INTERNATIONAL STANDARD

ISO/IEC 21778

First edition
2017-11

Information technology — The JSON data interchange syntax

Technologies de l'information — Syntaxe d'échange de données JSON

Reference number
ISO/IEC 21778:2017(E)

© ISO/IEC 2017



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword ..... **iv**
Introduction ..... **v**
1 Scope **1**
2 Conformance **1**
3 Normative References **1**
4 JSON Text **1**
5 JSON Values **2**
6 Objects **2**
7 Arrays **3**
8 Numbers **3**
9 String **4**
Bibliography **6**

© ISO/IEC 2017 – All rights reserved

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

ISO/IEC 21778 was prepared by Ecma International (as ECMA-404) and was adopted, under a special "fast-track procedure", by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

© ISO/IEC 2017 – All rights reserved

Introduction

JSON[1)] is a text syntax that facilitates structured data interchange between all programming languages. JSON is a syntax of braces, brackets, colons, and commas that is useful in many contexts, profiles, and applications. JSON stands for JavaScript Object Notation and was inspired by the object literals of JavaScript aka ECMAScript as defined in the ECMAScript Language Specification, Third Edition.[1] However, it does not attempt to impose ECMAScript's internal data representations on other programming languages. Instead, it shares a small subset of ECMAScript's syntax with all other programming languages. The JSON syntax is not a specification of a complete data interchange. Meaningful data interchange requires agreement between a producer and consumer on the semantics attached to a particular use of the JSON syntax. What JSON does provide is the syntactic framework to which such semantics can be attached

JSON syntax describes a sequence of Unicode code points. JSON also depends on Unicode in the hex numbers used in the `\u` escapement notation.

JSON is agnostic about the semantics of numbers. In any programming language, there can be a variety of number types of various capacities and complements, fixed or floating, binary or decimal. That can make interchange between different programming languages difficult. JSON instead offers only the representation of numbers that humans use: a sequence of digits. All programming languages know how to make sense of digit sequences even if they disagree on internal representations. That is enough to allow interchange.

Programming languages vary widely on whether they support objects, and if so, what characteristics and constraints the objects offer. The models of object systems can be wildly divergent and are continuing to evolve. JSON instead provides a simple notation for expressing collections of name/value pairs. Most programming languages will have some feature for representing such collections, which can go by names like `record, struct, dict, map, hash,` or `object`.

JSON also provides support for ordered lists of values. All programming languages will have some feature for representing such lists, which can go by names like `array, vector`, or `list`. Because objects and arrays can nest, trees and other complex data structures can be represented. By accepting JSON's simple convention, complex data structures can be easily interchanged between incompatible programming languages.

JSON does not support cyclic graphs, at least not directly. JSON is not indicated for applications requiring binary data.

It is expected that other standards will refer to this one, strictly adhering to the JSON syntax, while imposing semantics interpretation and restrictions on various encoding details. Such standards may require specific behaviours. JSON itself specifies no behaviour.

Because it is so simple, it is not expected that the JSON grammar will ever change. This gives JSON, as a foundational notation, tremendous stability.

JSON was first presented to the world at the `JSON.org` website in 2001. A definition of the JSON syntax was subsequently published as IETF RFC 4627 in July 2006. ECMA-262, Fifth Edition (2009) included a normative specification of the JSON grammar. This specification, ECMA-404, replaces those earlier definitions of the JSON syntax. Concurrently, the IETF published RFC 7158/7159 and in 2017 RFC 8259 as updates to RFC 4627. The JSON syntax specified by this specification and by RFC 8259 are intended to be identical.

1) Pronounced **/ˈdʒeɪˑsən/**, as in "Jason and The Argonauts".

© ISO/IEC 2017 – All rights reserved

Information technology — The JSON data interchange syntax

1 Scope

JSON is a lightweight, text-based, language-independent syntax for defining data interchange formats. It was derived from the ECMAScript programming language, but is programming language independent. JSON defines a small set of structuring rules for the portable representation of structured data.

The goal of this specification is only to define the syntax of valid JSON texts. Its intent is not to provide any semantics or interpretation of text conforming to that syntax. It also intentionally does not define how a valid JSON text might be internalized into the data structures of a programming language. There are many possible semantics that could be applied to the JSON syntax and many ways that a JSON text can be processed or mapped by a programming language. Meaningful interchange of information using JSON requires agreement among the involved parties on the specific semantics to be applied. Defining specific semantic interpretations of JSON is potentially a topic for other specifications. Similarly, language mappings of JSON can also be independently specified. For example, ECMA-262 defines mappings between valid JSON texts and ECMAScript's runtime data structures.

2 Conformance

A conforming JSON text is a sequence of Unicode code points that strictly conforms to the JSON grammar defined by this specification.

A conforming processor of JSON texts should not accept any inputs that are not conforming JSON texts. A conforming processor may impose semantic restrictions that limit the set of conforming JSON texts that it will process.

3 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 10646, *Information Technology – Universal Coded Character Set (UCS)*

The Unicode Consortium. The Unicode Standard http://www.unicode.org/versions/latest

Bray, T., Ed. " The JavaScript Object Notation (JSON) Data Interchange Format", RFC 8259.

This specification and [RFC 8259] both provide specifications of the JSON grammar but do so using different formalisms. The intent is that both specifications define the same syntactic language. If a difference is found between them, Ecma International and the IETF will work together to update both documents. If an error is found with either document, the other should be examined to see if it has a similar error, and fixed if possible. If either document is changed in the future, Ecma International and the IETF will work together to ensure that the two documents stay aligned through the change. RFC 8259, also defines various semantic restrictions on the use of the JSON syntax. Those restrictions are not normative for this specification.

4 JSON Text

A JSON text is a sequence of tokens formed from Unicode code points that conforms to the JSON value grammar. The set of tokens includes six structural tokens, strings, numbers, and three literal name tokens.

© ISO/IEC 2017 – All rights reserved