

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 30 September 2026

klfr
LunarEclipse
29 March 2026

The Well Known Button Information Specification
draft-filmroellchen-lunar-well-known-button-00

Abstract

This document specifies the well-known URI `/.well-known/button.json`, which describes a web site's "buttons". Buttons are usually 88x31 pixel images representing the web site with text, logos, artwork, and animations.

`/.well-known/button.json` files facilitate sharing buttons between web site owners and alleviate issues commonly encountered when doing so. By utilizing a standardized, machine-readable format, automated tools can also utilize the provided information.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-filmroellchen-lunar-well-known-button/>.

Source for this draft and an issue tracker can be found at <https://codeberg.org/LunarEclipse/well-known-button>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 30 September 2026.

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. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Language	4
1.2. Definitions	4
2. button.json Files	5
2.1. Button Objects	6
2.1.1. Required Properties	6
2.1.2. Optional Properties	7
2.1.3. Accessibility Properties	10
3. Examples	12
4. Implementation Considerations	14
5. IANA Considerations	14
6. Legal Considerations	14
7. Privacy Considerations	15
8. Security Considerations	15
8.1. imageRendering CSS Injection	16
8.2. Denial-of-Service Concerns Due To Hotlinking	16
9. References	16
9.1. Normative References	16
9.2. Informative References	18
Appendix A. JSON Schema for button.json	19
Appendix B. Migrating from earlier versions of this specification	21
B.1. Migrating from draft 2024-05	21
B.2. Migrating from draft 2024-06	21
Acknowledgements	21
Contributors	22
Authors' Addresses	22

1. Introduction

"Buttons", in the context of this specification, refer to a certain type of common image on the Web. Buttons are graphics, usually 88 pixels wide and 31 pixels high, featuring text, logos, artwork, as well as animations. The purpose of a button is to represent the web site or its owner(s)/author(s), often using minimal information to do so, owing to the low resolution available. For this representational purpose, buttons are usually intended to be included on other web sites, most commonly in a footer or other dedicated section. This way, web site owners may show their affiliation with, or appreciation of other's web sites.

Buttons originate from the early days of the Web as a form of banners, originally representing the site or technology used to host a particular home page. The 88x31 format, also named "Micro Button", originated with the free hosting provider GeoCities.com and their mandatory advertising banner. For a detailed historical account, see [Tekeye]. Buttons have seen a renewed surge in popularity in the 2020s with certain subcultures.

The goal of this specification is to provide a clearly defined standard for web site authors who wish to share web site buttons with other web site authors and end users, and lay out a format for a common endpoint that can be used to fetch and/or embed the buttons. To this end, this specification introduces a new well-known [RFC8615] URI utilizing a standard JSON format to specify one or more buttons.

The intended use cases are:

- * to share buttons in a standardized way
- * to avoid issues with crediting the original button author(s)
- * to avoid issues with button authors' consent to including buttons on other pages
- * to facilitate automation around inclusion of buttons, including but not limited to:
 - button updates with caching
 - helping web site owners discover available buttons
 - auto-selecting the most suitable button variant for theming and accessibility

The specification aims to be easy to implement for web sites using shared hosting providers, which are unable to change HTTP response headers and cannot host extension-less files.

Accessibility is a primary concern of this specification. Images and fast animations are notoriously inaccessible to users with vision impairment, light-sensitivity, and other disabilities. This specification attempts to combat that by requiring image descriptions, and providing a standard mechanism for selecting between multiple versions of a button for accessibility purposes.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Definitions

A server is any web server providing a `/.well-known/button.json` file.

A client is any program accessing the `/.well-known/button.json` of a server. Clients include other web servers, automated programs, as well as programs manually invoked on behalf of a user.

An animated image (as opposed to a static image) is an image which has multiple frames of nonzero and non-infinite length, where at least one of the frames is visually distinct from the others.

Examples:

- * A GIF image with three frames, all containing different text in front of the same background, is animated.
- * A WEBP image with two very distinct frames is animated.
- * An AVIF image with several hundred frames and over two minutes in runtime, featuring a "gloss" effect that very slowly moves across the image, is animated.
- * A GIF image abusing a frame time of zero to show more than 256 colors at once is animated, as almost every modern program restricts the frame time to a minimum of 10ms.
- * An AVIF image with five frames, all of which are identical, is not animated.

- * A GIF image with ten frames, all of which only contain imperceptible color differences in a few pixels, is not animated.
- * An APNG image with two thousand identical frames is not animated.
- * A GIF image with a single frame is not animated.

2. button.json Files

The main element of this specification is the `/.well-known/button.json` file. This is a well-known [RFC8615] URI registered with IANA; see Section 5.

A server supporting this specification MUST serve a conforming file on the top-level path `/.well-known/button.json`. The file SHOULD be available via HTTPS [RFC9110], and MAY be served via HTTP. The file SHOULD be served using the Content-Type header set to `application/json; charset=utf-8`. The file MUST be valid JSON [RFC8259], and MUST be encoded using UTF-8 [RFC3629]. HTTP caching headers and content encoding MAY be used.

A normative description for requirements of various `/.well-known/button.json` entries follows. A normative JSON Schema [JSON-Schema] which defines the basic syntax rules is provided in Appendix A. and can be used to validate `/.well-known/button.json` files. Not all files conforming to the schema are necessarily correct implementations of this specifications.

The top-level `$schema` property MUST be present and link to the canonical location of the schema version used by the document. The server MAY provide a `/.well-known/button.schema.json` file containing a copy of the used schema.

The `$schema` value for this RFC is `https://codeberg.org/LunarEclipse/well-known-button/raw/branch/main/drafts/draft-filmroellchen-lunar-well-known-button-00.schema.json`.

Note to the RFC editor: During draft stage, the `$schema` value is `https://codeberg.org/LunarEclipse/well-known-button/raw/branch/main/drafts/draft-filmroellchen-lunar-well-known-button-XX.schema.json`, where XX is the draft version. After publication, the `$schema` property should be changed to `https://codeberg.org/LunarEclipse/well-known-button/raw/branch/main/rfcXXXX.schema.json`, where XXXX is the RFC number of this specification. This change must also apply to all examples and the JSON Schema below.

The top-level buttons property MUST be present and contain a list of objects, each of which specifies a single button.

The top-level default property is OPTIONAL and specifies which button is considered the default one when multiple are present. It MUST contain an id matching that of one of the buttons present in the file.

2.1. Button Objects

The button objects are the objects in the buttons list. Each one specifies an independent single button. The validity of button objects SHOULD be determined independently by clients; one invalid button object provided by the server does not affect the validity of another button object in the same file.

2.1.1. Required Properties

Clients MUST reject button objects which do not provide all of the id, uri, and alt properties.

2.1.1.1. id

The id property serves to identify the button uniquely across revisions. This property is REQUIRED.

Revisions of the same button MUST share the same id property, and consequently MUST NOT appear in the same button.json file. Multiple different buttons MUST NOT have the same id value.

The server chooses the id property arbitrarily. Clients MUST NOT make any assumptions about the content of the id property, and MUST treat it as an arbitrary [Unicode] string. Clients SHALL NOT assume IDs are unique across web sites.

2.1.1.2. uri

The uri property is the canonical location of the button, from which any client can access the button's image file. This property is REQUIRED. The uri property MUST contain a valid URI [RFC3986] using the https protocol name, and must subsequently be accessible using the HTTPS [RFC9110] Internet protocol.

Referenced image files SHOULD have a height of 31 pixels and width of 88 pixels. Images MAY be larger in size, but MUST maintain the aspect ratio, and SHOULD be legible when scaled down to a width of 88 pixels and height of 31 pixels.

Image files may be either static or animated; see Section 1.2 for a detailed definition including edge-case examples. The following image file formats are RECOMMENDED in order of preference:

- * AVIF [AVIF]: animated and static images
- * WEBP [RFC9649]: animated and static images
- * PNG [PNG]: static images
- * GIF [GIF]: animated and static images

Button images MUST NOT be lossily compressed. Of the recommended image formats, AVIF and WEBP are capable of lossy compression and care must be taken to avoid it.

2.1.1.3. alt

The alt property MUST be provided, describing the content of the button for vision-impaired users.

2.1.2. Optional Properties

2.1.2.1. link

Buttons displayed on a webpage SHOULD be contained in a hyperlink element such as <a>, linking back to the button author's web page. The link property facilitates this. It is an OPTIONAL property containing a URI [RFC3986]. The button's surrounding hyperlink element MUST point to the URI specified in the link property of the button, if available. Otherwise, it SHOULD point to the base path from where the /.well-known/button.json was fetched. For example, if a client fetches <https://example.org/.well-known/button.json>, and one of the contained buttons does not feature a link property, the client should embed this button in a hyperlink pointing to <https://example.org/>.

2.1.2.2. hotlink

The OPTIONAL hotlink boolean flag specifies whether the button image should be hotlinked, i.e. included directly from the source uri instead of copying to and then hosting from the client's web server. Many authors prefer their images not to be hotlinked, as it creates additional load on the server. This is the default (false). However, some authors prefer their images to be hotlinked for a variety of reasons, such as serving dynamically generated buttons. These authors can set hotlink to true for the relevant buttons, in order to indicate to clients to hotlink the button.

Clients SHOULD NOT hotlink button images without the hotlink property set to true. Servers MAY impose restrictions on clients which they suspect to be in violation of this requirement.

2.1.2.3. sha256

The sha256 property is OPTIONAL. It facilitates caching and validation of the image file. This property MUST contain the SHA256 [RFC6234] content hash of the image file as a hexadecimal number (64 characters). Clients SHOULD assume that a button image has not changed if the sha256 property has not changed since the last time the client downloaded the button image, even if other properties have been changed, including the uri property.

2.1.2.4. caption

The OPTIONAL caption property provides a caption or title for the button. The contents of this property are distinct from the alt property, as they are intended to be shown to every user. RECOMMENDED methods of displaying the caption text include the HTML title attribute [html-title], as well as conventional figure caption text.

2.1.2.5. imageRendering

The OPTIONAL imageRendering property provides guidance on how to render scaled images. In several circumstances, such as rendering buttons at increased size, or when using displays with high DPI, images need to be scaled to more than 88x31 physical device pixels. However, while some buttons are designed for nearest-neighbor scaling (preserving a pixelated look), some are designed for smooth scaling (as is commonplace for most graphics). Using this property, the author can specify which of the two scaling methods the button is intended for.

In general, any of the possible image-rendering [image-rendering] CSS properties MAY be used. In this case, web sites MAY use the imageRendering property verbatim as the CSS image-rendering property for the button in this case. (See Section 8.1 for the security implications of this approach.)

The following values are RECOMMENDED for use in imageRendering. Other values SHOULD be interpreted as per the image-rendering [image-rendering] CSS specification.

- * auto: Client decides which rendering method to use; equivalent to omitting this property.

- * `pixelated`: Nearest neighbor scaling to an integer multiple of the image dimensions.
- * `smooth`: Smooth scaling, such as bilinear filtering.

At time of publication, the CSS value `smooth` is not yet honored by all major browsers. However, the default image rendering method is almost always smooth scaling, therefore `auto` MAY be used in CSS instead.

If this property is not specified, clients MAY choose any rendering method, such as one in line with a page's style. Even if this property is specified, clients MAY choose other rendering methods for a variety of reasons. However, client authors and users MUST be aware of the subsequent degradation of authorial intent, since the image author might have designed the button around a very specific rendering method.

2.1.2.6. `license`

The OPTIONAL `license` property contains an SPDX license expression [SPDX-license-expressions] identifying the license that applies to the button image file and this button object describing it. See Legal considerations (Section 6) for a discussion of button licensing, including recommended license choices.

2.1.2.7. `licenseText`

The OPTIONAL `licenseText` property contains arbitrary text augmenting the license data in the `license` property. This property should be considered as legally binding as the `license` property. It is NOT RECOMMENDED to specify `licenseText` without `license`; several SPDX license expressions are available which reference nonstandard licenses, such as `DocumentRef` and `LicenseRef`.

RECOMMENDED use cases for `licenseText` are:

- * Adding authorship information, especially when the image author is not the web site author
- * Adding a copyright year
- * Adding a completely custom license in combination with a `LicenseRef` entry in the `license` property

2.1.3. Accessibility Properties

By using the `groupId` property, a group of interchangeable and equivalent buttons can be defined. The client's specific choice of button among these can then be tailored to the user or the application, improving accessibility.

2.1.3.1. `groupId`

The OPTIONAL `groupId` property defines a group of button versions. Multiple buttons using the same `groupId` entry belong to one group. A client SHALL assume those buttons are interchangeable, and they MAY choose one of them depending on the user's requirements.

The server chooses the `groupId` property arbitrarily. Clients MUST NOT make any assumptions about the content of the `id` property, and MUST treat it as an arbitrary [Unicode] string. Clients SHALL NOT assume IDs are unique across web sites.

The following properties, `colorScheme`, `animations`, and `contrast`, allow providing different versions of a button to fit accessibility and/or styling needs. They SHOULD be present if the server shares different versions of a button with the same `groupId`. Within the same `groupId`, multiple buttons SHOULD NOT have identical values for these properties.

The following properties MAY also be provided if the `groupId` property is absent. In this case, the properties provide accessibility guidance without the possibility of client choice.

2.1.3.2. `colorScheme`

The OPTIONAL `colorScheme` property specifies what color scheme the button version uses. It is to be interpreted as per [prefers-color-scheme] CSS media query.

It can take on the following values:

- * `other`: the default, equivalent to not specifying this property, clients SHOULD NOT make any assumptions about the color scheme
- * `light`: light theme, typically dark text on a light background
- * `dark`: dark theme, typically light text on a dark background

2.1.3.3. animations

The OPTIONAL `animations` property SHOULD be provided if the button is animated, even if the button is not in a group. It specifies the amount and severity of the animations present in the button. Clients MAY consider all animated buttons without this property to have the value `high`.

It can take on the following values:

- * `none`: the image is not animated at all
- * `minimal`: the image contains only very subtle animations
- * `high`: the image contains animations that are permanently active, distracting, or could trigger medical conditions such as light-sensitive epilepsy

Authors SHOULD use the `high` value for animated buttons if there is any doubt about the severity of the animations.

2.1.3.4. contrast

The OPTIONAL `contrast` property MUST have one of the following values:

- * `standard`: the default, equivalent to not specifying this property.
- * `more`: a higher-contrast version of the button, whose text is STRONGLY RECOMMENDED to follow the WCAG 2.2 AA or AAA guidelines for color contrast in normal text [WCAG-contrast], i.e. 4.5:1 or 7:1.
- * `less`: a lower-contrast version of the button, whose text color contrast is below 3:1. This means that it does not follow the WCAG 2.2 color contrast guidelines at all, not even for large text or graphics.

If two buttons with substantially distinct color contrasts would fall into the same contrast category according to the above definitions, distinct contrast values SHOULD be used by applying the following rules:

- * If both buttons are at `less` contrast, the higher-contrast one SHOULD be instead marked as `standard`.
- * If both buttons are at `more` contrast, the lower-contrast one SHOULD be instead marked as `standard`.

- * If both buttons are at standard contrast, the lower-contrast one SHOULD be instead marked as less.

Since none of these rules are strictly required, implementations MAY deviate in order to utilize all three contrast values.

3. Examples

A typical example is as follows:

```
{
  "$schema": "https://codeberg.org/LunarEclipse/well-known-button/raw/branch/main/drafts/draft-filmroellchen-lunar-well-known-button-00.schema.json",
  "default": "my.web site",
  "buttons": [
    {
      "id": "my.web site",
      "uri": "https://my.web site.example.org/my.web site.png",
      "alt": "Button to my web site!",
      "link": "https://my.web site.example.org",
      "sha256": "66a421c7e726e9de99eeb88c57b93b49278d64b2a4602a6f90f7d64baee154cf",
      "hotlink": true
    }
  ]
}
```

A minimal example:

```
{
  "$schema": "https://codeberg.org/LunarEclipse/well-known-button/raw/branch/main/drafts/draft-filmroellchen-lunar-well-known-button-00.schema.json",
  "buttons": [
    {
      "id": "some button id",
      "uri": "https://example.com/res/my-button.gif",
      "alt": "button saying example.com"
    }
  ]
}
```

And an exhaustive example:

```
{
  "$schema": "https://codeberg.org/LunarEclipse/well-known-button/raw/branch/main/drafts/draft-filmroellchen-lunar-well-known-button-00.schema.json",
  "default": "8b556a30-c5d9-4117-88a5-b779a3f2f567",
  "buttons": [
    {
      "id": "8b556a30-c5d9-4117-88a5-b779a3f2f567",
      "groupId": "mainbutton",
      "uri": "https://web site.example.com/res/my-button.png",

```

```

    "alt": "button saying example.com with black text on a white background",
    "caption": "Example web site",
    "sha256": "e35a78bcb7f9b9cc0c3929d1763b96b6013071b0f9950886a20d2bcc0e943612",
    "link": "https://web site.example.com/",
    "colorScheme": "light",
    "animations": "none",
    "contrast": "more",
    "license": "CC-BY-SA-4.0",
    "licenseText": "Copyright 2024 by the Example Author",
    "imageRendering": "pixelated",
    "hotlink": false
  },
  {
    "id": "64dbf02d-44e0-4aa9-ad45-c4959eadd3db",
    "groupId": "mainbutton",
    "uri": "https://web site.example.com/res/my-button-dark.png",
    "alt": "button saying example.com with white text on a black background",
    "caption": "Example web site",
    "sha256": "4598d79c6c1877aa9121c8b0845fe4e8f031684fabb09caa10357dfe5d295986",
    "link": "https://web site.example.com/",
    "colorScheme": "dark",
    "animations": "none",
    "contrast": "more",
    "license": "CC-BY-SA-4.0",
    "licenseText": "Copyright 2024 by the Example Author",
    "imageRendering": "auto",
    "hotlink": false
  },
  {
    "id": "57ad38e5-94ad-4b64-a6bc-583f41b7c3b5",
    "groupId": "mainbutton",
    "uri": "https://web site.example.com/res/my-button-rainbow.gif",
    "alt": "button saying example.com with black text on an animated rainbow backgro
und",
    "caption": "Example web site",
    "sha256": "2fecb1bbd1fdb1f8fa634632f9726f9bce7f653fe94cde59b616f4032b70a758",
    "link": "https://web site.example.com/",
    "colorScheme": "other",
    "animations": "high",
    "contrast": "standard",
    "license": "LicenseRef-Commercial",
    "licenseText": "Copyright 2024 by the Example Author, all rights reserved. You c
an include this button on your page but nothing else.",
    "imageRendering": "smooth",
    "hotlink": true
  },
  {
    "id": "ee5cc4b3-b88b-4b1c-aelf-fb9a3de063c9",
    "uri": "https://web site.example.com/res/blog-button.gif",
    "alt": "example.com blog button with some stars gently twinkling in the backgro
und",

```

```
    "caption": "Example web site",
    "sha256": "b2fe6da951362a7e3909390c5634fe4804cb845edddccad8dcea5819122f94be0",
    "link": "https://web site.example.com/blog/",
    "animations": "minimal",
    "license": "CC-BY-SA-4.0",
    "licenseText": "Copyright 2024 by the Example Author",
    "hotlink": true
  }
]
```

4. Implementation Considerations

This section has not yet been written, and the authors appreciate any input for it.

5. IANA Considerations

IANA will register the well-known URI /.well-known/button.json in the well-known URIs registry [IANA-well-known] in conformance with the requirements for this registration in [RFC8615]. The following information is provided to facilitate the registration:

- * URI suffix: button.json
- * Change controller: IETF
- * Specification document(s): This RFC
- * Status: permanent

In case a JSON Schema registry is set up with IANA in the future, the button.json Schema (Appendix A), as defined in this specification and any of its updates and errata, shall be registered by IANA with this JSON Schema registry.

6. Legal Considerations

Buttons are usually copyrightable artworks. As such, unauthorized copying and inclusion of a button is a legal offense in most jurisdictions. This specification does not supersede the legal frameworks for copyright, licenses and permissions, but it attempts to aid all involved parties in avoiding legal issues.

Publication of a /.well-known/button.json file, as per this specification, implies a permission to download (copy) the linked-to buttons, and include them on other pages, provided that a button's link (as per this specification) is persisted. These are the minimal

permissions required to make the data in `/.well-known/button.json` useful to third parties. This does not waive copyright and is not comparable to stronger copyleft licenses. It is also not as legally unambiguous as the license properties, so the use of these properties is strongly RECOMMENDED.

The `license` and `licenseText` properties allow the button author to provide a copyright license. In case the author is not provided in either of those properties, it can be assumed to be the same as the author of the web site's content, or the web site's owner.

It is RECOMMENDED that clients use SPDX tooling and human-in-the-loop verification of both `license` and `licenseText` to verify that the client's use of the button is allowed by the license.

The following licenses are RECOMMENDED for use in `/.well-known/button.json`:

- * Any of the current Creative Commons licenses [CC], including CC BY, CC BY-SA, and CC BY-ND. The no-derivative licenses including the latter are closest to the default permissions implied with publication of a `/.well-known/button.json` file.
- * A public domain license, such as [CC0] or [Unlicense].

A full list of standard licenses tracked by SPDX is available at <https://spdx.org/licenses/> (<https://spdx.org/licenses/>).

7. Privacy Considerations

Specifying the location of a button image in `/.well-known/button.json` exposes it to automated scrapers, including malicious ones. If a web server operator wishes to decrease the visibility of the buttons, `robots.txt` [RFC9309] can be used to discourage scrapers from accessing `/.well-known/button.json`, but since some scrapers do not respect `robots.txt` properly, the web server may additionally need to block certain user agents and IP addresses from accessing `/.well-known/button.json`.

8. Security Considerations

8.1. imageRendering CSS Injection

The imageRendering property may be set to an arbitrary string by a malicious web site operator. Clients which use the property without additional checks as part of the button's CSS properties (whether in stylesheets or inline style in HTML), are subsequently vulnerable to CSS or HTML injection attacks. Therefore, clients MUST NOT use this property without validating it, and reject any unknown values.

8.2. Denial-of-Service Concerns Due To Hotlinking

Hotlinking buttons imposes an additional load on the server, as it has to serve the button to every visitor of a client's web page, not just its own page(s). Therefore, usage of the "hotlink": true setting should be carefully evaluated, as even well-intentioned clients may cause increased load on the server if their pages receive many views.

Malicious or careless clients may ignore the hotlink attribute and always hotlink the button image. This scenario is no different from ordinary HTTP Denial-of-Service attacks and should be addressed similarly.

9. References

9.1. Normative References

- [RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <<https://www.rfc-editor.org/rfc/rfc8615>>.
- [RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <<https://www.rfc-editor.org/rfc/rfc6234>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/rfc/rfc3986>>.
- [RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <<https://www.rfc-editor.org/rfc/rfc9110>>.

- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/rfc/rfc8259>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<https://www.rfc-editor.org/rfc/rfc3629>>.
- [JSON-Schema] Wright, A., Andrews, H., Hutton, B., Dennis, G., and JSON Schema, "JSON Schema, Internet-Draft 01", 10 June 2022, <<https://datatracker.ietf.org/doc/html/draft-bhutton-json-schema-01>>.
- [image-rendering] World Wide Web Consortium, "CSS Images Module Level 3 Editor's Draft: 5.2: The 'image-rendering' property", n.d., <<https://drafts.csswg.org/css-images/#the-image-rendering>>.
- [SPDX-license-expressions] The Linux Foundation, "SPDX Specification 3.0.1 Annex B: SPDX license expressions", February 2024, <<https://spdx.github.io/spdx-spec/v3.0.1/annexes/spdx-license-expressions/>>.
- [Unicode] Unicode, Inc., "The Unicode Standard, Version 17.0.0", n.d., <<https://www.unicode.org/versions/Unicode17.0.0/>>.
- [prefers-color-scheme] World Wide Web Consortium, "CSS Media Queries Level 5 Working Draft: 12.5: The 'prefers-color-scheme' feature", n.d., <<https://www.w3.org/TR/mediaqueries-5/#descdef-media-prefers-color-scheme>>.
- [WCAG-contrast] World Wide Web Consortium, "Web Content Accessibility Guidelines 2.2: 1.4.3 Contrast (Minimum)", December 2024, <<https://www.w3.org/TR/WCAG22/#contrast-minimum>>.
- [IANA-well-known] Internet Assigned Numbers Authority, "Well-Known URIs Registry", n.d., <<https://www.iana.org/assignments/well-known-uris/well-known-uris.xhtml>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

9.2. Informative References

- [RFC9309] Koster, M., Illyes, G., Zeller, H., and L. Sassman, "Robots Exclusion Protocol", RFC 9309, DOI 10.17487/RFC9309, September 2022, <<https://www.rfc-editor.org/rfc/rfc9309>>.
- [Tekeye] Fowler, D., "Remembering Powered By and 88×31", 27 March 2012, <https://www.tekeye.uk/computer_history/powered-by>.
- [AVIF] Guyon, Y., Barnes, L., Chang, W.-T., Concolato, C., Kerr, P., Klements, A., and Alliance for Open Media, "AV1 Image File Format (AVIF)", 2025, <<https://aomediacodec.github.io/av1-avif/v1.2.0.html>>.
- [RFC9649] Zern, J., Massimino, P., and J. Alakuijala, "WebP Image Format", RFC 9649, DOI 10.17487/RFC9649, November 2024, <<https://www.rfc-editor.org/rfc/rfc9649>>.
- [PNG] Blume, C., Lemieux, P.-A., Lilley, C., Needham, C., Rosenthol, L., Seeger, C. A., Thompson, S., Truta, C., and World Wide Web Consortium, "Portable Network Graphics Specification (Third Edition)", 2025, <<https://www.w3.org/TR/png-3/>>.
- [GIF] CompuServe Incorporated, "Graphics Interchange Format", 1987, <<https://www.w3.org/Graphics/GIF/spec-gif87.txt>>.
- [html-title] WHATWG, "HTML: 3.2.6.1: The 'title' attribute", n.d., <<https://html.spec.whatwg.org/multipage/dom.html#the-title-attribute>>.
- [CC] Creative Commons, "Creative Commons Licenses", n.d., <<https://creativecommons.org/share-your-work/cclicenses/>>.
- [CC0] Creative Commons, "CC0 1.0 Universal", n.d., <<https://creativecommons.org/publicdomain/zero/1.0/>>.

[Unlicense]

Bendiken, A., "The Unlicense", 2010,
<<https://unlicense.org/>>.

Appendix A. JSON Schema for button.json

The following is the JSON Schema [JSON-Schema] file for /.well-known/button.json. This JSON Schema is **normative**; all /.well-known/button.json implementations MUST conform to this schema.

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://codeberg.org/LunarEclipse/well-known-button/raw/branch/main/drafts/draft-filmroellchen-lunar-well-known-button-00.schema.json",
  "title": ".well-known 88x31 Button JSON Schema",
  "type": "object",
  "properties": {
    "$schema": {
      "title": "The URI of the schema used by this document.",
      "type": "string",
      "format": "uri"
    },
    "default": {
      "title": "The id of the button to use as the default.",
      "type": "string"
    },
    "buttons": {
      "type": "array",
      "items": {
        "title": "An 88x31 button available from the web site",
        "type": "object",
        "properties": {
          "id": {
            "title": "The ID of the button",
            "type": "string"
          },
          "uri": {
            "title": "The canonical uri of the image file",
            "type": "string",
            "format": "uri",
            "pattern": "^https://"
          },
          "alt": {
            "title": "Alt text for the image",
            "type": "string"
          },
          "caption": {
            "title": "Image caption or title",
            "type": "string"
          }
        }
      }
    }
  }
}
```

```

    },
    "link": {
      "title": "Where the image should link to if used on another web site",
      "type": "string",
      "format": "uri"
    },
    "hotlink": {
      "title": "Whether image should be hotlinked",
      "type": "boolean",
      "default": false
    },
    "sha256": {
      "title": "SHA256 hash of the current version of the image file",
      "type": "string",
      "format": "regex",
      "pattern": "^[A-Za-f0-9]{64}$"
    },
    "license": {
      "title": "SPDX license expression specifying the license for the image file",
      "type": "string"
    },
    "licenseText": {
      "title": "Additional license information to augment the license property",
      "type": "string"
    },
    "groupId": {
      "title": "A group of interchangeable versions of a button with different accessibility properties",
      "type": "string"
    },
    "colorScheme": {
      "title": "What color scheme does this button version match",
      "type": "string",
      "enum": ["light", "dark", "other"],
      "default": "other"
    },
    "animations": {
      "title": "Whether this button version is animated, and to what degree",
      "type": "string",
      "enum": ["none", "minimal", "high"]
    },
    "contrast": {
      "title": "The contrast level of this button version",
      "type": "string",
      "enum": ["standard", "more", "less"],
      "default": "standard"
    },
    "imageRendering": {
      "title": "Image rendering algorithm",

```

```

        "description": "Which scaling algorithm to use for the image if it is not
displayed at native resolution, recommended: 'smooth', 'auto' or 'pixelated'",
        "type": "string",
        "default": "auto"
    },
    },
    "required": ["id", "uri", "alt"]
}
},
"required": ["$schema", "buttons"]
}

```

Appendix B. Migrating from earlier versions of this specification

B.1. Migrating from draft 2024-05

At a basic level, this can be done easily with jq
(<https://github.com/jqlang/jq>):

```
jq '{ "$schema": "https://codeberg.org/LunarEclipse/well-known-button/raw/branch/main/
drafts/draft-filmroellchen-lunar-well-known-button-00.schema.json", "buttons": . }' $old_
filename > $new_filename
```

You will no longer need special logic for your web server, instead
the new path for the file is `./well-known/button.json`.

You may add a copy of the schema version matching your file in
`./well-known/button.schema.json`.

You might want to add a default property, especially if you provide
multiple buttons.

You should consider adding the extra accessibility properties where
the default values aren't sufficient (especially animations).

B.2. Migrating from draft 2024-06

The current schema is backwards-compatible with the 2024-06 one,
adding only optional properties.

You should consider adding the new properties where applicable,
especially license.

Acknowledgements

This specification would not have been possible without the many
excellent ideas from various beings, most of which are listed under
Appendix "Contributors". The authors thank Arch Retriever in
particular for the original `./well-known/button.json` idea.

Contributors

Arch Retriever

URI: <https://arch.dog>

Arch came up with the idea and created the original version of this specification.

Reina

Reina contributed the idea of an alt text property.

Sugar

Sugar contributed ideas on how to make this specification more accessible to people using shared hosting services.

Seirdy

Seirdy contributed the idea of multiple versions of a button with different accessibility properties.

Soblow Opale Xaselgio

France

Email: soblowdragon@gmail.com

URI: <https://xaselgio.net>

Soblow contributed examples and correction to the 2024 second draft, as well as the hotlink attribute.

Natty

URI: <https://natty.sh>

Natty contributed the idea of a caption property distinct from the alt property.

Authors' Addresses

kleines Filmroellchen

Germany

Email: kleines+ietf@filmroellchen.eu

URI: <https://filmroellchen.eu>

Lunar Eclipse

Poland

Email: luna@lunareclipse.zone

URI: <https://lunareclipse.zone>