TBD
Internet-Draft
Intended status: Standards Track
Expires: December 9, 2019

R. Stepanek
FastMail
June 7, 2019

JSContact: A JSON representation of addressbook data
draft-stepanek-jscontact-01

Abstract

This specification defines a data model and JSON representation of contact information that can be used for data storage and exchange in address book or directory applications. It aims to be an alternative to the vCard data format and to be unambiguous, extendable and simple to process. In contrast to the JSON-based jCard format, it is not a direct mapping from the vCard data model and expands semantics where appropriate.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines a data model for contact data normally used in address book or directory applications and services. It aims to be an alternative to the vCard data format [RFC6350] and to provide a JSON-based standard representation of contacts data.

The key design considerations for this data model are as follows:

o Most of the initial set of attributes should be taken from the vCard data format [RFC6350], but the specification should add new attributes or value types, or not support existing ones, where appropriate. Conversion between the data formats need not fully preserve semantic meaning.

o The attributes of the contacts data represented must be described as a simple key-value pair, reducing complexity of its representation.

o The data model should avoid all ambiguities and make it difficult to make mistakes during implementation.

o Extensions, such as new properties and components, MUST NOT lead to requiring an update to this document.

The representation of this data model is defined in the I-JSON format [RFC7493], which is a strict subset of the JavaScript Object Notation (JSON) Data Interchange Format [RFC8259]. Using JSON is mostly a pragmatic choice: its widespread use makes JSContact easier to adopt, and the availability of production-ready JSON implementations

eliminates a whole category of parser-related interoperability issues.

1.1. Relation to the xCard and jCard formats

The xCard [RFC6351] and jCard [RFC7095] specifications define alternative representations for vCard data, in XML and JSON format respectively. Both explicitly aim to not change the underlying data model. Accordingly, they are regarded as equal to vCard in the context of this document.

1.2. Terminology

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.

2. Contact

MIME type: "application/jscontact+json;type=jscontact"

A JSContact object stores contact information about a person, organization or company. It has the following properties:

- o uid: String (mandatory). A globally unique identifier, used to associate the object as the same across different systems, addressbooks and views. The value of this property MUST be unique across _all_ JSContact objects. [RFC4122] describes a range of established algorithms to generate universally unique identifiers (UUID), and the random or pseudo-random version is recommended. For compatibility with [RFC6350] UIDs, implementations MUST accept both URI and free-form text.

- o kind: String (optional). The kind of the entity the Contact represents. The value MUST be either one of the following values, registered in a future RFC, or a vendor-specific value:

  * "individual": a single person

  * "org": an organization

  * "location": a named location

- o fullName: String (mandatory) The full name(s) of a contact (e.g. the personal name and surname of an individual, the name of an organization).

- prefix: String[] (optional). The honorific title(s) of the contact (e.g. "Mr", "Ms", "Dr").

- personalName: String[] (optional). The personal name(s) of a contact (also known as "first name", "give name").

- surname: String[] (optional). The surname(s) of a contact (also known as "last name", "family name").

- additionalName: String[] (optional). The additional name(s) of a contact (also known as "middle name").

- suffix: String[] (optional). The honorific suffix(es) of the contact (e.g. "B.A.", "Esq.").

- nickname: String[] (optional). The nickname(s) of the contact.

- birthday: String (optional). The contact's birth date in the form "YYYY-MM-DD" (any part may be all 0s for unknown).

- anniversary: String (optional). The contact's anniversary date in the form "YYYY-MM-DD" (any part may be all 0s for unknown).

- organization: String[] (optional). The company or organization name and units associated with this contact. The first entry in the list names the organization, and any following entries name organizational units.

- jobTitle: String (optional). The job title or functional position of the contact.

- role: String (optional). The role, function or part played in a particular situation by the contact. In contrast to a job title, the role might differ for example in project contexts.

- emails: ContactInformation[] (optional). An array of ContactInformation objects where the values are email addresses. Types are:

  * "personal" The address is for emailing the contact in a personal context.

  * "work" The address is for emailing the contact in a professional context.

  * "other" The address is for some other purpose. A label property MAY be included to display next to the address to help the user identify its purpose.

- phones: ContactInformation[] (optional). An array of ContactInformation objects where the values are phone numbers. Types are:

  * "voice" The number is for calling the contact.

  * "fax" The number is for sending faxes to the contact.

  * "pager" The number is for a pager or beeper associated with the contact.

  * "other" The number is for some other purpose. A label property MAY be included to display next to the number to help the user identify its purpose.

  The following labels are pre-defined for phone contact information:

  * "private" The phone number should be used in a private context.

  * "work" The phone number should be used in a professional context

- online: ContactInformation[] (optional). An array of ContactInformation objects where the values are URIs or usernames associated with the contact for online services. Types are:

  * "uri" The value is a URI, e.g. a website link.

  * "username" The value is a username associated with the contact (e.g. for social media, or an IM client). A label property SHOULD be included to identify what service this is for. For compatibility between clients, this label SHOULD be the canonical service name, including capitalisation. e.g. "Twitter", "Facebook", "Skype", "GitHub", "XMPP".

  * "other" The value is something else not covered by the above categories. A label property MAY be included to display next to the number to help the user identify its purpose.

- preferredContactMethod: String (optional) Defines the preferred contact method. The value MUST be the property name of one of the ContactInformation lists: "emails", "phones", "online", "other".

- addresses: Address[] (optional). An array of Address objects, containing physical locations associated with the contact.

- notes: String (optional). Arbitrary notes about the contact.

A ContactInformation object has the following properties:

- o type: String (mandatory). Specifies the context of the contact information. This MUST be taken from the set of values allowed depending on whether this is part of the phones, emails or online property (see above).

- o label: String (optional). A label describing the value in more detail, especially if the type property has value "other" (but MAY be included with any type).

- o value: String (mandatory). The actual contact information, e.g. the email address or phone number.

- o isPreferred: Boolean (optional, default: "false"). Whether this ContactInformation is the preferred for its type. This SHOULD only be one per type.

An Address object has the following properties:

- o type: String (mandatory). Specifies the context of the address information. The value MUST be either one of the following values, registered in a future RFC, or a vendor-specific value:

  * "home" An address of a residence associated with the contact.

  * "work" An address of a workplace associated with the contact.

  * "billing" An address to be used with billing associated with the contact..

  * "postal" An address to be used for delivering physical items to the contact.

  * "other" An address not covered by the above categories.

- o label: String (optional). A label describing the value in more detail.

- o fullAddress: String (optional). The complete address, excluding type and label. This property is mainly useful to represent addresses of which the individual address components are unknown.

- o street: String (optional). The street address. This MAY be multiple lines; newlines MUST be preserved.

- o locality: String (optional). The city, town, village, post town, or other locality within which the street address may be found.

- region: String (optional). The province, such as a state, county, or canton within which the locality may be found.

- postcode: String (optional). The postal code, post code, ZIP code or other short code associated with the address by the relevant country's postal system.

- country: String (optional). The country name.

- countryCode: String (optional). The ISO-3166-1 country code.

- coordinates: String (optional) A [RFC5870] "geo:" URI for the address.

- timeZone: String (optional) Identifies the time zone this address is located in. This SHOULD be a time zone name registered in the IANA Time Zone Database [1]. Unknown time zone identifiers MAY be ignored by implementations.

- isPreferred: Boolean (optional, default: "false"). Whether this Address is the preferred for its type. This SHOULD only be one per type.

3. Contact Group

MIME type: "application/jscontact+json;type=jscontactgroup"

A JSContactGroup object represents a named set of contacts. It has the following properties:

- uid: String (mandatory). A globally unique identifier. The same requirements as for the JSContact uid property apply.

- name: String (optional). The user-visible name for the group, e.g. "Friends". This may be any UTF-8 string of at least 1 character in length and maximum 255 octets in size. The same name may be used by two different groups.

- contactIds: String[] (mandatory). The ids of the contacts in the group. Implementations MUST preserve the order of list entries.

4. IANA Considerations

TBD

5. Security Considerations

TBD

6. References

6.1. Normative References

[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/info/rfc2119>.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <https://www.rfc-editor.org/info/rfc4122>.

[RFC5870] Mayrhofer, A. and C. Spanring, "A Uniform Resource Identifier for Geographic Locations ('geo' URI)", RFC 5870, DOI 10.17487/RFC5870, June 2010, <https://www.rfc-editor.org/info/rfc5870>.

[RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, DOI 10.17487/RFC6350, August 2011, <https://www.rfc-editor.org/info/rfc6350>.

[RFC6351] Perreault, S., "xCard: vCard XML Representation", RFC 6351, DOI 10.17487/RFC6351, August 2011, <https://www.rfc-editor.org/info/rfc6351>.

[RFC7095] Kewisch, P., "jCard: The JSON Format for vCard", RFC 7095, DOI 10.17487/RFC7095, January 2014, <https://www.rfc-editor.org/info/rfc7095>.

[RFC7493] Bray, T., Ed., "The I-JSON Message Format", RFC 7493, DOI 10.17487/RFC7493, March 2015, <https://www.rfc-editor.org/info/rfc7493>.

[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/info/rfc8174>.

[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/info/rfc8259>.

6.2. URIs

[1] https://www.iana.org/time-zones

Author's Address

Robert Stepanek
FastMail
PO Box 234, Collins St West
Melbourne VIC 8007
Australia

Email: rsto@fastmailteam.com