

RFC 9982 : JSContact Version 2.0: A JSON Representation of Contact Data

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 mai 2026

Date de publication du RFC : Mai 2026

<https://www.bortzmeyer.org/9982.html>

Voici la version 2 du format de représentation d'entités (personnes ou organisations) JSContact (RFC 9553¹). En fait, ce numéro de version est trompeur, il n'y a qu'un seul changement, le membre `uid` qui était obligatoire devient facultatif. Mais ce petit changement, qui casse la compatibilité, oblige à changer de numéro de version.

Dans la section 2.1.9 du RFC 9553, l'`uid` ("*User Identifier*") était obligatoire. Alors que le vieux format vCard (RFC 6350) le disait facultatif, ce qui rendait difficile toute traduction automatique de vCard vers JSContact. En outre, s'il était cool d'avoir la garantie d'un identificateur unique pour chaque carte de visite au format JSContact, cela ne convenait pas dans tous les cas. Ainsi, RDAP (RFC 9083) n'utilise pas du tout cette propriété `uid`.

Donc, seul changement entre les versions 1 et 2, l'attribut `uid` devient optionnel <<https://www.iana.org/assignments/jscontact/jscontact.xml#jscontact-properties>>. On passe de :

```
uid: String (mandatory)
```

à :

```
*uid: String (optional).*
```

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc9553.txt>

Mais c'est suffisant pour obliger à changer le numéro de version de JSContact <<https://www.iana.org/assignments/jscontact/jscontact.xml#jscontact-version>> (RFC 9553, section 1.9).

Par exemple, cet objet JSContact (qui n'a pas d'uid) est désormais légal :

```
{
"@type": "Card",
"version": "2.0",
"name": {"components": [{"kind": "given", "value": "Jean"},
{"kind": "surname", "value": "Durand"}]},
"emails": {
"email": {
"address": "jean.durand@example.com"
}
}
}
```

(Avec la version 1, il aurait fallu quelque chose comme "uid": "a73c940e-b1d3-4f3c-aa50-c9749352c253" après la version.)