Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 2, 2013

M. Kelly
Stateless

July 1, 2012

JSON Hypermedia API Language
draft-kelly-json-hal-02

Abstract

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

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 http://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 January 2, 2013.

Copyright Notice

Copyright (c) 2012 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 (http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Requirements 3
3. HAL Documents 3
4. Resource Objects 4
4.1. Reserved Properties 4
4.1.1. _links 4
4.1.2. _embedded 5
5. Link Objects 5
5.1. href 5
5.2. templated 5
5.3. type 5
5.4. name 5
5.5. profile 6
5.6. title 6
5.7. hreflang 6
6. Example Document 6
7. Media Type Parameters 8
7.1. profile 8
8. Recommendations 8
8.1. Self Link 8
8.2. Link relations 8
9. Security Considerations 9
10. IANA Considerations 9
11. Normative References 9
Appendix A. Acknowledgements 9
Appendix B. Frequently Asked Questions 10
B.1. How should a client know the meaning/structure/semantics/type of a resource? 10
B.2. Where can I find libraries for working with HAL? 10
B.3. Why are the reserved properties prefixed with an underscore? 10
B.4. Are all underscore-prefixed properties reserved? 10
B.5. Why does HAL have no forms? 10
Author's Address 10

1. Introduction

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

The JSON Hypermedia API 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 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.

It SHOULD NOT be used for a 'self' relation, the reserved "_self" property SHOULD be used instead.

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.

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. name

The "name" property is OPTIONAL.

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

5.5. 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.

5.6. 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]).

5.7. 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. A CURIE is established within a HAL document via a "curie" link on the root Resource Object. This link contains a URI Template with the token 'rel', and is named via the "name" property of the Link Object.

```
{
  "_links": {
    "self": { "href": "/orders" },
    "curie": {
      "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.

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-02 (work in progress), June 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]
McCarron, S. and M. Birbeck, "CURIE Syntax 1.0", World Wide Web Consortium NOTE NOTE-curie-20101216, December 2010, <http://www.w3.org/TR/2010/NOTE-curie-20101216>.

[1] <http://stateless.co/hal_specification.html>

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 and published at the HAL Home Page [1]

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

We elected for a prefix character to minimize 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 dicison 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/