Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 12, 2016

M. Kelly
Stateless
May 11, 2016

# JSON Hypertext Application Language
# draft-kelly-json-hal-08

Abstract

This document proposes a media type for representing resources and their relations with hyperlinks.

Status of This Memo

Copyright Notice

Table of Contents

## 1. Introduction

There is an emergence of non-HTML HTTP applications ("Web APIs") which use hyperlinks to direct clients around their resources.

The JSON Hypertext Application Language (HAL) is a standard which establishes conventions for expressing hypermedia controls, such as links, with JSON [RFC4627].

HAL is a generic media type with which Web APIs can be developed and exposed as series of links.  Clients of these APIs can select links by their link relation type and traverse them in order to progress through the application.

HAL's conventions result in a uniform interface for serving and consuming hypermedia, enabling the creation of general-purpose libraries that can be re-used on any API utilising HAL.

The primary design goals of HAL are generality and simplicity.  HAL can be applied to many different domains, and imposes the minimal amount of structure necessary to cover the key requirements of a hypermedia Web API.

## 2. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 3. HAL Documents

A HAL Document uses the format described in [RFC4627] and has the media type "application/hal+json".

Its root object MUST be a Resource Object.

For example:

```
GET /orders/523 HTTP/1.1
Host: example.org
Accept: application/hal+json

HTTP/1.1 200 OK
Content-Type: application/hal+json

{
  "_links": {
    "self": { "href": "/orders/523" },
    "warehouse": { "href": "/warehouse/56" },
    "invoice": { "href": "/invoices/873" }
  },
  "currency": "USD",
  "status": "shipped",
  "total": 10.20
}
```

Here, we have a HAL document representing an order resource with the URI "/orders/523". It has "warehouse" and "invoice" links, and its own state in the form of "currency", "status", and "total" properties.

## 4. Resource Objects

A Resource Object represents a resource.

It has two reserved properties:

(1) "_links": contains links to other resources.

(2) "_embedded": contains embedded resources.

All other properties MUST be valid JSON, and represent the current state of the resource.

### 4.1. Reserved Properties

#### 4.1.1. _links

The reserved "_links" property is OPTIONAL.

It is an object whose property names are link relation types (as defined by [RFC5988]) and values are either a Link Object or an array of Link Objects. The subject resource of these links is the Resource Object of which the containing "_links" object is a property.

#### 4.1.2. _embedded

The reserved "_embedded" property is OPTIONAL

It is an object whose property names are link relation types (as defined by [RFC5988]) and values are either a Resource Object or an array of Resource Objects.

Embedded Resources MAY be a full, partial, or inconsistent version of the representation served from the target URI.

## 5. Link Objects

A Link Object represents a hyperlink from the containing resource to a URI. It has the following properties:

## 5.1. href

The "href" property is REQUIRED.

Its value is either a URI [RFC3986] or a URI Template [RFC6570].

If the value is a URI Template then the Link Object SHOULD have a "templated" attribute whose value is true.

## 5.2. templated

The "templated" property is OPTIONAL.

Its value is boolean and SHOULD be true when the Link Object's "href" property is a URI Template.

Its value SHOULD be considered false if it is undefined or any other value than true.

## 5.3. type

The "type" property is OPTIONAL.

Its value is a string used as a hint to indicate the media type expected when dereferencing the target resource.

## 5.4. deprecation

The "deprecation" property is OPTIONAL.

Its presence indicates that the link is to be deprecated (i.e. removed) at a future date.  Its value is a URL that SHOULD provide further information about the deprecation.

A client SHOULD provide some notification (for example, by logging a warning message) whenever it traverses over a link that has this property.  The notification SHOULD include the deprecation property's value so that a client manitainer can easily find information about the deprecation.

## 5.5. name

The "name" property is OPTIONAL.

Its value MAY be used as a secondary key for selecting Link Objects which share the same relation type.

## 5.6. profile

The "profile" property is OPTIONAL.

Its value is a string which is a URI that hints about the profile (as defined by [I-D.wilde-profile-link]) of the target resource.

## 5.7. title

The "title" property is OPTIONAL.

Its value is a string and is intended for labelling the link with a human-readable identifier (as defined by [RFC5988]).

## 5.8. hreflang

The "hreflang" property is OPTIONAL.

Its value is a string and is intended for indicating the language of the target resource (as defined by [RFC5988]).

## 6. Example Document

The following is an example document representing a list of orders

```
GET /orders HTTP/1.1
Host: example.org
Accept: application/hal+json

HTTP/1.1 200 OK
Content-Type: application/hal+json

{
  "_links": {
    "self": { "href": "/orders" },
    "next": { "href": "/orders?page=2" },
    "find": { "href": "/orders{?id}", "templated": true }
  },
  "_embedded": {
    "orders": [{
        "_links": {
          "self": { "href": "/orders/123" },
          "basket": { "href": "/baskets/98712" },
          "customer": { "href": "/customers/7809" }
        },
        "total": 30.00,
        "currency": "USD",
        "status": "shipped",
      },{
        "_links": {
          "self": { "href": "/orders/124" },
          "basket": { "href": "/baskets/97213" },
          "customer": { "href": "/customers/12369" }
        },
        "total": 20.00,
        "currency": "USD",
        "status": "processing"
    }]
  },
  "currentlyProcessing": 14,
  "shippedToday": 20
}
```

Here, the order list document provides a "next" link directing to the next page, and a "find" link containing a URI Template which can be expanded with an 'id' variable to go directly to a specific order.

It also has two embedded resources, "orders".  Each of these has its own links to the associated "basket" and "customer" resources, and properties showing their "total", "currency" and "status".

Additionally, the order list resource has its own properties "currentlyProcessing" and "shippedToday".

## 7. Media Type Parameters

### 7.1. profile

The media type identifier application/hal+json MAY also include an additional "profile" parameter (as defined by [I-D.wilde-profile-link])

HAL documents that are served with the "profile" parameter still SHOULD include a "profile" link belonging to the root resource.

## 8. Recommendations

### 8.1. Self Link

Each Resource Object SHOULD contain a 'self' link that corresponds with the IANA registered 'self' relation (as defined by [RFC5988]) whose target is the resource's URI.

### 8.2. Link relations

Custom link relation types (Extension Relation Types in [RFC5988]) SHOULD be URIs that when dereferenced in a web browser provide relevant documentation, in the form of an HTML page, about the meaning and/or behaviour of the target Resource.  This will improve the discoverability of the API.

The CURIE Syntax [W3C.NOTE-curie-20101216] MAY be used for brevity for these URIs.  CURIEs are established within a HAL document via a set of Link Objects with the relation type "curies" on the root Resource Object.  These links contain a URI Template with the token 'rel', and are named via the "name" property.

```
{
  "_links": {
    "self": { "href": "/orders" },
    "curies": [{
      "name": "acme",
      "href": "http://docs.acme.com/relations/{rel}",
      "templated": true
    }],
    "acme:widgets": { "href": "/widgets" }
  }
}
```

The above demonstrates the relation "http://docs.acme.com/relations/widgets" being abbreviated to "acme:widgets" via CURIE syntax.

## 8.3. Hypertext Cache Pattern

The "hypertext cache pattern" allows servers to use embedded resources to dynamically reduce the number of requests a client makes, improving the efficiency and performance of the application.

Clients MAY be automated for this purpose so that, for any given link relation, they will read from an embedded resource (if present) in preference to traversing a link.

To activate this client behaviour for a given link, servers SHOULD add an embedded resource into the representation with the same relation.

Servers SHOULD NOT entirely "swap out" a link for an embedded resource (or vice versa) because client support for this technique is OPTIONAL.

The following examples shows the hypertext cache pattern applied to an "author" link:

Before:

```
{
  "_links": {
    "self": { "href": "/books/the-way-of-zen" },
    "author": { "href": "/people/alan-watts" }
  }
}
```

After:

```
{
  "_links": {
    "self": { "href": "/blog-post" },
    "author": { "href": "/people/alan-watts" }
  },
  "_embedded": {
    "author": {
      "_links": { "self": { "href": "/people/alan-watts" } },
      "name": "Alan Watts",
      "born": "January 6, 1915",
      "died": "November 16, 1973"
    }
  }
}
```

## 9. Security Considerations

TBD

## 10. IANA Considerations

TBD

## 11. Normative References

[I-D.wilde-profile-link]
Wilde, E., "The 'profile' Link Relation Type", draft-wilde-profile-link-04 (work in progress), October 2012.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.

[RFC5988] Nottingham, M., "Web Linking", RFC 5988, October 2010.

[RFC6570] Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., and D. Orchard, "URI Template", RFC 6570, March 2012.

[W3C.NOTE-curie-20101216]
Birbeck, M. and S. McCarron, "CURIE Syntax 1.0", World Wide Web Consortium NOTE NOTE-curie-20101216, December 2010, <http://www.w3.org/TR/2010/NOTE-curie-20101216>.

## Appendix A. Acknowledgements

Thanks to Darrel Miller, Mike Amundsen, and everyone in hal-discuss for their suggestions and feedback.

The author takes all responsibility for errors and omissions.

## Appendix B. Frequently Asked Questions

## B.1. How should a client know the meaning/structure/semantics/type of a resource?

There are two main approaches to solving this problem. Both involve exposing additional documentation describing the resource which may be human and/or machine readable (i.e. an HTML page and/or a JSON Schema document). The difference between the two approaches is in where that URI is shared with the client, which is either:

(1) The URI that was the preceding link relation type.

(2) A 'profile' link from the resource itself.

## B.2. Where can I find libraries for working with HAL?

A list of libraries is maintained here: http://stateless.co/hal_specification.html

## B.3. Why are the reserved properties prefixed with an underscore?

We elected for a prefix character to minimise risk of collisions with properties that represent the resource's state, and underscore was the character picked.

Another reason for prefixing the reserved properties is to make it visually apparent that the reserved properties are distinct from standard properties belonging to the resource.

## B.4. Are all underscore-prefixed properties reserved?

No, HAL only reserves the names detailed in this specification.

## B.5. Why does HAL have no forms?

Omitting forms from HAL was an intentional design decision that was made to keep it focused on linking for APIs. HAL is therefore a good candidate for use as a base media type on which to build more complex capabilities. An additional media type is planned for the future which will add form-like controls on top of HAL.

## Author's Address

Mike Kelly
Stateless

Email: mike@stateless.co
URI:   http://stateless.co/