext information systems. This document describes the overall architecture of HTTP, establishes common terminology, and defines aspects of the protocol that are shared by all versions. In this definition are core protocol elements, extensibility mechanisms, and the "http" and "https" Uniform Resource Identifier (URI) schemes.

This document updates RFC 3864 and obsoletes RFCs 2818, 7231, 7232, 7233, 7235, 7538, 7615, 7694, and portions of 7230.

Status of This Memo

This is an Internet Standards Track document.

INTERNET STANDARD
This document has errata.

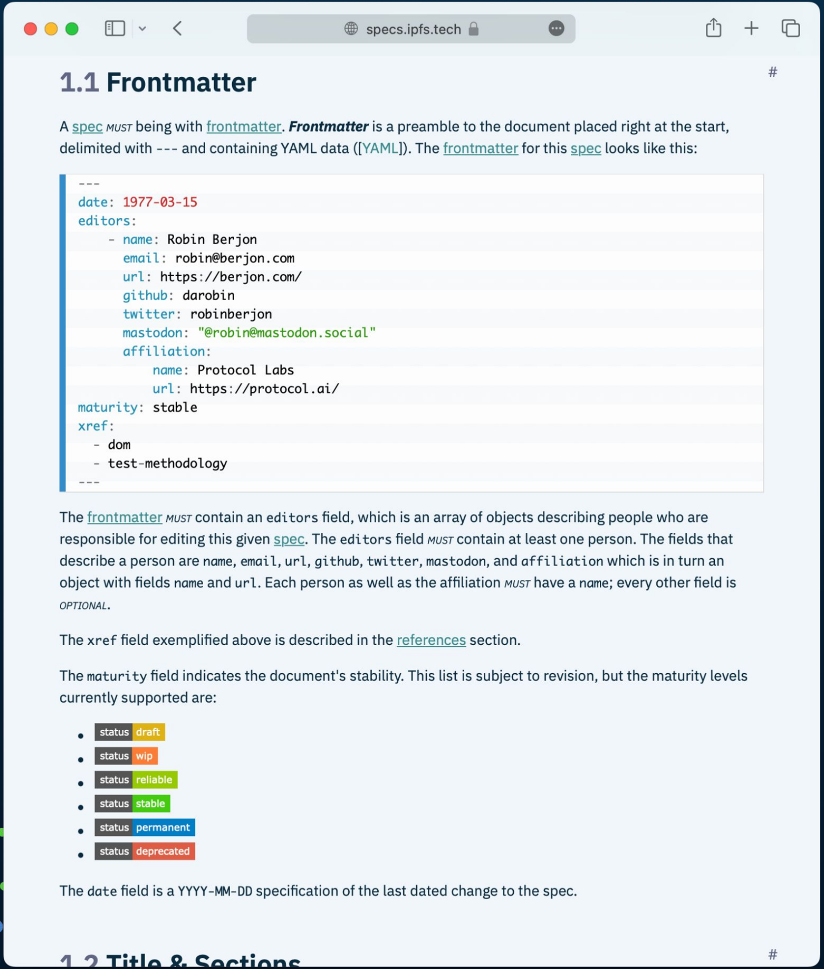
This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and

RFC 9110

1. Introduction
1.1. Purpose
1.2. History and Evolution
1.3. Core Semantics
1.4. Specifications Obsoleted by This Document
2. Conformance
2.1. Syntax Notation
2.2. Requirements Notation
2.3. Length Requirements
2.4. Error Handling
2.5. Protocol Version
3. Terminology and Core Concepts
3.1. Resources
3.2. Representations
3.3. Connections, Clients, and Servers
3.4. Messages
3.5. User Agents
3.6. Origin Server
3.7. Intermediaries
3.8. Caches
3.9. Example Message Exchange
4. Identifiers in HTTP
4.1. URI References
4.2. HTTP-Related URI Schemes
4.2.1. http URI Scheme
4.2.2. https URI Scheme
4.2.3. http(s) Normalization and Comparison
4.2.4. Deprecation of userinfo in http(s) URIs
4.2.5. http(s) References with Fragment Identifiers

What are the goals? 🎖️

- Looking ***goooood***.
- Give them visibility. A bunch of markdown files in a repository was not enough.
- Keep it simple: it's still all Markdown.
- Stable links & references management.
- Support typical spec features (definitions, references, RFC 2119 keywords, metadata, etc.)
- Progressively add more tooling on top to make people's lives easier.



specs.ipfs.tech

1.1 Frontmatter

A spec MUST being with frontmatter. ***Frontmatter*** is a preamble to the document placed right at the start, delimited with `---` and containing YAML data ([YAML]). The frontmatter for this spec looks like this:

```
---
date: 1977-03-15
editors:
    - name: Robin Berjon
      email: robin@berjon.com
      url: https://berjon.com/
      github: darobin
      twitter: robinberjon
      mastodon: "@robin@mastodon.social"
      affiliation:
          name: Protocol Labs
          url: https://protocol.ai/
maturity: stable
xref:
  - dom
  - test-methodology
---
```

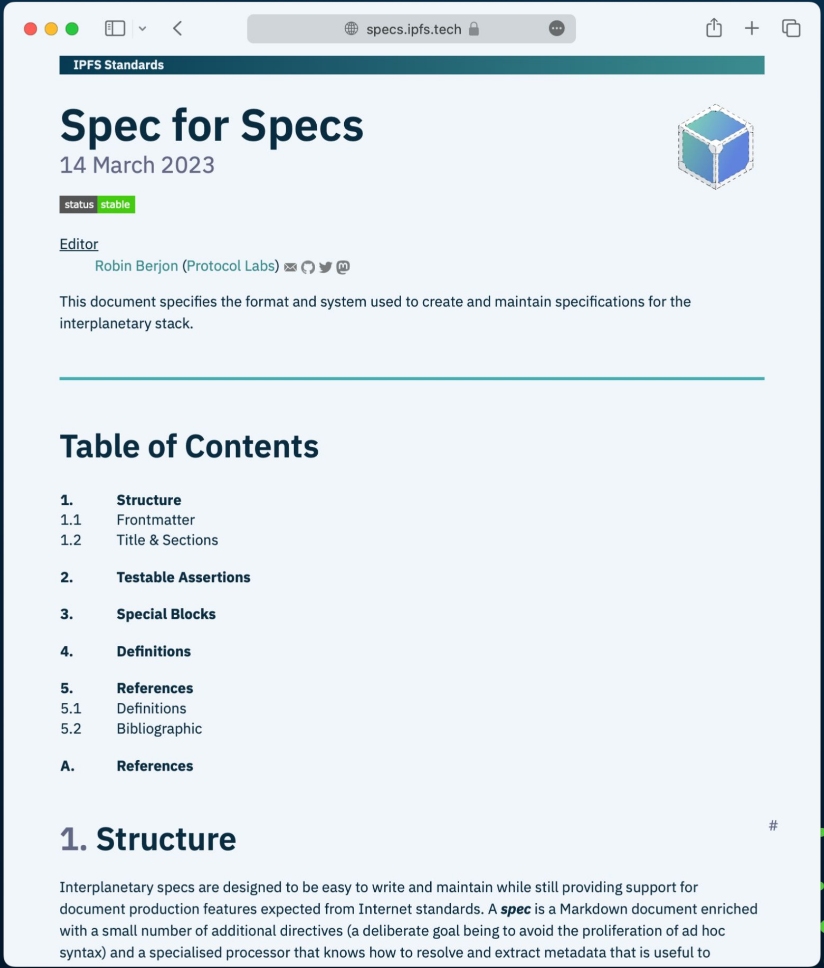
The frontmatter MUST contain an `editors` field, which is an array of objects describing people who are responsible for editing this given spec. The `editors` field MUST contain at least one person. The fields that describe a person are `name`, `email`, `url`, `github`, `twitter`, `mastodon`, and `affiliation` which is in turn an object with fields `name` and `url`. Each person as well as the affiliation MUST have a `name`; every other field is OPTIONAL.

The `xref` field exemplified above is described in the references section.

The `maturity` field indicates the document's stability. This list is subject to revision, but the maturity levels currently supported are:

- status draft
- status wip
- status reliable
- status stable
- status permanent
- status deprecated

The `date` field is a `YYYY-MM-DD` specification of the last dated change to the spec.


specs.ipfs.tech

IPFS Standards

Spec for Specs

14 March 2023

status stable

Editor

Robin Berjon (Protocol Labs)

This document specifies the format and system used to create and maintain specifications for the interplanetary stack.

Table of Contents

1. Structure

Interplanetary specs are designed to be easy to write and maintain while still providing support for document production features expected from Internet standards. A ***spec*** is a Markdown document enriched with a small number of additional directives (a deliberate goal being to avoid the proliferation of ad hoc syntax) and a specialised processor that knows how to resolve and extract metadata that is useful to

How can you participate? 🫂

- **specs**: ipfs/specs
- **generator**: ipfs/spec-generator

- Read the "**Spec for Specs**" (mostly Markdown with some lightweight additions & conventions).
- Or just copy a spec and figure it out…
- Run `make watch` and you're in business!

- *That's it!*

🙏 Þank you! 🙏