calext
Internet-Draft
Updates: 9553, 9555 (if approved)
Intended status: Standards Track
Expires: 24 January 2026

R. Stepanek
Fastmail

23 July 2025

JSContact Version 2.0: A JSON Representation of Contact Data
draft-ietf-calext-jscontact-uid-05

Abstract

This document defines version "2.0" of JSContact. It defines the "uid" property of a Card object to be optional, rather than mandatory as defined in previous version "1.0". Other than changing the "uid" property, all other definitions of JSContact version "1.0" remain as defined in RFC 9553. This document updates RFC 9555 by redefining how to convert the now optional "uid" property from and to vCard.

Table of Contents

1. Notational Conventions

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.

The ABNF definitions in this document use the notations of [RFC5234]. ABNF rules not defined in this document are defined in either [RFC5234] (such as the ABNF for CRLF, WSP, DQUOTE, VCHAR, ALPHA, and DIGIT) or [RFC6350].

2. Introduction

[RFC9553] defines the "uid" property of a Card object, a mandatory property which contains an unique identifier for the entity represented by that contact card. This property being mandatory has shown to be applicable for some use cases, but has turned out to be an issue in other contexts.

For example, the CardDAV protocol [RFC6352] requires the UID property of a vCard object [RFC6350] to be set. Accordingly, an internet server that implements both CardDAV and JMAP for Contacts [RFC9610]

requires the "uid" property of a JSContact Card to be set. In contrast, protocols such as RDAP [RFC9083] have no use for the "uid" property, either because they use different identifiers, or they prefer to not include any unique identifier in the contact data at all. JSContact should not require them to generate unique identifiers that are irrelevant to their use case.

Also, one of the stated goals of JSContact is to be compatible with the semantics of the vCard data format (Section 1 of [RFC9553]). But [RFC6350] defines the UID property of a vCard to be optional, and consequently the semantics of JSContact and vCard differ for such a crucial common element.

In case of vCards without a UID property [RFC6350] (Section 6.7.6) being converted to JSContact, requiring unique identifiers is especially problematic: the Card "uid" property is mandatory and accordingly Section 2.1.1 of [RFC9555] requires implementations to generate some unique identifier for it during conversion, but it does not guarantee it to be the same across implementations or even one implementation converting the same Card multiple times. A recipient being unaware that the "uid" property value of such a Card object is ephemeral might refer to it in the "members" or "relatedTo" properties of another Card object, introducing invalid relations between contact cards.

3. JSContact Version 2.0

This document redefines the "uid" property of a Card object to become optional. Other than that, the property definition is left unchanged. This change requires the major version of JSContact to change, so this document defines the JSContact version to become "2.0".

Implementations MUST create JSContact data that complies with the definitions of version "2.0" (or some later registered version) and MUST set the "version" property of the JSContact Card object to that version. They MUST NOT reject a Card object without the "uid" property as invalid unless specified differently in another document, or unless the Card "version" property has value "1.0". As any valid version "1.0" JSContact Card also is valid according to version "2.0", there is no need to migrate existing JSContact data.

If to set the "uid" property is use-case specific. If an implementation is able to consistently generate the exact same unique identifier for a JSContact Card representing the same entity and no protocol-specific concerns prevail, it is recommended to set the "uid" property.

4. Redefined "uid" Property

This document redefines the type signature of the "uid" property, originally defined in Section 2.1.9 of [RFC9553]. The new type signature is:

uid: String (optional).

The remaining property definition is left unchanged, with the following additional paragraph:

A Card without a uid property can not be referred to as group member in the "members" property [RFC9553] (Section 2.1.6), or put in relation to another Card object in the "relatedTo" property [RFC9553] (Section 2.1.8).

5. Redefined Conversion Rule for the "uid" Property

This document redefines how to convert the Card "uid" property from vCard, originally defined in Section 2.1.1 of [RFC9555]. The new conversion rule is:

Implementations that convert a vCard without a UID property [RFC6350] (Section 6.7.6) to a Card of version "2.0" or higher MUST NOT generate a unique identifier as value for the "uid" property [RFC9553] (Section 2.1.9).

When converting a vCard without UID property to obsoleted JSContact version "1.0", implementations MUST generate a value for the "uid" property. Generating unique identifiers is implementation-specific. An implementation SHOULD generate the same value when generating the same Card multiple times, but MAY generate different values for each conversion. Section 2 describes why this is problematic. Consequently, implementations SHOULD NOT convert to version "1.0" Card objects.

6. IANA Considerations

6.1. Update to the JSContact Version Registry

IANA will update the "JSContact Version" registry, originally created in Section 3.4 of [RFC9553]. It will add the following record:

Major Version	Highest Minor Version	Reference
2	0	This document

Table 1: JSContact Version Registry

6.2. Update to the JSContact Properties Registry

IANA will update the "JSContact Properties" registry, originally created in Section 3.5 of [RFC9553]: For the "Reference/Description" column of the "uid" property, it will replace the reference to [RFC9553] with a reference to Section 4 of this document.

7. Security Considerations

This document does not provide new security considerations. The security considerations of Section 4 of [RFC9553] apply.

8. References

8.1. 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>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

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

[RFC6352] Daboo, C., "CardDAV: vCard Extensions to Web Distributed Authoring and Versioning (WebDAV)", RFC 6352, DOI 10.17487/RFC6352, August 2011, <https://www.rfc-editor.org/info/rfc6352>.

[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>.

[RFC9083] Hollenbeck, S. and A. Newton, "JSON Responses for the Registration Data Access Protocol (RDAP)", STD 95, RFC 9083, DOI 10.17487/RFC9083, June 2021, <https://www.rfc-editor.org/info/rfc9083>.

[RFC9553] Stepanek, R. and M. Loffredo, "JSContact: A JSON Representation of Contact Data", RFC 9553, DOI 10.17487/RFC9553, May 2024, <https://www.rfc-editor.org/info/rfc9553>.

[RFC9555] Loffredo, M. and R. Stepanek, "JSContact: Converting from and to vCard", RFC 9555, DOI 10.17487/RFC9555, May 2024, <https://www.rfc-editor.org/info/rfc9555>.

[RFC9610] Jenkins, N., Ed., "JSON Meta Application Protocol (JMAP) for Contacts", RFC 9610, DOI 10.17487/RFC9610, December 2024, <https://www.rfc-editor.org/info/rfc9610>.

Author’s Address

Robert Stepanek
Fastmail
PO Box 234
Collins St. West
Melbourne VIC 8007
Australia
Email: rsto@fastmailteam.com