Excel Services REST Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact dochelp@microsoft.com.
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/19/2010</td>
<td>1.0</td>
<td>Major</td>
<td>Initial Availability.</td>
</tr>
<tr>
<td>3/31/2010</td>
<td>1.01</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>4/30/2010</td>
<td>1.02</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>6/7/2010</td>
<td>1.03</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>6/29/2010</td>
<td>1.04</td>
<td>Editorial</td>
<td>Changed language and formatting in the technical content.</td>
</tr>
<tr>
<td>7/23/2010</td>
<td>1.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/27/2010</td>
<td>1.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>11/15/2010</td>
<td>1.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>12/17/2010</td>
<td>1.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>1.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/10/2011</td>
<td>1.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>1/20/2012</td>
<td>2.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/11/2012</td>
<td>2.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>2.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/12/2012</td>
<td>2.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>2.0.1</td>
<td>Editorial</td>
<td>Changed language and formatting in the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>3.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/30/2013</td>
<td>4.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>4.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>4.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>4.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>4.2</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>4.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/26/2016</td>
<td>5.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/15/2016</td>
<td>5.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
</tbody>
</table>

[MS-ESREST] - v20190319
Excel Services REST Protocol
Copyright © 2019 Microsoft Corporation
Release: March 19, 2019
<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>5.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/24/2018</td>
<td>6.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/1/2018</td>
<td>7.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>3/19/2019</td>
<td>7.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
</tbody>
</table>
Table of Contents

1 Introduction ........................................................................................................... 6
  1.1 Glossary .......................................................................................................... 6
  1.2 References ..................................................................................................... 7
  1.2.1 Normative References ............................................................................. 7
  1.2.2 Informative References ............................................................................ 8
  1.3 Overview ......................................................................................................... 8
  1.4 Relationship to Other Protocols ..................................................................... 8
  1.5 Prerequisites/Preconditions .......................................................................... 9
  1.6 Applicability Statement ................................................................................ 9
  1.7 Versioning and Capability Negotiation ....................................................... 9
  1.8 Vendor-Extensible Fields ............................................................................. 10
  1.9 Standards Assignments .............................................................................. 10

2 Messages .............................................................................................................. 11
  2.1 Transport ...................................................................................................... 11
  2.2 Common Message Syntax ........................................................................... 11
    2.2.1 Namespaces ............................................................................................ 11
    2.2.2 Messages ............................................................................................... 11
    2.2.3 Elements ............................................................................................... 11
    2.2.4 Complex Types ...................................................................................... 11
      2.2.4.1 Range ............................................................................................... 12
      2.2.4.1.1 Range XML Value ...................................................................... 12
      2.2.4.1.2 Range JSON Value .................................................................. 12
      2.2.4.2 Row ............................................................................................... 12
      2.2.4.2.1 Row XML Value ....................................................................... 12
      2.2.4.2.2 Row JSON Value ..................................................................... 12
      2.2.4.3 Cell ................................................................................................. 13
      2.2.4.3.1 Cell XML Value ........................................................................ 13
      2.2.4.3.2 Cell JSON Value ..................................................................... 13
      2.2.4.4 JSONReference ............................................................................ 13
      2.2.4.5 JSONCollection ............................................................................ 14
      2.2.5 Simple Types ........................................................................................ 14
      2.2.5.1 Error .............................................................................................. 14
      2.2.6 Attributes ........................................................................................... 15
      2.2.7 Groups ................................................................................................. 15
      2.2.8 Attribute Groups ............................................................................... 15
      2.2.9 Common Data Structures .................................................................... 15

3 Protocol Details .................................................................................................. 16
  3.1 Server Details ............................................................................................... 16
    3.1.1 Abstract Data Model ............................................................................ 16
      3.1.1.1 Model Entity Set ........................................................................... 17
      3.1.1.2 Session Entity Set ......................................................................... 18
      3.1.1.3 Ranges Entity Set ........................................................................... 20
      3.1.1.4 Range Entity .................................................................................. 20
      3.1.1.5 Tables Entity Set ........................................................................... 21
      3.1.1.6 Table Entity .................................................................................... 21
      3.1.1.7 PivotTables Entity Set ................................................................. 22
      3.1.1.8 PivotTable Entity ......................................................................... 22
      3.1.1.9 Charts Entity Set .......................................................................... 22
      3.1.1.10 Sessions Entity Set ................................................................. 23
      3.1.1.11 Slicers ......................................................................................... 24
      3.1.1.12 Slicer ............................................................................................ 24
      3.1.1.13 Timelines ..................................................................................... 26
      3.1.1.14 Timeline ..................................................................................... 26
1 Introduction

The Excel Services REST Protocol extensions to the Atom Publishing Protocol (AtomPub) [RFC5023] and its extensions specified in the Open Data Protocol (ODATA) Specification [MS-ODATA] enable applications to expose spreadsheet data, using common Web technologies, as a data service that can be consumed by clients within corporate networks and across the Internet.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

atom feed: An XML structure that contains metadata about content, such as the language version and the date when the content was last modified, and is sent to subscribers by using the Atom Publishing Protocol (AtomPub), as described in [RFC4287].

cell: A box that is formed by the intersection of a row and a column in a worksheet or a table. A cell can contain numbers, strings, and formulas, and various formats can be applied to that data.

cell error value: Any of a number of special values that are returned as a result of an unsuccessful formula calculation.

data culture: The language that is used to specify number formatting for data.

formula: A logical equation or function that produces a result in a spreadsheet application.

HTML fragment: Lines of text that adhere to HTML tag rules, as described in [HTML], but do not have processing instructions or any other type of header information.

Hypertext Markup Language (HTML): An application of the Standard Generalized Markup Language (SGML) that uses tags to mark elements in a document, as described in [HTML].

JavaScript Object Notation (JSON): A text-based, data interchange format that is used to transmit structured data, typically in Asynchronous JavaScript + XML (AJAX) web applications, as described in [RFC7159]. The JSON format is based on the structure of ECMAScript (Jscript, JavaScript) objects.

list: An organization of a region of cells into a tabular structure in a workbook.

named range: See defined name.

PivotTable: An interactive table that summarizes large amounts of data from various sources by using format and calculation methods. Row and column headings can be rotated to view different summaries of the source data, filter the data, or display detail data for specific areas.

published item: A specific named object that is in a published workbook.

range: An addressable region that is in a workbook. A range typically consists of zero or more cells and represents a single, contiguous rectangle of cells on a single sheet.

row: A single set of data that is displayed horizontally in a worksheet or a table.

slicer: A mechanism that is used to filter data in one or more PivotTable reports or cube functions.

table: A list that is defined in a workbook.

UI culture: The language that is used to display strings and graphical elements in a user interface.
**Uniform Resource Locator (URL):** A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

**Web Services Description Language (WSDL):** An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

**workbook:** A container for a collection of sheets.

**workbook file:** A file that contains a byte stream representation of a workbook.

**worksheet:** A single logical container for a set of tabular data and other objects in a workbook.

**XML:** The Extensible Markup Language, as described in [XML1.0].

**XML namespace:** A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

**XML schema:** A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

### 1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the **Errata**.

#### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

- [MS-ODATA] Microsoft Corporation, "Open Data Protocol (OData)".
- [MS-RDL] Microsoft Corporation, "Report Definition Language File Format".
1.3 Overview

This protocol enables protocol clients to access information stored in **workbooks** maintained by a protocol server using common Web technologies. Various types of resources within spreadsheets can be retrieved, such as **ranges**, **tables**, charts, **slicers**, timelines, interactive reports and **PivotTable** reports. The information returned by the protocol can be in one of several formats, such as **HTML**, image, **JavaScript Object Notation (JSON)**, and as an **Atom feed**. This protocol client also allows for values to be inserted into **cells** of the workbook.

A typical scenario for using this protocol is a custom application where workbook data is used as a data source that is connected to other processes that act on that data, such as displaying it in a novel way or taking specific action based on the contents of that data. Another typical scenario for this protocol is in embedding of workbook data into applications that support common Web standards.

1.4 Relationship to Other Protocols

This protocol uses the AtomPub messaging protocol for formatting requests and responses as described in [RFC5023] and URI and Payload extensions to the AtomPub protocol as described in [MS-ODATA]. It transmits these messages using the HTTP protocol as described in [RFC2616] or the HTTPS protocol as described in [RFC2818].

The following diagram shows the underlying messaging and transport stack that the protocol uses:
1.5 Prerequisites/Preconditions

This protocol operates against a resource in a workbook that is identified by a URL that is known by protocol clients. The URL is formed by appending the workbook relative path and the resource location to "http://<ServerName>/_vti_bin/ExcelRest.aspx", for example: http://www.example.com/_vti_bin/ExcelRest.aspx/Docs/Documents/sampleWorkbook.xlsx/model/Charts('SampleChart'). It is a prerequisite that the client obtain a URL to the server before these extensions can be used.

1.6 Applicability Statement

AtomPub, as described in [RFC5023], in combination with the extensions defined in this document, is appropriate for use in Web services that need a uniform, flexible, general purpose interface for exposing retrieve operations on a data model to clients.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported transports**: This document can be implemented on top of the Atom Publishing Protocol as described in section 2.1.
- **Protocol versions**: Servers specify the protocol version by using the X-XLSVersion response header.

  The X-XLSVersion response header is present on any response message.

  The syntax of the X-XLSVersion header is defined as follows:

  X-XLSVersion = "X-XLSVersion:" VersionNum

  VersionNum = DIGIT *DIGIT "." DIGIT *DIGIT "." DIGIT *DIGIT

  The VersionNum section of the header value states the version of the protocol server.

- **Localization**: This document does not specify any localization-dependent behavior.
- **Capability negotiation**: The protocol that is defined in this document enables limited capability negotiation using the X-XLSVersion response header.

  On a response from the server to the client, the X-XLSVersion header needs to be specified. The value states the version of the protocol the server used to generate the response.
1.8 Vendor-Extensible Fields
None.

1.9 Standards Assignments
None.
2 Messages

2.1 Transport

No new transports are required except for those specified in [MS-ODATA] section 2.1.

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses XML schema, as specified in [XMLSCHEMA1/2] and [XMLSCHEMA2/2], and WSDL, as specified in [WSDL].

2.2.1 Namespaces

This specification defines and references various XML namespaces by using the mechanisms specified in [XMLNS]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>tns</td>
<td><a href="http://schemas.microsoft.com/office/excel/server/webservices">http://schemas.microsoft.com/office/excel/server/webservices</a></td>
<td></td>
</tr>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>[XMLSCHEMA1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[XMLSCHEMA2]</td>
</tr>
<tr>
<td>atom</td>
<td><a href="http://www.w3.org/2005/Atom">http://www.w3.org/2005/Atom</a></td>
<td>[RFC4287]</td>
</tr>
</tbody>
</table>

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

This specification does not define any common XML schema element definitions.

2.2.4 Complex Types

The following table summarizes the set of common XML schema complex type definitions defined by this specification. XML schema complex type definitions that are specific to a particular operation are described with the operation.

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>A range of cells in a worksheet.</td>
</tr>
<tr>
<td>Row</td>
<td>Represents a row in a worksheet.</td>
</tr>
<tr>
<td>Cell</td>
<td>Represents a cell in a worksheet.</td>
</tr>
<tr>
<td>JSONReference</td>
<td>Represents a reference to an Entity Set.</td>
</tr>
</tbody>
</table>
### Complex type

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSONCollection</td>
<td>Represents a collection of Entities.</td>
</tr>
</tbody>
</table>

#### 2.2.4.1 Range

Represents a range of cells in a worksheet.

#### 2.2.4.1.1 Range XML Value

If Range value is used in response XML, it MUST conform to the following XML schema.

```xml
<xs:complexType name="Range">
    <xs:element name="row" type="Row" minOccurs="0" maxOccurs="unbounded"/>
    <xs:attribute name="name" type="string" use="required"/>
</xs:complexType>
```

- **row**: Represents a row from the spreadsheet.
- **name**: Contains the name of the range

#### 2.2.4.1.2 Range JSON Value

If Range value is used in response XML, it is represented as a JSON object. It has the JSON members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>A JSON string value representing the name of the range. This member MUST be present.</td>
</tr>
<tr>
<td>rows</td>
<td>A JSON array representing a collection of row in a worksheet. This member MUST be present.</td>
</tr>
</tbody>
</table>

#### 2.2.4.2 Row

Represents a row in a worksheet.

#### 2.2.4.2.1 Row XML Value

If Row value is used in response XML, it MUST conform to the following XML schema.

```xml
<xs:complexType name="Row">
    <xs:element name="c" type="Cell" minOccurs="0" maxOccurs="unbounded"/>
</xs:complexType>
```

- **c**: Represents a cell in the worksheet. The number of c elements inside a row element MUST be the same for all row elements inside the same range element

#### 2.2.4.2.2 Row JSON Value
If **Row** value is used in response JSON text, it MUST conform to JSON array representation format. The JSON array representing the **Row** SHOULD contain **Cell** objects.

### 2.2.4.3 Cell

Represents a **cell** in a **worksheet**.

#### 2.2.4.3.1 Cell XML Value

If **Cell** value is used in response XML, it MUST conform to the following XML schema.

```xml
<xs:complexType name="Cell">
  <xs:element name="fv" xsi:type="xs:string" minOccurs="0" maxOccurs="1"/>
  <xs:element name="v" xsi:type="xs:anyType" minOccurs="0" maxOccurs="1"/>
</xs:complexType>
```

**fv**: Represents a cell formatted using a **number format** according to how the **worksheet** cell is number formatted in the workbook value in a cell. If the **v** element is present and the **xsi:type** attribute on it is set to "string" then this element MUST NOT be present. MUST NOT be present if the cell is empty.

**v**: Represents the value of the cell. The **xsi:type** attribute MUST be set to either "double" (as the default), "Boolean", "String" or "Error". If the **xsi:type** attribute is set to "Error", the value of the cell MUST be one of values defined in Error type (section 2.2.5.1). MUST be present if the cell is not empty.

#### 2.2.4.3.2 Cell JSON Value

If **Cell** value is used in response JSON text, it MUST conform to JSON object value representation format. It has the JSON members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>v</strong></td>
<td>A <strong>JSON</strong> value representing the value of the cell. The type of the JSON value is one of the intrinsic JSON types – string, number or boolean. MUST NOT be present if the cell is empty or the cell is an error.</td>
</tr>
<tr>
<td><strong>fv</strong></td>
<td>A <strong>JSON</strong> string value representing a cell formatted using a number format according to how the <strong>worksheet</strong> cell is number formatted in the workbook value in a cell. MUST NOT be present if the type of <strong>v</strong> member is <strong>JSON</strong> string.</td>
</tr>
<tr>
<td><strong>t</strong></td>
<td>A <strong>JSON</strong> string value representing the type of data in a cell. MUST be present and set to &quot;error&quot; if the cell is an error. MUST NOT be present if the cell is empty.</td>
</tr>
</tbody>
</table>

### 2.2.4.4 JSONReference

Represents a reference to an **Entity Set** in a workbook.

If **JSONReference** value is used in response JSON text, it MUST conform to JSON object value representation format. It has the JSON member listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Member</strong></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>Member Name</td>
<td>Member Value</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>baseUri</td>
<td>A JSON string value representing the URL that is used to access the Entity Set using the default Entity Set representation. The default representation is XML for Entity Sets accessed in the context of Model Entity Set. The default representation is JSON for Entity Sets accessed in the context of Session Entity Set.</td>
</tr>
<tr>
<td>jsonUri</td>
<td>A JSON string value representing the URL that is used to access the Entity Set using the JSON representation.</td>
</tr>
</tbody>
</table>

### 2.2.4.5 JSONCollection

Represents a collection of Entities.

If JSONCollection value is used in response JSON text, it MUST conform to JSON object value representation format. It has the JSON member listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>items</td>
<td>A JSON array representing a collection of objects of the same type.</td>
</tr>
</tbody>
</table>

### 2.2.5 Simple Types

The following table summarizes the set of common XML schema simple type definitions defined by this specification. XML schema simple type definitions that are specific to a particular operation are described with the operation.

<table>
<thead>
<tr>
<th>Simple type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>An enumeration of the set of cell error values that a worksheet cell can contain.</td>
</tr>
</tbody>
</table>

#### 2.2.5.1 Error

The Error type is an enumeration of the set of cell error values that a worksheet cell can contain.

This simple type is defined as follows.

```xml
<xs:simpleType name="Error">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Div0"/>
    <xs:enumeration value="Name"/>
    <xs:enumeration value="Num"/>
    <xs:enumeration value="Value"/>
    <xs:enumeration value="NotApplicable"/>
    <xs:enumeration value="Null"/>
    <xs:enumeration value="Ref"/>
  </xs:restriction>
</xs:simpleType>
```
The following table specifies the allowable values for Error:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Div0</td>
<td>Worksheet cell contains a number that is divided by 0.</td>
</tr>
<tr>
<td>NotApplicable</td>
<td>Worksheet cell contains a value not available to a formula.</td>
</tr>
<tr>
<td>Name</td>
<td>Worksheet cell contains text not recognized by a formula.</td>
</tr>
<tr>
<td>Null</td>
<td>Worksheet cell contains an empty intersection of two areas.</td>
</tr>
<tr>
<td>Num</td>
<td>Worksheet cell contains invalid numeric values in a formula.</td>
</tr>
<tr>
<td>Ref</td>
<td>Worksheet cell contains a reference that is not valid.</td>
</tr>
<tr>
<td>Value</td>
<td>Worksheet cell contains a formula that includes worksheet cells that contain different data types.</td>
</tr>
</tbody>
</table>

2.2.6 Attributes

This specification does not define any common XML schema attribute definitions.

2.2.7 Groups

This specification does not define any common XML schema group definitions.

2.2.8 Attribute Groups

This specification does not define any common XML schema attribute group definitions.

2.2.9 Common Data Structures

This specification does not define any common XML schema data structures.
3 Protocol Details

3.1 Server Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The relationship between the Excel Services REST Protocol data model and the Entity Data Model defined in [MC-CSDL] are described as follows. The mapping between Entity Data Model concepts and those of the AtomPub specification is described in [MS-ODATA] section 2.2.1. The described relationship is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model, as long as their external behavior is consistent with the behavior described in [MS-ODATA] and [MC-CSDL].

<table>
<thead>
<tr>
<th>Spreadsheet data structure</th>
<th>Description</th>
<th>Entity Data Model term</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Information about the spreadsheet.</td>
<td>EntitySet</td>
</tr>
<tr>
<td>Session</td>
<td>Information about a view session to be created on the protocol server.</td>
<td>EntitySet</td>
</tr>
<tr>
<td>Ranges</td>
<td>The list of ranges that are exposed by the spreadsheet.</td>
<td>EntitySet</td>
</tr>
<tr>
<td>Charts</td>
<td>The list of charts that are exposed by the spreadsheet.</td>
<td>EntitySet</td>
</tr>
<tr>
<td>Tables</td>
<td>The list of tables that are exposed by the spreadsheet.</td>
<td>EntitySet</td>
</tr>
<tr>
<td>PivotTables</td>
<td>The list of pivot tables that are exposed by the spreadsheet.</td>
<td>EntitySet</td>
</tr>
<tr>
<td>Range</td>
<td>A range from the spreadsheet.</td>
<td>Entity</td>
</tr>
<tr>
<td>Table</td>
<td>A table from the spreadsheet.</td>
<td>Entity</td>
</tr>
<tr>
<td>PivotTable</td>
<td>A PivotTable from the spreadsheet.</td>
<td>Entity</td>
</tr>
<tr>
<td>Chart</td>
<td>A chart from the spreadsheet.</td>
<td>None. Returned as a binary response to the request.</td>
</tr>
<tr>
<td>OData</td>
<td>The OData service document, as specified in [MS-ODATA].</td>
<td>None. This is the OData service document, as specified in [MS-ODATA].</td>
</tr>
<tr>
<td>Sessions</td>
<td>The entry point for creating a view session on the protocol server.</td>
<td>EntitySet</td>
</tr>
<tr>
<td>Slicers</td>
<td>The list of slicers that are exposed by the spreadsheet.</td>
<td>EntitySet</td>
</tr>
<tr>
<td>Slicer</td>
<td>A slicer from the spreadsheet.</td>
<td>Entity</td>
</tr>
<tr>
<td>Spreadsheet data structure</td>
<td>Description</td>
<td>Entity Data Model term</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Timelines</td>
<td>The list of timelines that are exposed by the spreadsheet.</td>
<td>EntitySet</td>
</tr>
<tr>
<td>Timeline</td>
<td>A timeline from the spreadsheet.</td>
<td>Entity</td>
</tr>
<tr>
<td>_unsupported_InteractiveReports</td>
<td>The list of interactive reports that are exposed by the workbook.</td>
<td>EntitySet</td>
</tr>
<tr>
<td>Store</td>
<td>The Store data of interactive reports exposed by the workbook.</td>
<td>Entity</td>
</tr>
<tr>
<td>ConnectionInfo</td>
<td>The ConnectionInfo data of interactive reports exposed by the workbook.</td>
<td>Entity</td>
</tr>
<tr>
<td>State</td>
<td>The sort operations on PivotTables that are exposed by the workbook.</td>
<td>Entity</td>
</tr>
</tbody>
</table>

When the $format (as specified in [MS-ODATA] section 2.2.3.6.1.5) Uniform Resource Locator (URL) parameter is specified, it overrides the logic specified in [MS-ODATA] with relation to the ACCEPT HTTP header. When there are multiple response formats that satisfy the ACCEPT HTTP header, if one of the content types corresponds to the default format for the EntityType or EntitySet, that is the one that will be returned. If none of the content types specified in the ACCEPT HTTP header is the default one, the protocol server MUST return the supported format with the highest priority in the ACCEPT HTTP header.

Requests can contain a URL parameter where the key starts with "Ranges(" and ends with ")" (case insensitive), between these two strings, the key will contain a string that refers to a range. When such a parameter is passed, the protocol server MUST place the value associated with the parameter in the spreadsheet range specified by the parameter and recalculate the spreadsheet before returning the result from the same spreadsheet. The range specified in the parameter MUST correspond to a single cell in the spreadsheet.

Model and Session Entities Sets are also used as contexts for other entities in the workbook. An entity is said to be "in the context" of Model or Session if the URL path used to access the entities is respectively using the /Model or /Session URL path part.

Unless specified otherwise, all access to the above entities MUST use HTTP GET verb. Whenever PUT or DELETE verbs are specified, calling these verbs needs to be implemented through the usage of HTTP tunneling technique. Such technique requires that HTTP POST verb is used with X-HTTP-Method request header with the value of "PUT" or "DELETE" respectively.

### 3.1.1.1 Model Entity Set

The Model entity set represents the spreadsheet as a whole.

The protocol server will return a value as follows:

- If a URL parameter with a key of "$format" and value of "workbook" is passed, or the ACCEPT HTTP header, as specified in [RFC2616], fits one of the following content types: "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet", "application/vnd.ms-excel.sheet.binary.macroEnabled.12" or "application/vnd.ms-excel.sheet.macroEnabled.12", then the protocol server will return a workbook file.
If a URL parameter with a key of "$format" and value of "json" is passed, accessing Model Entity Set will result in JSON text returned. This JSON text represents the list of other Entity Sets supported by the spreadsheet. The following table lists the JSON members.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranges</td>
<td>A JSONReference object representing ranges in the worksheet and referencing the corresponding Ranges Entity Set.</td>
</tr>
<tr>
<td>Charts</td>
<td>A JSONReference object representing charts in the worksheet and referencing the corresponding Charts Entity Set.</td>
</tr>
<tr>
<td>Tables</td>
<td>A JSONReference object representing tables in the worksheet and referencing the corresponding Tables Entity Set.</td>
</tr>
<tr>
<td>pivotTables</td>
<td>A JSONReference object representing PivotTables in the worksheet and referencing the corresponding PivotTables Entity Set.</td>
</tr>
</tbody>
</table>

Otherwise, accessing The Model Entity Set as an entity set as specified in [MS-ODATA] will result in the list of other entity sets supported by the spreadsheet. The protocol server returns an EntitySet containing 4 entity sets, as follows:

- An entity set with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, with the term attribute set to "ExcelServices.Ranges" which represents the ranges in the workbook.
- An entity set with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, with the term attribute set to "ExcelServices.Charts" which represents the charts in the workbook.
- An entity set with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, with the term attribute set to "ExcelServices.Tables" which represents the tables in the workbook.
- An entity set with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, with the term attribute set to "ExcelServices.PivotTables" which represents the PivotTables in the workbook.

### 3.1.1.2 Session Entity Set

The Session entity set represents a view session that was created as described in section 3.1.1.10. All requests to a Session entity set, or other entities accessed in the context of Session, MUST specify a URL parameter "$sid" with the value set to the session identifier returned by the protocol server for a Sessions request.

When accessing the entity using HTTP GET verb, the protocol server will return a value as follows:

- If a URL parameter with a key of "$format" and value of "json" is passed, accessing Session Entity Set will result in JSON text returned. This JSON text represents the list of other Entity Sets supported by the spreadsheet. It has the JSON members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranges</td>
<td>A JSONReference object representing ranges in the worksheet and referencing the corresponding Ranges Entity Set.</td>
</tr>
<tr>
<td>Member Name</td>
<td>Member Value</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Charts</td>
<td>A JSONReference object representing charts in the worksheet and referencing the corresponding Charts Entity Set.</td>
</tr>
<tr>
<td>Tables</td>
<td>A JSONReference object representing tables in the worksheet and referencing the corresponding Tables Entity Set.</td>
</tr>
<tr>
<td>pivotTables</td>
<td>A JSONReference object representing PivotTables in the worksheet and referencing the corresponding PivotTables Entity Set.</td>
</tr>
<tr>
<td>Slicers</td>
<td>A JSONReference object representing slicers in the worksheet and referencing the corresponding Slicers Entity Set.</td>
</tr>
<tr>
<td>Timelines</td>
<td>A JSONReference object representing timelines in the worksheet and referencing the corresponding Timelines Entity Set.</td>
</tr>
<tr>
<td>_unsupported_InteractiveReports</td>
<td>A JSONReference object representing interactive reports in the workbook and referencing the corresponding _unsupported_InteractiveReports Entity Set.</td>
</tr>
</tbody>
</table>

- Otherwise, accessing The Session Entity Set as an entity set as specified in [MS-ODATA] will result in the list of other entity sets supported by the spreadsheet. The protocol server returns an EntitySet containing 4 entity sets, as follows:
  - An entity set with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, with the term attribute set to "ExcelServices.Ranges" which represents the ranges in the workbook.
  - An entity set with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, with the term attribute set to "ExcelServices.Charts" which represents the charts in the workbook.
  - An entity set with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, with the term attribute set to "ExcelServices.Tables" which represents the tables in the workbook.
  - An entity set with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, with the term attribute set to "ExcelServices.PivotTables" which represents the PivotTables in the workbook
  - An entity set with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, with the term attribute set to "ExcelServices.Slicers" which represents the slicers in the workbook.
  - An entity set with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, with the term attribute set to "ExcelServices.Timelines" which represents the timelines in the workbook.

When accessing the entity using HTTP DELETE verb, the protocol server will close the view session and all subsequent attempts to access the same session will fail.
3.1.1.3 Ranges Entity Set

Ranges in the spreadsheet are represented as an EntitySet, which contains Entities of a single EntityType as specified in section 3.1.1.4. The Ranges entity set represents the list of ranges exposed by the spreadsheet. The protocol server will return a value as follows:

- If a URL parameter with a key of "$format" and value of "atom" is passed, or is accessed in the context of Model Entity Set then the protocol server returns an entity with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, and the term attribute set to "ExcelServices.Range" for every published item in the workbook that represents a named range.

- If a URL parameter with a key of "$format" and value of "json" is passed, or is accessed in the context of Session Entity Set then the protocol server returns JSON text containing JSONCollection as defined in section 2.2.4.5. The JSONCollection represents a list of entities for every published item in the workbook that represents a named range. Each entry in the JSONCollection has members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>A JSON string value representing the name of a named range.</td>
</tr>
</tbody>
</table>

3.1.1.4 Range Entity

The Range entity represents a range in the spreadsheet.

The protocol server will return a value as follows:

- If a URL parameter with a key of "$format" and value of "html" is passed, or the ACCEPT HTTP header, as specified in [RFC2616] fits the content type: "text/html" then the protocol server will return an HTML fragment corresponding to the requested range.

- Accessing the Range Entity as an entity as specified in [MS-ODATA] (or with a URL parameter with a key of "$format" and value of "atom") will result in a single Entity Type containing the complex type Range as defined in section 2.2.4.1.1 in a content element with a type attribute set to "application/xml", as specified in [MS-ODATA].

- Accessing the Range Entity with a URL parameter with a key of "$format" and value of "json" will result in a JSON text containing the complex type Range as defined in section 2.2.4.1.2.

- Accessing the Range Entity with a URL parameter with a key of "$format" and value of "image" will result in a PNG image of the requested range. Also, the protocol server supports cropping and scaling the resulting image when optional URL parameters are specified. These parameters are described in the following table:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$cropw</td>
<td>An unsigned integer representing the width in pixels used to limit the number of columns of the range of the returned image. The cumulative pixel width of all visible columns in the requested range will be compared with $cropw. If the cumulative width of a column is equal or exceeds $cropw, then that column will be last one rendered into the returned image.</td>
</tr>
<tr>
<td>$croph</td>
<td>An unsigned integer representing the height in pixels used to limit the number of rows of the range of the</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>returned image. The cumulative pixel height of all visible columns in the requested range will be compared with $croph$. If the cumulative height of a column is equal or exceeds $croph$, then that row will be last one rendered into the returned image.</td>
</tr>
<tr>
<td>$width$</td>
<td>An unsigned integer representing the width in pixels the returned image needs to scale to. The protocol server only provides scaling down and preserves the aspect ratio of the original image and that can cause that the width of the resulting image is smaller than $width$.</td>
</tr>
<tr>
<td>$height$</td>
<td>An unsigned integer representing the height in pixels the returned image needs to scale to. The protocol server only provides scaling down and preserves the aspect ratio of the original image and that can cause that the height of the resulting image is smaller than $height$.</td>
</tr>
</tbody>
</table>

### 3.1.1.5 Tables Entity Set

Tables in the spreadsheet are represented as an EntitySet, which contains Entities of a single EntityType as specified in section 3.1.1.6. The Tables entity set represents the list of tables exposed by the spreadsheet. The protocol server will return a value as follows:

- If a URL parameter with a key "$format" and value of "atom" is passed, or is accessed in the context of Model Entity Set then the protocol server returns an entity with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, and the term attribute set to "ExcelServices.Table" for every published item in the workbook that represents a list.

- If a URL parameter with a key of "$format" and value of "json" is passed, or is accessed in the context of Session Entity Set then the protocol server returns JSON text containing a JSONCollection as defined in section 2.2.4.5. The JSONCollection represents a list of entities for every published item in the workbook that represents a list. Each entry in the JSONCollection has members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>A JSON string value representing the name of a list.</td>
</tr>
</tbody>
</table>

### 3.1.1.6 Table Entity

The Table entity represents a table in the spreadsheet.

The protocol entity server will return a value as follows:

- If a URL parameter with a key of "$format" and value of "html" is passed, or the ACCEPT HTTP header (as specified in [RFC2616]) fits the content type: "text/html", the protocol server will return an HTML fragment representing the requested table.

- Accessing the Table Entity as an entity as specified in [MS-ODATA] (or with a URL parameter with a key of "$format" and value of "atom") will result in a single Entity containing the complex type.
Range as defined in section 2.2.4.1 in a content element with a type attribute set to "application/xml", as specified in [MS-ODATA].

- Accessing the Table Entity with a URL parameter with a key of "$format" and value of "json" will result in a JSON text containing the complex type Range as defined in section 2.2.4.1.

### 3.1.1.7 PivotTables Entity Set

PivotTables in the spreadsheet are represented as an EntitySet, which contains Entities of a single EntityType as specified in section 3.1.1.8. The PivotTables entity set represents the list of PivotTables exposed by the spreadsheet. The protocol server will return a value as follows:

- If a URL parameter with a key "$format" and value of "atom" is passed, or is accessed in the context of Model Entity Set then the protocol server returns an entity with the category element present (as specified in [MS-ODATA] section 2.2.6.2.1) and the term attribute set to "ExcelServices.PivotTable" for every published item in the workbook that represents a PivotTable.

- If a URL parameter with a key of "$format" and value of "json" is passed, or is accessed in the context of Session Entity Set then the protocol server returns JSON text containing JSONCollection as defined in section 2.2.4.5. The JSONCollection represents a list of entities for every published item in the workbook that represents a PivotTable. Each entry in the JSONCollection has members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A JSON string value representing the name of a PivotTable.</td>
</tr>
</tbody>
</table>

### 3.1.1.8 PivotTable Entity

The PivotTable entity represents a PivotTable in the spreadsheet. The protocol server will return a value as follows:

- If a URL parameter with a key of "$format" and value of "html" is passed, or the ACCEPT HTTP header, as specified in [RFC2616] fits the content type: "text/html" then the protocol server will return an HTML fragment representing the requested PivotTable.

- Accessing the PivotTable Entity as an entity as specified in [MS-ODATA] (or with a URL parameter with a key of "$format" and value of "atom") will result in a single Entity Type containing the complex type Range as defined in section 2.2.4.1.1 in a content element with a type attribute set to "application/xml", as specified in [MS-ODATA].

- Accessing the PivotTable Entity with a URL parameter with a key of "$format" and value of "json" will result in a JSON text containing the complex type Range as defined in section 2.2.4.1.

### 3.1.1.9 Charts Entity Set

Charts in the spreadsheet are represented as an EntitySet. The protocol server will return a value as follows:

- If a URL parameter with a key "$format" and value of "json" is passed, or is accessed in the context of Model Entity Set then the protocol server returns an entry element with the category element present, as specified in [MS-ODATA] section 2.2.6.2.1, and the term attribute set to "ExcelServices.Chart" for every published item in the workbook that represents a chart. Each
entry element has a content element with the type attribute set to "image/png" and the src attribute set to a URL that will return the image representing the requested chart.

- If a URL parameter with a key of "$format" and value of "json" is passed, or is accessed in the context of Session Entity Set then the protocol server returns JSON text containing JSONCollection as defined in section 2.2.4.5. The JSONCollection represents a list of entities for every published item in the workbook that represents a Chart. Each entry in the JSONCollection has members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>A JSON string value representing the name of a Chart.</td>
</tr>
</tbody>
</table>

When requesting an individual Chart a PNG image is returned. This image can be scaled from its original size by optionally specifying width or height or both. The scaling maintains the original aspect ratio, except when both width and height is specified. In case when both width and height is specified, the protocol server MUST fit the image into the rectangular area defined by those constraints. The protocol server recognizes URL parameters listed in the following table.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$width</td>
<td>An unsigned integer representing, in pixels, the desired width of the returned image.</td>
</tr>
<tr>
<td>$height</td>
<td>An unsigned integer representing, in pixels, the desired height of the returned image.</td>
</tr>
</tbody>
</table>

### 3.1.1.10 Sessions Entity Set

A view session is opened on the protocol server when a request is made to the Sessions Entity Set using a POST. The protocol server recognizes URL parameters listed in the following table.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$uc</td>
<td>A language tag, as specified in [RFC1766], Section 2. Specifies the UI culture.</td>
</tr>
<tr>
<td>$dc</td>
<td>A language tag, as specified in [RFC1766], Section 2. Specifies the data culture.</td>
</tr>
</tbody>
</table>

- If a URL parameter with a key of "$format" and value of "atom" is passed the content of the Entity Set result is single entity that represents the session. Making a request to the entity URL will return a result as described in section 3.1.1.1.
- If a URL parameter with a key of "$format" and value of "json" is passed, or the ACCEPT HTTP header, as specified in [RFC2616] fits the content type: "application/json" then the protocol server will return a JSON text. It has the JSON members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionId</td>
<td>A JSON string value representing the session id of the view session created by protocol server.</td>
</tr>
<tr>
<td>baseUri</td>
<td>A URL representing the Session Entity.</td>
</tr>
<tr>
<td>Member Name</td>
<td>Member Value</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>itemViewerUri</td>
<td>A URL representing an interactive item view for the view session created by the protocol server.</td>
</tr>
</tbody>
</table>

### 3.1.1.11 Slicers

Slicers in the spreadsheet are represented as an **EntitySet**, which contains **Entities** of a single **EntityType** as specified in section 3.1.1.12. The Slicers entity set represents the list of Slicers exposed by the spreadsheet. Slicers can only be accessed in the context of Session Entity Set. If a URL parameter with a key of "$format" and value of "json" is passed, or the ACCEPT HTTP header, as specified in [RFC2616] fits the content type: "application/json" the protocol server returns **JSON** text containing a **JSONCollection** as defined in section 2.2.4.5. The **JSONCollection** represents a list of entities for every published item in the workbook that represents a Slicer. Each entry in the **JSONCollection** has members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>A <strong>JSON</strong> string value representing the name of a Slicer.</td>
</tr>
<tr>
<td>caption</td>
<td>A <strong>JSON</strong> string value representing the caption of a Slicer.</td>
</tr>
<tr>
<td>updatesAllowed</td>
<td>A <strong>JSON</strong> boolean value representing whether the protocol server allows modifying the state of the Slicer as defined in section 3.1.1.12.</td>
</tr>
<tr>
<td>connectedPublishedObjects[]</td>
<td>A <strong>JSON</strong> array containing <strong>JSON</strong> string values representing the names of published items whose state would be affected if the selection state of the Slicer is modified.</td>
</tr>
</tbody>
</table>

### 3.1.1.12 Slicer

The **Slicer** entity represents a Slicer in the spreadsheet. The protocol server supports retrieving the Slicer through HTTP GET verb as well as applying changes to the current state of the slicer through HTTP POST and PUT verbs.

When a request is made using a GET:

- If a URL parameter with a key of "$format" and value of "json" is passed, or the ACCEPT HTTP header, as specified in [RFC2616] fits the content type: "application/json" then the protocol server will return a **JSON** text representing the items in the requested Slicer. The protocol server recognizes URL parameters listed in the following table.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$skip</td>
<td>An unsigned integer representing the index of the first slicer item that is to be returned by the protocol server. If not specified, the default value is 0.</td>
</tr>
<tr>
<td>$top</td>
<td>An unsigned integer representing the maximum number of slicer items that are to be returned by the protocol server. If not specified, the default value is 100. The maximum number of slicer items returned is</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>totalItemCount</td>
<td>A JSON number value representing the total number of items in a Slicer. MUST be present.</td>
</tr>
<tr>
<td>ItemOffset</td>
<td>A JSON number value representing the 0-based index of the first item returned in the items array. MUST be present.</td>
</tr>
<tr>
<td>hasEffectiveSelection</td>
<td>A JSON boolean value representing some of the slicer items are in unselected state. MUST be present.</td>
</tr>
<tr>
<td>NextOffset</td>
<td>A JSON number.</td>
</tr>
<tr>
<td>Items[]</td>
<td>A JSON array containing data about slicer items.</td>
</tr>
<tr>
<td>Items[].displayName</td>
<td>The JSON string value representing the display name of the slicer item.</td>
</tr>
<tr>
<td>Items[].uniqueName</td>
<td>The JSON string value representing the unique name of the slicer item. This name MUST be used when programmatic referencing the slicer item in the slicer update operation.</td>
</tr>
<tr>
<td>items[].unselected</td>
<td>The JSON boolean value representing that this item has been unselected. Present only when true.</td>
</tr>
<tr>
<td>items[].noData</td>
<td>The JSON boolean value representing that this item has no data. Present only when true.</td>
</tr>
</tbody>
</table>

When a request is made using PUT or POST:

If a URL parameter with a key of "$format" and value of "json" is passed, or the ACCEPT HTTP header, as specified in [RFC2616] fits the content type: "application/json" then the protocol server interprets the body of the HTTP request as a JSON text. The JSON object expected by protocol server has members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameters</td>
<td>A JSON nested object representing the operation parameters passed to the protocol server. MUST be present.</td>
</tr>
</tbody>
</table>
| parameters.selectionMode | A JSON string value representing the way the slicer state needs to be modified. Supports the following string values:  
  - cumulative  
  - clearFilter  
  - selectedOnly (this is the default) |
| parameters.selected[] | A JSON array of JSON string values representing unique names of slicer items that are to be selected. MUST be ignored if the value of selectionMode is set |
3.1.1.13 Timelines

Timelines in the spreadsheet are represented as an **EntitySet**, which contains **Entities** of a single **EntityType** as specified in section 3.1.1.14. The Timelines entity set represents the list of timelines exposed by the spreadsheet. The content of the Entity Set result is a JSON formatted array of **Entities** for every published item in the workbook that represents a Timeline. Timelines can only be accessed in the context of Session Entity Set. If a URL parameter with a key of "$format" and value of "json" is passed, or the ACCEPT HTTP header, as specified in [RFC2616] fits the content type: "application/json" the protocol server returns **JSON** text containing a **JSONCollection** as defined in section 2.2.4.5. The **JSONCollection** represents a list of entities for every **published item** in the workbook that represents a Timeline. Each entry in the JSONCollection has members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameters.unselected[]</td>
<td>A JSON array of JSON string values representing unique names of slicer items that are not to be selected. MUST be ignored if the value of <strong>selectionMode</strong> is set to <strong>clearFilter</strong> or <strong>selectedOnly</strong>.</td>
</tr>
</tbody>
</table>

3.1.1.14 Timeline

The **Timeline** entity represents a timeline in the spreadsheet.

The protocol server supports retrieving the timeline through HTTP GET verb as well as applying changes to the current state of the timeline through HTTP POST and PUT verbs.

When a request is made using a GET:

- If a URL parameter with a key of "$format" and value of "json" is passed, or the ACCEPT HTTP header, as specified in [RFC2616] fits the content type: "application/json" then the protocol server will return a **JSON** text representing the items in the requested Timeline. The JSON object returned by protocol server has members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>minDate</td>
<td>JSON string value representing the minimum date the timeline is defined on (YYYY-MM-DD format). MUST be present.</td>
</tr>
<tr>
<td>Member Name</td>
<td>Member Value</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>maxDate</td>
<td>JSON string value representing the maximum date the timeline is defined on (YYYY-MM-DD format). MUST be present.</td>
</tr>
<tr>
<td>selectionBeginDate</td>
<td>JSON string value representing the begin date of timeline selected range (YYYY-MM-DD format). MUST NOT be present if selectionState value is equal to &quot;clearFilter&quot;.</td>
</tr>
<tr>
<td>selectionEndDate</td>
<td>JSON string value representing the end date of timeline selected range (YYYY-MM-DD format). MUST NOT be present if selectionState value is equal to &quot;clearFilter&quot;.</td>
</tr>
<tr>
<td>displayLevel</td>
<td>A JSON string value representing timeline’s display level. MUST be present and be set to one of the following string values: years, quarters, months, days.</td>
</tr>
<tr>
<td>selectionState</td>
<td>A JSON string value representing timeline’s selection state. MUST be present and be set to one of the following string values: between, clearFilter, unsupported.</td>
</tr>
<tr>
<td>displayLabel</td>
<td>The JSON string value representing the descriptive label of the timeline.</td>
</tr>
</tbody>
</table>

When a request is made using PUT or POST:

If a URL parameter with a key of "$format" and value of "json" is passed, or the ACCEPT HTTP header, as specified in [RFC2616] fits the content type: "application/json" then the protocol server interprets the body of the HTTP request as a JSON text. The JSON object expected by protocol server has members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>A JSON nested object representing the operation parameters passed to the protocol server. MUST be present.</td>
</tr>
<tr>
<td>parameters.selectionType</td>
<td>A JSON string value representing the requested selection Type. MUST be present and be set to one of the following string values: between, clearFilter</td>
</tr>
</tbody>
</table>
Member Name | Member Value
---|---
parameters.selectionBeginDate | A JSON string value representing the selection begin date. Ignored if the value of selectionType is set to clearFilter.
parameters.selectionEndDate | A JSON string value representing the selection end date. Ignored if the value of selectionType is set to clearFilter.

3.1.1.15 _unsupported_InteractiveReports

_unsupported_InteractiveReports in the workbook are represented as an EntitySet, which contains items representing each interactive report exposed in the workbook. The content of the Entity Set result is a JSON formatted array of items for every published interactive report in the workbook. _unsupported_InteractiveReports can only be accessed in the context of a Session Entity Set. If a URL parameter with a key of "$format" and value of "json" is passed, or the ACCEPT HTTP header, as specified in [RFC2616], fits the content type: "application/json", then the protocol server returns JSON text containing a JSONCollection as defined in section 2.2.4.5. The JSONCollection represents a list of interactive reports for every published item in the workbook that represents an interactive report. Each entry in the JSONCollection has members listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
</table>
sheetName | A JSON string value representing the name (spreadsheet name) of an interactive report. |
slideId | A JSON string value representing the identifier of an interactive report slide. |

3.1.1.16 Store

Store contains data for interactive reports exposed in the workbook. The content result is a JSON formatted object consisting of properties for every published interactive report in the workbook. Store can only be accessed in the context of _unsupported_InteractiveReports as a sub-resource. If a URL parameter with a key of "$format" and value of "json" is passed, or the ACCEPT HTTP header, as specified in [RFC2616], fits the content type: "application/json", then the protocol server returns a JSON object. The JSON object contains properties for interactive reports in the workbook. The JSON object has properties listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
</table>
correlationId | A GUID representing the correlation identifier. |
dataConnectionsInfo | A JSON object containing information about data connections in the workbook. |
dataConnectionsInfo.dataConnectionInfo[] | A JSON array of data connection information. |
dataConnectionsInfo.dataConnectionInfo[].connectionName | A JSON string value representing the data connection’s connection name. |
### Member Name | Member Value
--- | ---
dataConnectionsInfo.dataConnectionInfo[].connectionString | A JSON string value representing the data connection’s connection string.
dataConnectionsInfo.dataConnectionInfo[].dataSourceName | A JSON string value representing the data connection’s data source name.
dataConnectionsInfo.dataConnectionInfo[].isEmbedded | A JSON boolean value representing whether the data connection is embedded.
dataConnectionsInfo.dataConnectionInfo[].refreshTimestamp | A JSON string timestamp in ISO 8601 format representing the data connection’s last refresh date and time.
dataConnectionsInfo.dataConnectionInfo[].securityToken | A JSON string value representing the data connection’s security token.
dataConnectionsInfo.dataConnectionInfo[].slidesIds[] | A JSON array of slide identifiers associated with this data connection.
dataConnectionsInfo.dataConnectionInfo[].sourceIndex | A JSON string value representing the data connection’s source index.
dataConnectionsInfo.dataConnectionInfo[].state | A JSON string value representing the data connection state. MUST be set to one of the following string values:
   - Failed
   - Streaming
   - Succeeded
dataConnectionsInfo.dataSources | A JSON string value representing the data sources in the workbook.
reportData | A JSON string value representing the report data as described in [MS-DPRDL] and as specified in [MS-RDL].

### 3.1.1.17 ConnectionInfo

ConnectionInfo contains data for interactive reports exposed in the workbook. The difference between ConnectionInfo and Store is that ConnectionInfo does not contain reportData for optimization purposes. The content result is a JSON formatted object consisting of properties for every published interactive report in the workbook. ConnectionInfo can only be accessed in the context of _unsupported_InteractiveReports as a sub-resource. If a URL parameter with a key of "$format" and value of "json" is passed, or the ACCEPT HTTP header, as specified in [RFC2616], fits the content type: "application/json", then the protocol server returns a JSON object. The JSON object contains properties for interactive reports in the workbook. The JSON object has properties listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
</table>
correlationId    | A GUID representing the correlation identifier.                             |
dataConnectionsInfo | A JSON object containing information about data                             |
<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataConnectionsInfo.dataConnectionInfo[]</td>
<td>A JSON array of data connection information.</td>
</tr>
<tr>
<td>dataConnectionsInfo.dataConnectionInfo[].</td>
<td>A JSON string value representing the data connection’s connection name.</td>
</tr>
<tr>
<td>connectionName</td>
<td></td>
</tr>
<tr>
<td>dataConnectionsInfo.dataConnectionInfo[].</td>
<td>A JSON string value representing the data connection’s connection string.</td>
</tr>
<tr>
<td>connectionString</td>
<td></td>
</tr>
<tr>
<td>dataConnectionsInfo.dataConnectionInfo[].</td>
<td>A JSON string value representing the data connection’s data source name.</td>
</tr>
<tr>
<td>dataSourceName</td>
<td></td>
</tr>
<tr>
<td>dataConnectionsInfo.dataConnectionInfo[].</td>
<td>A JSON boolean value representing whether the data connection is embedded.</td>
</tr>
<tr>
<td>isEmbedded</td>
<td></td>
</tr>
<tr>
<td>dataConnectionsInfo.dataConnectionInfo[].</td>
<td>A JSON string timestamp in ISO 8601 format</td>
</tr>
<tr>
<td>refreshTimestamp</td>
<td>representing the data connection’s last refresh date and time.</td>
</tr>
<tr>
<td>dataConnectionsInfo.dataConnectionInfo[].</td>
<td>A JSON string value representing the data connection’s security token.</td>
</tr>
<tr>
<td>securityToken</td>
<td></td>
</tr>
<tr>
<td>dataConnectionsInfo.dataConnectionInfo[].</td>
<td>A JSON array of slide Ids associated with this data connection.</td>
</tr>
<tr>
<td>slidesIds[]</td>
<td></td>
</tr>
<tr>
<td>dataConnectionsInfo.dataConnectionInfo[].</td>
<td>A JSON string value representing the data connection’s source index.</td>
</tr>
<tr>
<td>sourceIndex</td>
<td></td>
</tr>
<tr>
<td>dataConnectionsInfo.dataConnectionInfo[].</td>
<td>A JSON string value representing the data connection state. MUST be set to one of the following string values:</td>
</tr>
<tr>
<td>state</td>
<td>• Failed</td>
</tr>
<tr>
<td></td>
<td>• Streaming</td>
</tr>
<tr>
<td></td>
<td>• Succeeded</td>
</tr>
<tr>
<td>dataConnectionsInfo.dataSources</td>
<td>A JSON string value representing the data sources in the workbook.</td>
</tr>
</tbody>
</table>

### 3.1.1.18 State

State contains data for **PivotTables** stored in the workbook. The content result is a **JSON** formatted object representing a PivotTable in the workbook. If a URL parameter with a key of "$format" and value of "json" is passed, or the ACCEPT HTTP header, as specified in [RFC2616], fits the content type "application/json", then the protocol server returns a JSON object. The JSON object represents a PivotTable in the workbook. The JSON object has properties listed in the following table.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Member Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A JSON string value representing the current rest resource.</td>
</tr>
<tr>
<td>Parent</td>
<td>A JSON string value representing parent level rest</td>
</tr>
<tr>
<td>Member Name</td>
<td>Member Value</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Index</td>
<td>A JSON string value representing rest resource index</td>
</tr>
<tr>
<td>ValidatedParameters</td>
<td>A JSON string value representing the name collection of validated parameters.</td>
</tr>
<tr>
<td>VerifyMethodsCalled</td>
<td>A JSON boolean value representing whether verified methods are invoked.</td>
</tr>
</tbody>
</table>

### 3.1.2 Timers

None.

### 3.1.3 Initialization

None.

### 3.1.4 Message Processing Events and Sequencing Rules

None.

### 3.1.5 Timer Events

None.

### 3.1.6 Other Local Events

None.
4 Protocol Examples

The examples in this section use the following sample data and configuration:

- A workbook with the title Sales.xlsx exists on the site. The workbook contains the following resources:
  - A table named "SalesProjections".
  - A chart named "CommissionChart".
  - A range named "CommissionRate_Bikes".

4.1 Retrieve the Entity Container for the Content of a Workbook

The following example illustrates the exchange of messages required for a client to retrieve the entity container for the content of the Sales.xlsx workbook.

Request

GET /_vti_bin/ExcelRest.aspx/Cycles/Sales.xlsx/Model/ HTTP/1.1
Accept: application/atom+xml
Host: www.contoso.com

Response

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: application/atom+xml; charset=utf-8
Server: Microsoft-IIS/7.5
SPRequestGuid: 63c0d5ca-8b6e-43bf-92a5-d6ed45210f5b
X-SharePointHealthScore: 4
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
MicrosoftSharePointTeamServices: 14.0.0.4732
Date: Wed, 20 Jan 2010 19:48:13 GMT
Content-Length: 3158

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:x="http://schemas.microsoft.com/office/2008/07/excelservices/rest"
  <title type="text">Model</title>
  <id>http://www.contoso.com/_vti_bin/ExcelRest.aspx/Cycles/Sales.xlsx/Model/</id>
  <updated>2010-01-20T19:48:13Z</updated>
  <author>
    <name />
    <link rel="self" href="http://www.contoso.com/_vti_bin/ExcelRest.aspx/Cycles/Sales.xlsx/Model/" title="Model" />
  </author>
  <entry>
    <title>Ranges</title>
    <id>http://www.contoso.com/_vti_bin/ExcelRest.aspx/Cycles/Sales.xlsx/Model/Ranges</id>
    <updated>2010-01-20T19:48:13Z</updated>
    <author>
      <name />
4.2 Retrieve a Table from a Workbook as HTML

The following example illustrates the exchange of messages required for a client to retrieve the HTML representation of a table called SalesProjections within the Sales.xlsx workbook.

Request
<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Phone</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samantha</td>
<td>sam@gmail</td>
<td>123456</td>
<td>new</td>
</tr>
<tr>
<td>Joe</td>
<td><a href="mailto:joe@abc.com">joe@abc.com</a></td>
<td>67890</td>
<td></td>
</tr>
<tr>
<td>Sam</td>
<td><a href="mailto:sam@xyz.com">sam@xyz.com</a></td>
<td>11111</td>
<td></td>
</tr>
</tbody>
</table>

**Total:** 3,195 entries
<table>
<thead>
<tr>
<th>Name</th>
<th>Current Price</th>
<th>Previous Price</th>
<th>Change</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albert</td>
<td>$3454</td>
<td>$2996</td>
<td>$458</td>
<td>Office</td>
</tr>
<tr>
<td>Dieter</td>
<td>$3011</td>
<td>$2780</td>
<td>$231</td>
<td>Warehouse</td>
</tr>
</tbody>
</table>

Albert's price has increased by $458 compared to his previous price. Dieter's price has increased by $231 compared to his previous price.
4.3 Retrieve a Range from a Workbook as an Atom Feed

The following example illustrates the exchange of messages required for a client to retrieve a range from the Sales.xlsx workbook as an Atom feed.

Request

GET /_vti_bin/ExcelRest.aspx/Cycles/Sales.xlsx/Model/Ranges('C17%7CE17')?$format=atom
HTTP/1.1
Accept: application/atom+xml
Host: www.contoso.com

Response

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: application/atom+xml;charset=utf-8
Server: Microsoft-IIS/7.5
SPRequestGuid: a8fecc3b-a7ef-48e5-a988-b6152207fad1
X-SharePointHealthScore: 4
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
MicrosoftSharePointTeamServices: 14.0.0.4732
Date: Wed, 20 Jan 2010 19:50:23 GMT
Content-Length: 1183

<?xml version="1.0" encoding="utf-8"?>
    xmlns:x="http://schemas.microsoft.com/office/2008/07/excelservices/rest"
    xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
    xmlns:m="http://schemas.microsoft.com/office/word/2006/main"
    <id>urn:uuid:777d2c15-853f-446b-852e-c8a5c765699d</id>
    <contentStreams/>
    <author/>
    <link rel="edit" href="/_vti_bin/ExcelRest.aspx/Cycles/Sales.xlsx/Model/Ranges('C17%7CE17')?$format=atom"/>
    <link rel="self" href="/_vti_bin/ExcelRest.aspx/Cycles/Sales.xlsx/Model/Ranges('C17%7CE17')?$format=atom"/>
    <link rel="related" href="/_vti_bin/ExcelRest.aspx/Cycles/Sales.xlsx/Model"/>
    <created>2010-01-20T19:50:23Z</created>
    <modified>2010-01-20T19:50:23Z</modified>
    <title>Retrieve a Range from a Workbook as an Atom Feed</title>
    <summary>The following example illustrates the exchange of messages required for a client to retrieve a range from the Sales.xlsx workbook as an Atom feed.</summary>
    <fullText></fullText>
    <contentBytes/>
</entry>
Retrieve a Chart from a Workbook After Setting a Value in the Workbook

The following example illustrates the exchange of messages required for a client to change the value of a range named CommissionRate_Bikes in the Sales.xlsx workbook while also retrieving a chart named CommissionChart from the same workbook. Note the chart is returned as an image. The binary image data is represented here in hexadecimal.

Request

GET /_vti_bin/ExcelRest.aspx/Cycles/Sales.xlsx/Model/Charts('CommissionChart')?$format=image&ranges('CommissionRate_Bikes')=0.03 HTTP/1.1
Accept: image/png
Host: www.contoso.com

Response

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: image/png
Server: Microsoft-IIS/7.5
SPRequestGuid: 976f5ad3-5748-4049-9471-eb2ad4035047
X-SharePointHealthScore: 4
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
MicrosoftSharePointTeamServices: 14.0.0.4732
Date: Wed, 20 Jan 2010 19:51:32 GMT
Content-Length: 9325
Retrieve the OData service document for a workbook

The following example illustrates the exchange of messages required for a client to request the OData service document.

Request

GET
http://office/15/teams/DBI/xlservices/_vti_bin/ExcelRest.aspx/Users/adean/DatesAndTimes.xlsx/OData HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Host: office

Response

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: application/xml;charset=utf-8
Server: Microsoft-IIS/7.5
X-SharePointHealthScore: 0
DataServiceVersion: 1.0,
X-AspNet-Version: 4.0.30319
SPRequestGuid: 7cab629b-4882-d026-23a5-2bd43a07d980
X-Powered-By: ASP.NET
MicrosoftSharePointTeamServices: 15.0.0.3407
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<workspace>
  <atom:title>Default</atom:title>
  <collection href="DatesTextFormated">
    <atom:title>DatesTextFormated</atom:title>
  </collection>
</workspace>
</service>
5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.
6 Appendix A: Full WSDL

None.
7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft SharePoint Server 2010
- Microsoft SharePoint Server 2013
- Microsoft SharePoint Server 2016
- Microsoft SharePoint Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.
8 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.
9 Index

A
Abstract data model
charts entity set 22
model entity set 17
PivotTable entity 22
PivotTables entity set 22
range entity 20
ranges entity set 20
server 16
sessions entity set 23
table entity 21
tables entity set 21
Applicability 9
Attribute groups 15
Attributes 15

C
Capability negotiation 9
Cell complex type 13
Change tracking 52
Charts entity set 22
Common data structures 15
Complex types 11
Cell 13
JSONCollection 14
JSONReference 13
Range 12
Row 12

D
Data model - abstract
server 16

E
Error simple type 14
Events
local - server 31
timer - server 31
Examples
overview 32
Retrieve a chart from a workbook after setting a value in the workbook 38
Retrieve a range from a workbook as an atom feed 37
Retrieve a table from a workbook as HTML 33
Retrieve the entity container for the content of a workbook 32
Retrieve the OData service document for a workbook 47

F
Fields - vendor-extensible 10
Full WSDL 50

G
Glossary 6

Groups 15

I
Implementer - security considerations 49
Index of security parameters 49
Informative references 8
Initialization
server 31
Introduction 6

J
JSONCollection complex type 14
JSONReference complex type 13

L
Local events
server 31

M
Message processing
server 31
Messages
attribute groups 15
attributes 15
Cell complex type 13
cardinal data structures 15
cardinal types 11
elements 11
enumerated 11
Error simple type 14
groups 15
JSONCollection complex type 14
JSONReference complex type 13
namespaces 11
Range complex type 12
Row complex type 12
simple types 14
syntax 11
transport 11
Model entity set 17

N
Namespaces 11
Normative references 7

O
Overview (synopsis) 8

P
Parameters - security index 49
PivotTable entity 22
PivotTables entity set 22
Preconditions 9
Prerequisites 9
Product behavior 51