INTERNATIONAL STANDARD

ISO/IEC 20802-2

Information technology — Open data protocol (OData) v4.0

Part 2:
OData JSON Format

Technologies de l'information — Protocole de données ouvertes (OData) v4.0 —

Partie 2: Format OData JSON

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 20802-2:2016
https://standards.iteh.ai/catalog/standards/sist/764d8eb3-95c9-4840-9b7a-
17a9adce6f84/iso-iec-20802-2-2016

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

ISO/IEC 20802-2 was prepared by OASIS and was adopted, under the PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by the national bodies of ISO and IEC.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

OData JSON Format Version 4.0 Plus Errata 02

OASIS Standard incorporating Approved Errata 02

30 October 2014

Specification URIs

This version:

http://docs.oasis-open.org/odata/odata-json-format/v4.0/errata02/os/odata-json-format-v4.0-errata02-os-complete.doc (Authoritative)

http://docs.oasis-open.org/odata/odata-json-format/v4.0/errata02/os/odata-json-format-v4.0-errata02-os-complete.html

http://docs.oasis-open.org/odata/odata-json-format/v4.0/errata02/os/odata-json-format-v4.0-errata02-os-complete.pdf

Previous version:

http://docs.oasis-open.org/odata/odata-json-format/v4.0/errata01/os/odata-json-format-v4.0-errata01-os-complete.doc (Authoritative)

http://docs.oasis-open.org/odata/odata-json-format/v4.0/errata01/os/odata-json-format-v4.0-errata01-os-complete.html

http://docs.oasis-open.org/odata/odata-json-format/v4.0/errata01/os/odata-json-format-v4.0-errata01-os-complete.pdf

Latest version:

http://docs.oasis-open.org/odata/odata-json-format/v4.0/odata-json-format-v4.0.doc (Authoritative)

http://docs.oasis-open.org/odata/odata-json-format/v4.0/odata-json-format-v4.0.html

http://docs.oasis-open.org/odata/odata-json-format/v4.0/odata-json-format-v4.0.pdf

Technical Committee:

OASIS Open Data Protocol (OData) TC

Chairs:

Ralf Handl (ralf.handl@sap.com), SAP AG

Ram Jeyaraman (Ram.Jeyaraman@microsoft.com), Microsoft

Editors:

Ralf Handl (ralf.handl@sap.com), SAP AG

Michael Pizzo (mikep@microsoft.com), Microsoft

Martin Zurmuehl (martin.zurmuehl@sap.com), SAP AG

Mark Biamonte (mark.biamonte@progress.com), Progress Software

Additional artifacts:

This prose specification is one component of a Work Product that also includes:

- *OData JSON Format Version 4.0 Errata 02*. Edited by Ralf Handl, Michael Pizzo, and Martin Zurmuehl. 30 October 2014. OASIS Approved Errata. http://docs.oasis-open.org/odata/odata-json-format/v4.0/errata02/os/odata-json-format-v4.0-errata02-os.html.
- Change-marked (redlined) version. *OData JSON Format Version 4.0 Plus Errata 02 (redlined)*. Edited by Ralf Handl, Michael Pizzo, Martin Zurmuehl, and Mark Biamonte. 30 October 2014. OASIS Standard incorporating Approved Errata 02. http://docs.oasis-

open.org/odata/odata-json-format/v4.0/errata02/os/odata-json-format-v4.0-errata02-os-redlined.html.

Related work:

This specification is related to:

- *OData Version 4.0.* OASIS Standard. Multi-part Work Product that includes:
  - *OData Version 4.0 Part 1: Protocol.* http://docs.oasis-open.org/odata/odata/v4.0/os/part1-protocol/odata-v4.0-os-part1-protocol.html.
  - *OData Version 4.0 Part 2: URL Conventions.* http://docs.oasis-open.org/odata/odata/v4.0/os/part2-url-conventions/odata-v4.0-os-part2-url-conventions.html.
  - *OData Version 4.0 Part 3: Common Schema Definition Language (CSDL).* http://docs.oasis-open.org/odata/odata/v4.0/os/part3-csdl/odata-v4.0-os-part3-csdl.html.
  - ABNF components: http://docs.oasis-open.org/odata/odata/v4.0/os/abnf/
  - Vocabulary components: http://docs.oasis-open.org/odata/odata/v4.0/os/vocabularies/
  - XML schemas: http://docs.oasis-open.org/odata/odata/v4.0/os/schemas/
  - OData Metadata Service Entity Model: http://docs.oasis-open.org/odata/odata/v4.0/os/models/MetadataService.edmx.
- *OData Atom Format Version 4.0.* Edited by Martin Zurmuehl, Michael Pizzo, and Ralf Handl. Latest version: http://docs.oasis-open.org/odata/odata-atom-format/v4.0/odata-atom-format-v4.0.html.

Abstract:

The Open Data Protocol (OData) for representing and interacting with structured content is comprised of a set of specifications. The core specification for the protocol is in OData Version 4.0 Part 1: Protocol; this document extends the former by defining representations for OData requests and responses using a JSON format.

Status:

This document was last revised or approved by the OASIS Open Data Protocol (OData) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=odata#technical.

TC members should send comments on this specification to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment" button on the TC's web page at https://www.oasis-open.org/committees/odata/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (https://www.oasis-open.org/committees/odata/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[OData-JSON-Format-v4.0-plus-Errata02]

OData JSON Format Version 4.0 Plus Errata 02. Edited by Ralf Handl, Michael Pizzo, Martin Zurmuehl, and Mark Biamonte. 30 October 2014. OASIS Standard incorporating Approved Errata 02. http://docs.oasis-open.org/odata/odata-json-format/v4.0/errata02/os/odata-json-format-v4.0-errata02-os-complete.html. Latest version: http://docs.oasis-open.org/odata/odata-json-format/v4.0/odata-json-format-v4.0.html.

Notices

Table of Contents

iTeh STANDARD PREVIEW
(standards.iteh.ai)

1 Introduction

The OData protocol is comprised of a set of specifications for representing and interacting with structured content. The core specification for the protocol is in **[OData-Protocol]**; this document is an extension of the core protocol. This document defines representations for the OData requests and responses using the JavaScript Object Notation (JSON), see **[RFC7159]**.

An OData JSON payload may represent:

- a single primitive value
- a collection of primitive values
- a single complex type value
- a collection of complex type values
- a single entity or entity reference
- a collection of entities or entity references
- a collection of changes
- a service document describing the top-level resources exposed by the service
- an error.

1.1 Terminology

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 Normative References

[GeoJSON]	Howard Butler, Martin Daly, Alan Doyle, Sean Gillies, Tim Schaub and Stefan Drees, "The GeoJSON Format" draft-butler-geojson-02, 15 March 2014. http://tools.ietf.org/html/draft-butler-geojson-02.
[I-JSON]	Bray, T., Ed., "The I-JSON Message Format" draft-bray-i-json-01, 06 January 2014. http://tools.ietf.org/html/draft-bray-i-json-01
[OData-ABNF]	*OData ABNF Construction Rules Version 4.0.* See link in "Related work" section on cover page.
[OData-CSDL]	OData Version 4.0 Part 3: Common Schema Definition Language (CSDL). See link in "Related work" section on cover page.
[OData-Protocol]	*OData Version 4.0 Part 1: Protocol.* See link in "Related work" section on cover page.
[OData-URL]	*OData Version 4.0 Part 2: URL Conventions.* See link in "Related work" section on cover page.
[OData-VocCap]	*OData Capabilities Vocabulary.* See link in "Related work" section on cover page.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.
[RFC3986]	Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", IETF RFC3986, January 2005. http://www.ietf.org/rfc/rfc3986.txt.
[RFC3987]	Duerst, M. and, M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, January 2005. http://www.ietf.org/rfc/rfc3987.txt.
[RFC7159]	Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014. http://tools.ietf.org/html/rfc7159.

| **[RFC5646]** | Phillips, A., Ed., and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009. http://tools.ietf.org/html/rfc5646. |
| **[ECMAScript]** | *ECMAScript Language Specification Edition 5,1*. June 2011. Standard ECMA-262. http://www.ecma-international.org/publications/standards/Ecma-262.htm. |

1.3 Typographical Conventions

Keywords defined by this specification use this `monospaced` font.

```
Normative source code uses this paragraph style.
```

Some sections of this specification are illustrated with non-normative examples.

Example 1: text describing an example uses this paragraph style

```
Non-normative examples use this paragraph style.
```

All examples in this document are non-normative and informative only.

All other text is normative unless otherwise labeled.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 20802-2:2016
https://standards.iteh.ai/catalog/standards/sist/764d8eb3-95c9-4840-9b7a-
17a9adce6f84/iso-iec-20802-2-2016

2 JSON Format Design

JSON, as described in [RFC7159], defines a text format for serializing structured data. Objects are serialized as an unordered collection of name-value pairs.

JSON does not define any semantics around the name/value pairs that make up an object, nor does it define an extensibility mechanism for adding control information to a payload.

OData's JSON format extends JSON by defining general conventions for name-value pairs that annotate a JSON object, property or array. OData defines a set of canonical annotations for control information such as ids, types, and links, and custom annotations MAY be used to add domain-specific information to the payload.

A key feature of OData's JSON format is to allow omitting predictable parts of the wire format from the actual payload. To reconstitute this data on the receiving end, expressions are used to compute missing links, type information, and other control data. These expressions (together with the data on the wire) can be used by the client to compute predictable payload pieces as if they had been included on the wire directly.

Annotations are used in JSON to capture control information that cannot be predicted (e.g., the next link of a collection) as well as a mechanism to provide values where a computed value would be wrong (e.g., if the media read link of one particular entity does not follow the standard URL conventions). Computing values from metadata expressions is compute intensive and some clients might opt for a larger payload size to avoid computational complexity; to accommodate for this the `Accept` header allows the client to control the amount of control information added to the response.

To optimize streaming scenarios, there are a few restrictions that MAY be imposed on the sequence in which name/value pairs appear within JSON objects. For details on the ordering requirements see Payload Ordering Constraints.

3 Requesting the JSON Format

The OData JSON format can be requested using the `$format` query option in the request URL with the MIME type `application/json`, optionally followed by format parameters, or the case-insensitive abbreviation `json` which MUST NOT be followed by format parameters.

Alternatively, this format can be requested using the `Accept` header with the MIME type `application/json`, optionally followed by format parameters.

If specified, `$format` overrides any value specified in the `Accept` header.

Possible format parameters are:

- `odata.metadata`
- `IEEE754Compatible`
- `odata.streaming`

Services SHOULD advertise the supported MIME types by annotating the entity container with the term `Capabilities.SupportedFormats` defined in **[OData-VocCap]**, listing all available formats and combinations of supported format parameters.

3.1 Controlling the Amount of Control Information in Responses

The amount of control information needed (or desired) in the payload depends on the client application and device. The `odata.metadata` parameter can be applied to the `Accept` header of an OData request to influence how much control information will be included in the response.

Other `Accept` header parameters (e.g., `odata.streaming`) are orthogonal to the `odata.metadata` parameter and are therefore not mentioned in this section.

If a client prefers a very small wire size and is intelligent enough to compute data using metadata expressions, the `Accept` header should include `odata.metadata=minimal`. If computation is more critical than wire size or the client is incapable of computing control information, `odata.metadata=full` directs the service to inline the control information that normally would be computed from metadata expressions in the payload. `odata.metadata=none` is an option for clients that have out-of-band knowledge or don't require control information.

3.1.1 `odata.metadata=minimal`

The `odata.metadata=minimal` format parameter indicates that the service SHOULD remove computable control information from the payload wherever possible. This is the default value for the `odata.metadata` parameter and will be assumed if no other value is specified in the `Accept` header or `$format` query option. The response payload MUST contain at least the following common annotations:

- `odata.context`: the root context URL of the payload and the context URL for any deleted entries or added or deleted links in a delta response, or for entities or entity collections whose set cannot be determined from the root context URL
- `odata.metadataEtag`: the ETag of the metadata document as applicable
- `odata.etag`: the ETag of the entity, as appropriate
- `odata.count`: the total count of a collection of entities or collection of entity references, if requested
- `odata.nextLink`: the next link of a collection with partial results
- `odata.deltaLink`: the delta link for obtaining changes to the result, if requested

In addition, `odata` annotations MUST appear in the payload for cases where actual values are not the same as the computed values and MAY appear otherwise. When `odata` annotations appear in the payload, they are treated as exceptions to the computed values.