

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 3 December 2025

S. Bortzmeyer
Afnic
1 June 2025

Simple namespace-like proposal for JSON using a separator, for the RPP
draft-bortzmeyer-rpp-json-dot-namespaces-01

Abstract

This document proposes a lightweight convention for namespaces in the JSON payloads of RPP: a registry of namespaces plus the convention to write JSON names as namespace_shortname. It is not intended to be published as a RFC, just to stir discussion.

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 <https://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 3 December 2025.

Copyright Notice

Copyright (c) 2025 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. The proposal	4
2.1. Syntax	4
2.2. Example	4
3. Towards a full extensibility proposal	4
4. IANA Considerations	5
5. Security Considerations	5
6. References	5
6.1. Normative References	5
6.2. Informative References	5
Appendix A. Discussions and alternatives	6
Appendix B. Acknowledgments	7
Author's Address	7

1. Introduction

In the current discussions about RPP (RESTful Provisioning Protocol, see [I-D.wullink-rpp-requirements]), there has been a lot of talk about the extensibility of the protocol (see section 13 of the above draft). One of the issues raised by this requirement of extensibility is the fact that JSON ([RFC8259]), unlike XML, has no concept of namespaces. In the JSON world, the common way to introduce new members in objects is for the receiver to ignore unknown members (section 7 of the above draft makes that a requirement, see the "strict vs. lenient" discussion). This allows future extensions but, for a provisioning protocol, it is annoying since it means that requests from the client will be partially and silently ignored. Also, the lack of namespaces can create collisions between two extensions using the same name.

As an example, if we have this payload describing a contact:

```
{
  "name": "John Foobar",
  "date-of-birth": "2025-05-30",
  "country": "ZZ"
}
```

What if we want to extend RPP to add the fact that John Foobar is a physical person, for instance for compliance with data privacy regulations that apply only to physical persons, like the GDPR?

```
{
  "name": "John Foobar",
  "date-of-birth": "2025-05-30",
  "country": "ZZ",
  "is-a-physical-person": true
}
```

How the "is-a-physical-person" member be marked and processed?

There have been a lot of proposals to add some form of namespaces in JSON, often as part as an effort to design a schema language for JSON. In the IETF world, see for instance JMAP ([RFC8620]), which uses URIs to indicate extensions. Another proposal was [I-D.saintandre-json-namespaces], using also URIs. But none has caught, in the sense of being widely recognized as a good way to create namespaces in JSON. There is an effort [I-D.ietf-regext-rdap-extensions], to add namespaces in RDAP, for extensibility, relying on a namespace_shortcode syntax for member names.

For this document, the requirements are:

- * Do not try to solve the general problem, a RPP-specific solution is fine, using existing IETF infrastructure such as an IANA registry.
- * Ergonomy (ease of typing) is not mandatory (we won't write the JSON manually) but nice (we may have to read the JSON objects).
- * Do not use the existing proposed schema languages since there is apparently no consensus on these but also do not try to invent a new one (<https://xkcd.com/927/>).
- * Try to be simple and lightweight.
- * Be as close as possible from solutions used in the domain name world.

1.1. Requirements Language

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. The proposal

Core names are not namespaced (at this stage, it is not decided if mappings for domains, contacts and hosts will be regarded as core, or as extensions).

Namespaces have to be registered with IANA, per Section 4.

2.1. Syntax

The syntax follows ABNF ([RFC5234]). A namespace is identified by a string in restricted ASCII, just letters and dash:

```
namespace = %x2D / %x41-5A / %x61-7A
```

The name of a member of a JSON object in RPP will be "namespace_shortcode":

```
name = [namespace %x5F] shortcode
```

Presently, shortcode is not defined in RPP but we can assume it will be a (may be restricted?) form of JSON strings. It MUST NOT include underscores.

2.2. Example

```
{  
  "name": "John Foobar",  
  "date-of-birth": "2025-05-30",  
  "country": "ZZ",  
  "physical-person_status": true  
}
```

3. Towards a full extensibility proposal

Of course, this small proposal is far from a complete solution to extend RPP. Such a solution would need to include:

- * A way for the RPP clients and servers to advertise their capabilities (which extensions are supported).
- * A way to extend non-payload stuff such as HTTP status codes.

4. IANA Considerations

IANA is requested to create a registry of namespaces. These namespaces must conform to the syntax of Section 2.1. The policy to follow is "First Come First Served" as defined in Section 4.4 of [RFC8126]. (Alternative, "Specification Required", because it would be nice to have a specification but I understand that, currently, the actual semantics of "Specification Required" are not clear; for instance, is an Internet Draft sufficient?)

5. Security Considerations

There is probably zero security consequences of this proposal.

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>>.
- [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>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [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. Informative References

- [RFC8620] Jenkins, N. and C. Newman, "The JSON Meta Application Protocol (JMAP)", RFC 8620, DOI 10.17487/RFC8620, July 2019, <<https://www.rfc-editor.org/info/rfc8620>>.

[RFC9535] G. H. S. Sner, S., Ed., Normington, G., Ed., and C. Bormann, Ed., "JSONPath: Query Expressions for JSON", RFC 9535, DOI 10.17487/RFC9535, February 2024, <<https://www.rfc-editor.org/info/rfc9535>>.

[I-D.ietf-regext-rdap-extensions] Newton, A., Singh, J., and T. Harrison, "RDAP Extensions", Work in Progress, Internet-Draft, draft-ietf-regext-rdap-extensions-06, 29 April 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-regext-rdap-extensions-06>>.

[I-D.wullink-rpp-requirements] Wullink, M. and P. Kowalik, "RESTful Provisioning Protocol (RPP) - Requirements", Work in Progress, Internet-Draft, draft-wullink-rpp-requirements-01, 7 May 2025, <<https://datatracker.ietf.org/doc/html/draft-wullink-rpp-requirements-01>>.

[I-D.saintandre-json-namespaces] Hildebrand, J., Miller, M. A., and P. Saint-Andre, "JavaScript Object Notation (JSON) Namespaces", Work in Progress, Internet-Draft, draft-saintandre-json-namespaces-00, 24 October 2011, <<https://datatracker.ietf.org/doc/html/draft-saintandre-json-namespaces-00>>.

Appendix A. Discussions and alternatives

To be deleted before publication?

The choice of the character used by a separator is not purely bike-shedding (https://en.wikipedia.org/wiki/Law_of_triviality), since it may affect the use of protocols like JSONPath [RFC9535]. The use of dots, for instance, will make JSONPath expressions more complicated. So, the criteria for choosing a separator are:

- * As little interference as possible with JSONPath and similar protocols (so, dot, colon and comma are not ideal).
- * If possible, use of a character which is legal identifier in most programming languages, to avoid mapping issues (so, dot, colon, comma but also hash, semi-colon, and exclamation mark, are not ideal).
- * Try to stay close from what is used in protocols of the domain name world (which pleads for the underscore of [I-D.ietf-regext-rdap-extensions]).

Appendix B. Acknowledgments

Thanks to Andrew Newton about the choice of the separator.

Author's Address

Stéphane Bortzmeyer
Afnic
7 avenue du 8 mai 1945
78280 Guyancourt
France
Email: bortzmeyer+ietf@nic.fr
URI: <https://www.afnic.fr/>