

webpub-manifest



Radium Web Publication Manifest

The Radium Web Publication Manifest is a JSON-based document meant to represent and distribute publications over HTTPS.

It is the primary exchange format used in the [Radium Architecture](#) and serves as the main building block for [OPDS 2.0](#).

Example

```
{
  "@context": "https://readium.org/webpub-manifest/context.jsonld",

  "metadata": {
    "@type": "http://schema.org/Book",
    "title": "Moby-Dick",
    "author": "Herman Melville",
    "identifier": "urn:isbn:978031600000X",
    "language": "en",
    "modified": "2015-09-29T17:00:00Z"
  },

  "links": [
    {"rel": "self", "href": "https://example.com/manifest.json", "type": "application/json"},
    {"rel": "alternate", "href": "https://example.com/publication.epub", "type": "application/epub+zip"},
    {"rel": "search", "href": "https://example.com/search{?query}", "type": "text/html"}
  ],

  "readingOrder": [
    {"href": "https://example.com/c001.html", "type": "text/html", "title": "Chapter 1"},
    {"href": "https://example.com/c002.html", "type": "text/html", "title": "Chapter 2"}
  ],

  "resources": [
    {"rel": "cover", "href": "https://example.com/cover.jpg", "type": "image/jpeg"},
    {"rel": "style", "href": "https://example.com/style.css", "type": "text/css"}
  ]
}
```

```
    {"href": "https://example.com/whale.jpg", "type": "image/jpeg"},  
    {"href": "https://example.com/boat.svg", "type": "image/svg+xml"},  
    {"href": "https://example.com/notes.html", "type": "text/html"}  
  ]  
}
```

1. Introduction

1.1. Goals

While the Web is the largest collection of interlinked documents ever created, it lacks a mechanism for expressing how a collection of resources, when grouped together can represent a publication.

Publication formats such as EPUB or CBZ/CBR group these documents together using a container format, making them easier to archive or transmit as a whole. But they also break an important promise of the Web: the resources of a publication are not available through HTTP to any client that would like to access them.

W3C has recently provided a definition for a [Web Publication](#):

A **Web Publication (WP)** is a collection of one or more constituent resources, organized together in a uniquely identifiable grouping, and presented using standard Open Web Platform technologies.

It also provides a definition for a manifest in the context of a Web Publication:

[...] **manifest** refers to an abstract means to contain information necessary to the proper management, rendering, and so on, of a publication. This is opposed to metadata that contains information on the content of the publication like author, publication date, and so on. The precise format of how such a manifest is stored is not considered in this document.

The Radium Web Publication Manifest is an attempt to standardize a JSON based manifest format that follows both definitions.

To facilitate the interoperability between EPUB and Web Publications, this document also defines a number of extension points to fully support EPUB specific features.

1.2. Terminology

Collection

A grouping of some variable number of data items. In the context of this specification, a collection is defined as a grouping of metadata, links and sub-collections.

Full Collection

A collection that contains at least two or more data items among metadata, links and sub-collections.

Compact Collection

A collection that contains one or more links, but doesn't contain any metadata or sub-collections.

Manifest

A manifest is a full collection that represents structured information about a publication.

1.3. Abstract Model

The Radium Web Publication Manifest is based on a hypermedia model inspired by [Atom](#), [HAL](#), [Siren](#) and [Collection+JSON](#) among others.

Every Radium Web Publication Manifest is a [collection](#) that **MUST** contain:

- [metadata](#)
- [links](#)
- a [reading order](#)

2. Syntax

2.1. Sub-Collections

This specification defines two collection roles that are the building blocks of any manifest:

Role	Definition	Compact Collection?	Required?
<code>readingOrder</code>	Identifies a list of resources in reading order for the publication.	Yes	Yes
<code>resources</code>	Identifies resources that are necessary for rendering the publication.	Yes	No

Both collections are full collections, which means that they contain one or more [Link Objects](#).

All additional collection roles are defined in the [Collection Roles registry](#).

Extensions that are not registered in the registry **MUST** use a URI for their role.

A manifest **MUST** contain a `readingOrder` sub-collection.

Other resources that are required to render the publication **SHOULD** be listed in `resources`.

All resources listed in `readingOrder` and `resources` **MUST** indicate their media type using `type` .

Example 1: Reading order and list of resources

```
"readingOrder": [
  {"href": "/chapter1", "type": "text/html"},
  {"href": "/chapter2", "type": "text/html"}
],

"resources": [
  {"href": "/style.css", "type": "text/css"},
  {"href": "/image1.jpg", "type": "image/jpeg"}
]
```

2.2. Metadata

JSON-LD provides an easy and standard way to extend existing JSON document: through the addition of a context, we can associate keys in a document to Linked Data elements from various vocabularies.

The Web Publication Manifest relies on JSON-LD to provide a context for the `metadata` object of the manifest.

While JSON-LD is very flexible and allows the context to be defined in-line (local context) or referenced (URI), the Web Publication Manifest restricts context definition strictly to references (URIs) at the top-level of the document.

The Web Publication Manifest defines an initial registry of well-known context documents, which currently includes the following references:

Name	URI	Description	Required?
Default Context	https://readium.org/webpub-manifest/context.jsonld	Default context definition used in every Web Publication Manifest.	Yes

Context documents are all defined and listed in the [Context Documents registry](#).

The Radium Web Publication Manifest has a single requirement in terms of metadata: all publications **MUST** include a [title](#).

In addition all publications **SHOULD** include a `@type` key to describe the nature of the publication.

Example 2: Minimal metadata

```
"metadata": {
  "@type": "http://schema.org/Book",
  "title": "Test Publication"
}
```

2.3. Links

Links are expressed using the `links` key that contains one or more [Link Objects](#).

A manifest **MUST** contain at least one link using the `self` relationship where `href` is an absolute URI to the canonical location of the manifest.

Example 3: Link to the canonical location of a manifest

```
"links": [
  {
    "rel": "self",
    "href": "http://example.org/manifest.json",
    "type": "application/webpub+json"
  }
]
```

A manifest **MAY** also contain other links, such as a `alternate` link to an EPUB 3.1 version of the publication for example.

Link relations that are currently used in this specification and its extensions are listed in the [Link Relations registry](#).

2.4. The Link Object

The Link Object is a core component of the Radium Web Publication Manifest JSON syntax.

It represents a link to a resource along with a set of metadata associated with that resource.

This specification defines the following keys for this JSON object:

Key	Definition	Format	Required?
<code>href</code>	URI or URI template of the linked resource	URI or URI template	Yes
<code>templated</code>	Indicates that <code>href</code> is a URI template	Boolean, defaults to <code>false</code>	Only when <code>href</code> is a URI template

Key	Definition	Format	Required?
type	Media type of the linked resource	MIME Media Type	No
title	Title of the linked resource	String	No
rel	Relation between the resource and its containing collection	One or more Link Relations	No
properties	Properties associated to the linked resource	Properties Object	No
height	Height of the linked resource in pixels	Integer	No
width	Width of the linked resource in pixels	Integer	No
duration	Duration of the linked resource in seconds	Float	No
bitrate	Bit rate of the linked resource in kilobits per second	Float	No
children	Resources that are children of the linked resource, in the context of a given collection role	One or more Link Objects	No

3. Resources in the Reading Order

The `readingOrder` of a manifest **MAY** contain references to any text, image, video or audio resource that can be opened in a Web browser.

4. Media Type

This specification introduces a dedicated media type value to identify the Radium Web Publication Manifest: `application/webpub+json`.

All HTTP responses for the manifest **MUST** indicate this media type in their headers.

4. Discovering a Manifest

The Radium Web Publication Manifest **MAY** be referenced by resources included in its `readingOrder` or resources using a link.

Such links **MUST** include:

- `application/webpub+json` as the media type of the manifest
- `manifest` as the relation of the link

Example 4: Link in HTML to a manifest

```
<link href="manifest.json" rel="manifest" type="application/webpub+json">
```

Example 5: Link in HTTP headers to a manifest

```
Link: <http://example.org/manifest.json>; rel="manifest";
      type="application/webpub+json"
```

5. Table of Contents

A Radium Web Publication Manifest **MAY** contain a reference to a table of contents.

In order to represent a table of contents in the manifest, this specification introduces an additional collection role:

Role	Definition	Compact Collection?	Required?
<code>toc</code>	Identifies the collection that contains a table of contents.	Yes	No

Example 6: Partial TOC for an audiobook

```
"toc": [
  {
    "href": "track1.mp3#t=71",
    "title": "Part 1 - This World",
    "children": [
      {
        "href": "track1.mp3#t=80",
        "title": "Section 1 - Of the Nature of Flatland"
      },
      {
        "href": "track1.mp3#t=415",
        "title": "Section 2 - Of the Climate and Houses in Flatland"
      },
      {
        "href": "track1.mp3#t=789",
        "title": "Section 3 - Concerning the Inhabitants of Flatland"
      }
    ]
  }
]
```

As a fallback mechanism, a Radium Web Publication Manifest **MAY** identify an HTML or XHTML resource in `readingOrder` or `resources` as a table of contents using the `contents` link relation.

Example 7: Reference to an HTML resource containing a TOC

```
{
  "rel": "contents",
  "href": "contents.html",
  "type": "text/html"
}
```

A User Agent **MAY** also rely on the `title` key included in each Link Object of the `readingOrder` to extract a minimal table of contents.

The EPUB extension also defines **additional collection roles** for embedding navigation directly in the manifest.

6. Cover

A Radium Web Publication Manifest **MAY** contain a reference to a cover.

Link Objects in `readingOrder`, `resources` or `links` can be identified as such using the `cover` link relation.

All Link Objects containing the `cover` link relation **MUST** reference an image directly. They **SHOULD** include a `height` and `width` to facilitate how they are processed by User Agents.

This specification recommends using one of the following media types: `image/jpeg`, `image/png`, `image/gif` or `image/svg+xml`.

Example 8: Reference to a cover

```
{
  "rel": "cover",
  "href": "cover.jpg",
  "type": "image/jpeg",
  "height": 600,
  "width": 400
}
```

7. Extensibility

The manifest provides multiple extension points:

- additional collection roles using the [registry of roles](#) or URIs
- additional metadata using schema.org, terms from the [registry of context documents](#) or URIs (for individual terms)
- additional link relations from the [IANA link registry](#) or URIs
- additional properties using the [registry of properties](#)

In addition to these extension points, this specification defines an [extension registry](#) as well, to document specific profiles of the manifest.

The initial registry, contains the following extensions:

Name	Description
EPUB Extension	Additional metadata and collection roles for representing EPUB publications.
Audiobook Profile	Defines a dedicated profile for audiobooks.
Visual Narrative Profile	Defines a dedicated profile for visual narratives (comics, manga and bandes dessinées).

8. Package

The Radium Web Publication Manifest is primarily meant to be distributed unpackaged on the Web.

That said, a Radium Web Publication Manifest **MAY** be included in an EPUB.

If a Radium Web Publication Manifest is included in an EPUB, the following restrictions apply:

- the manifest document **MUST** be named `manifest.json` and **MUST** appear at the top level of the container
- the OPF of the primary rendition **MUST** include a link to the manifest where the link relation is set to `alternate`

Example 9: Reference to a manifest in an OPF

```
<link rel="alternate"
      href="manifest.json"
      media-type="application/webpub+json" />
```

In addition to the EPUB format, a Radium Web Publication **MAY** also be distributed as a separate package where:

- its media type **MUST** be `application/webpub+zip`

- its file extension **MUST** be `.webpub`
- the package itself **MUST** be a ZIP archive and follow the restrictions expressed in [ISO/IEC 21320-1:2015](#)
- the manifest **MUST** be named `manifest.json` and **MUST** appear at the top level of the package
- all resources in `readingOrder`, `resources` and `links` **MUST** be referenced relatively to the manifest
- a publication where any resource is encrypted using a DRM **MUST** use a different media type and file extension

Appendix A. JSON Schema

A JSON Schema is available under version control at <https://github.com/readium/webpub-manifest/tree/master/schema>

For the purpose of validating a Radium Web Publication Manifest, use the following JSON Schema resource: <https://readium.org/webpub-manifest/schema/publication.schema.json>

This site is open source. [Improve this page.](#)