Internet Engineering Task Force	P. Hallam-Baker
Internet-Draft	Comodo Group Inc.
Intended status: Standards Track	July 31, 2012
Expires: February 1, 2013	

TOC

Simple Cloud Interoperable Jason Encoding for REST Web Services (SCIJER) Bindings draft-hallambaker-scijer-00

Abstract

SCIJER (pronounced skyjer) specifies a mapping of data types to JavaScript Object Notation (JSON) encoding. The JSON specification defines encodings for integers, strings, arrays and objects. This document specifies a mapping between JSON encoding and an extended set of types.

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 http://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 February 1, 2013.

Table of Contents

---

1. Definitions

TOC

1.1. Requirements Language

TOC

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in **[RFC2119]**.

1.2. Defined Terms

TOC

The following terms are used in this document:

---

2. Introduction

TOC

JSON is a format for encoding data structures in UTF8 text format with similar capabilities to XML and ASN.1 but with lower coding overhead. Unlike XML and ASN.1, JSON does not have a schema language.

This draft defines a mapping between JSON data types and programming language types that MAY be used as the basis for a schema language definition.

---

3. Bindings

TOC

Specify conversions between JSON encoding and C, C# and Java types.

3.1. Object

TOC

JSON
C
C#
Java

TOC

3.2. Serializable

TOC

3.2.1. Array

TOC

JSON
C
C#
Java

3.2.2. List

TOC

JSON
C
C#
Java

3.3. Boolean

TOC

JSON
: true | false

C
: int

C#
: bool

C#
: boolean

3.4. Integer

TOC

JSON
: number without decimal point

C, C#, Java
: int or long

3.5. Floating Point

TOC

JSON
: number with decimal point

C, C#, Java
: float or double

3.6. String

TOC

The String type maps to the JSON string encoding. The following subtypes are defined to represent commonly used syntactic restrictions.

JSON
: String encoding

C
: char * [NB must use appropriate wrappers to encode/decode UTF8]

C#, Java

String

URI

Label

Name

TOC

3.6.1. URI

A URI is a string that contains a URI as specified in **[RFC3986]**.

TOC

3.6.2. Label

A label is a string that MUST NOT contain any ASCII character other than digits (0-9), alphabetic (a-z, A-Z), underscore or dash. All other UNICODE characters are permitted.

The Label type provide a means of avoiding content injection attacks by ensuring that a field does not contain characters commonly used as control characters in scripting languages.

TOC

3.6.3. Name

A Name is a sequence of Labels separated by period characters '.'.

Note that the syntactic restriction of a Name is not exactly the same as the restriction on a DNS address.

TOC

3.7. Enumeration

An enumeration is encoded as a Label.

TOC

3.8. Time

SSCIJER supports two encodings

DateTime
: A string encoded in **[RFC3339]** format. This provides a human readable representation.

ElapsedTime
: A number that represents the number of seconds that have elapsed since January 1, 0001 at 00:00:00.000 in the Gregorian calendar. Decimals MAY be used to represent time intervals of less than a second.

Conversion between the two encodings MUST be informed by the times at which leap seconds were introduced into the UTC coding. This creates a challenge in the case that a device only has access to time in one form but not the other.

In order to address this problem, implementations MUST accept time values encoded in either format and SHOULD emit the encoding specified in the schema.

TOC

3.8.1. DateTime

JSON
: String [Encoded in **[RFC3339]** format.]

C
: System dependent

C#
: DateTime

Java
: Date

Represents a date and time in the Gregorian callendar using the format described in **[RFC3339]**.

TOC

3.8.2. ElapsedTime

JSON
: Number

C
: System dependent

C#
: DateTime

Java
: Date

A number that represents the number of seconds that have elapsed since January 1, 0001 at 00:00:00.000 in the Gregorian calendar. Decimals MAY be used to represent time intervals of less than a second.

TOC

3.9. Binary

JSON
: String [Base64 encoding of binary data]

C
: {unsigned char *, length}

C#, Java
: byte []

TOC

3.10.

TOC

4. Security Considerations

TOC

5. IANA Considerations

TOC

6. Acknowledgements

The name of this draft was inspired by Mark Nottingham.

TOC

7. Normative References

[RFC1035] Mockapetris, P., "**Domain names - implementation and specification**," STD 13, RFC 1035, November 1987

(**TXT**).

[RFC2119] **Bradner, S.**, "**Key words for use in RFCs to Indicate Requirement Levels**," BCP 14, RFC 2119, March 1997 (**TXT**, **HTML**, **XML**).

[RFC3339] **Klyne, G., Ed.** and **C. Newman**, "**Date and Time on the Internet: Timestamps**," RFC 3339, July 2002 (**TXT**, **HTML**, **XML**).

[RFC3986] **Berners-Lee, T.**, **Fielding, R.**, and **L. Masinter**, "**Uniform Resource Identifier (URI): Generic Syntax**," STD 66, RFC 3986, January 2005 (**TXT**, **HTML**, **XML**).

Author's Address

TOC

Phillip Hallam-Baker
Comodo Group Inc.
Email: **philliph@comodo.com**