

RCP - WEBAPI-010 Add Functionality to Web API Specification

Submitter Name	@ Sergio Del Rio
Submitter Organization	Templates for Business, Inc.
Submitter Email	Sergio.Del.Rio@t4bi.com
Co-submitter Name	
Co-submitter Organization	
Co-submitter Email	-

Document Name	RESO Web API v1.1 Draft
Document Version	1.1.0
Date Submitted	2017-04-24
Status	IN DRAFT
Status Change Date	2018-04-24

Synopsis

This document is the result of the Transport Workgroup effort to document how the RESO Web API specification will be altered to include all the necessary functionality required to add, update and delete data from MLS systems implementing the RESO Web API.

Rationale

To enable add, update and delete functionality to the RESO Web API, we need to:

- Document what OData HTTP methods will be added to the specification's set of supported methods.
- Document all exceptions that may be required.
- Document any functions that may be required.
- Document the warning / error payload to be returned from unsuccessful update operations.
- Add Examples for various different use-cases

Proposal

Global Changes Required

Update all references for the OData standard to Version 4.01, this will allow the use of the latest features to simplify the specification and to start embedding some rules into the Update specification.

Rework Section 1 - Proposal

As I started looking at where to add the HTTP method documentation, I realized that this entire section is no longer meaningful and has been left-over from the original proposed idea of a Web API specification. It is out of date and entirely wrong in some areas. This section should be renamed to: Section 1 – Overview.

Section 1.1 Purpose should be replaced with the following:

The RESO transport workgroup has defined an industry-wide standard for real-time access to real estate data (directly from Web and Mobile applications). The goal of this new standard is to provide a more open approach to data access using widely-adopted technology standards in use across industries, including the real estate industry. Specifically, the approach focuses on the use of the REST (REpresentational State Transfer) architectural approach documented by Roy Thomas Fielding and adopted by tens of thousands of developers worldwide.

The goal driving the move toward a RESTful standard for the real estate industry was to encourage and promote access to real estate information directly from Web, mobile, social and other HTTP-based applications. Using a RESTful transport will enable web applications to directly interact with RESO enabled data services. (note: more information on RESTful can be found here: https://en.wikipedia.org/wiki/Representational_state_transfer)

This workgroup sought to find an approach that does not deviate from either the solid foundations already employed from past RESO accomplishments or the existing technology standards that set out to solve similar problems for other industries.

The goals of this group were to:

- Honor existing data service capabilities from RETS 1[1].x

- Adopt existing standard technologies in use across industries
- Leverage existing production-ready software toolkits

As such the RESO Web API makes use of an existing standard that was designed specifically for data transport. The standard, Open Data Protocol or “OData” (<http://www.odata.org/>) serves as a set of fundamental building blocks for what the group is proposing.

The group chose OData for the following reasons:

- Well-established and robustly documented existing standard.
- Significant community adoption including “Open Government Data Initiative.”
- Well-defined functionality supports most significant RESO use cases.
- Existing open source technology implementations to support community adoption.
- Extensibility to handle specific use cases as needed.

Compatible RESO OData Transport client and server applications MUST be implemented according to the OData V4 standard specification. All further references to OData in this document refer to the OData V4 standard. Compatible server and client applications MUST send or receive data in JSON or ATOM/XML format. In keeping with OData both the client and server applications will use standard HTTP methods to perform the operations outlined by this document.

As a standards body, we will follow the OData standard and will extend, where needed, to fulfill our industry’s needs. We will not, however, deviate from the RESTful principles, standard capabilities or query syntax that is inherent to the OData standard.

The text from Section 1 – Proposal – Odata Overview should be added as section 1.2 to replace the current Section 1.2 completely. Section 1.2 should be called OData Overview.

Section 1.3 Approach should be deleted.

New Section: 2.2.3 Supported HTTP Methods

The following table illustrates the various HTTP methods that are supported by the RESO Web API and their expected functions:

HTTP Method	Functionality
GET	Search – Provides the ability to search a resource in various ways including support for a filter language.
POST	Insert – Provides the ability to insert a record into a resource. All the field data to be inserted must be in the body of the transaction. Note: This functionality can be overridden to the functionality of GET using the X-HTTP-Method-Override header.
PATCH	Update – Provides the ability to update one or more fields in an existing record for a resource.
DELETE	Delete – Provides the ability delete an existing record in a resource.

New Section 2.2.4 HTTP Methods that are Not Recommended

While we are not specifically disallowing the PUT operation, that nature of the PUT operation may cause confusion for clients in exactly what happens and how to properly update records on a server. For this reason the specification does not recommend that servers allow PUT operations for updating data. It is sufficient to allow users to use a combination of DELETE and POST to accomplish the same thing should the server allow them.

New Section 2.2.5 Preference Header Recommendations

Header	Functionality
omit=nulls	It is recommended that that servers fully support this functionality in order to reduce the outbound payload size.
omit=defaults	It is recommended that servers do not support this functionality in order to ensure that clients get important default values that are integral to the service.

New Section: 2.7 Update

Section 11 of the [OData V4](#) specification provides full details about OData service requests for updating data.

As per section 2.2.3 Supported HTTP Methods, update operations are sent to the server using one of POST, PATCH and DELETE.

These methods cover all the required operations needed to support full CRUD operations on resources. The OData specification must be followed as per Section 11.

Servers MAY refuse to perform any Update operation that is not permitted due to the permissions of the user making the request or due to any other reason. In this case, the server must return a 401 HTTP error.

Note that the specification indicates that clients can request whether created or modified resources are returned from the create, update and upsert operations using the return preference header and that if such a header is not provided, servers SHOULD return the updated entity.

New Section: 2.7.1 Creating Data

To create an entity in a collection, the client sends a POST request to that collection's URL. The POST body MUST contain a single valid entity representation as per the EDMX definition. It is not expected that the body contain any key fields or fields that are intended to be fully controlled by the server. If these are provided, the server MAY ignore the contents of these fields. These values may be altered by the server are returned to the client if the client specified the return preference header. If the server is returning data, the server MUST return the full entity.

New Section: 2.7.2 Updating Data

To update one or more fields in an entity, the client must send a PATCH request to to an entities edit URL. The PATCH body MUST contain the set of fields that are to be modified for a single valid entity representation as per the EDMX definition. If the server is returning data, the server MUST return the full entity.

New Section: 2.7.3 Deleting Data

To delete an entity, the client sends a DELETE request to an entity's edit URL.

New Section: 2.7.4 Functions and Actions

RESO OData servers may provide either functions, actions or both as per the OData specification at their discretion.

It is important to note the key difference between Functions and Actions. Functions are operations exposed by an OData service that MUST return data and MUST have no observable side effects whereas actions may have side effects when invoked and may return data. Actions must not be further composed with additional path segments.

New Section: 2.7.4.1 Mandatory Functions

At this time RESO does not require a server to support any functions.

New Section: 2.7.4.2 Mandatory Actions

The functions described in this section MUST be supported by all servers:

New Section: 2.7.4.2.1 OrderChildren

The contents of [RCP - WEBAPI-011 Child Order Action](#) will be inserted here to define the function.

New Section: 2.7.5 Additional Update Features

Servers MUST support the following update features:

1. Batch Requests - Using JSON Format
2. Change Sets - Change sets are considered important as they would allow the submission of related data in one atomic transaction. For example, a Property in addition to all it's child Room resource records could be added as a single atomic transaction.

Clients MUST support the following update features:

1. Asynchronous Requests

New Section: 2.7.6 Errors and Warnings

When an error occurs, it is expected that the server will return a comprehensive Error block as per Section 19 Error Response in the ODATA JSON Format or OData Atom Format documentation.

Each field that has an error must be identified using the target field of the details section of a more global error message. This construct also provides for additional trace and context sections allowing the server to provide even more details if necessary.

Error Message Example

```
{
  "error": {
    "code": "20100",
    "message": "Update Field Errors",
    "target": "Update",
    "details": [
      {
        "code": "30212",
        "target": "Property.ListPrice",
        "message": "List Price too Low"
      },
      {
        "code": "30327",
        "target": "Property.SalePrice",
        "message": "Sale Price too High"
      }
    ],
    "innererror": {
      "trace": [...],
      "context": {...}
    }
  }
}
```

In order to allow warnings to be returned in addition to the response payload, warnings MUST be passed back as Instance Annotations.

The only valid instance warning annotations is defined as:

RESO_Warning

The definition of this is as follows:

Warning Annotation Definition

```
<Term Name="RESO_Warning" Type="Collection(Core.MessageType)">
  <Annotation Term="Core.Description" String="A RESO Set of Warnings"
/>
</Term>
```

Each field that has an error must be identified using the target field of the details section of a more global warning message.

Warning Message Example

```
"RESO_Warning": {
  "code": "20200",
  "message": "Update Field Warnings",
```

```

    "target": "Update",
    "details": [
      {
        "code": "40212",
        "target": "Property.ListPrice",
        "message": "List Price may be too Low"
      },
      {
        "code": "40327",
        "target": "Property.SalePrice",
        "message": "Sale Price may be too High"
      }
    ]
  }
}

```

Appendix 5 - Approved RCPs

This section should be renamed to Appendix 6 - Approved RCPs

Appendix 5 - Insert, Update and Delete Examples

Appendix 5.1 - Partial Model for Examples

The partial model described below has been provided for reference and is used in all the Insert, Update and Delete examples. In the model, it should be noted that the use of the `Org.OData.Core.V1.OptimisticConcurrency` triggers the inclusion of an ETag when a service requires it to be returned with a Data modification Request or Action Request in order to apply optimistic concurrency when performing operations.

Data Model for Insert, Update and Delete Examples

```

<?xml version="1.0" encoding="utf-8"?>
<edmx:Edmx Version="4.0" xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
  <edmx:DataServices>
    <Schema Namespace="MyCompany.OData.Models" xmlns="http://docs.oasis-open.org/odata/ns/edm">
      ...
      <EntityType Name="Property">
        <Key>
          <PropertyRef Name="ListingKeyNumeric" />
        </Key>
        <Property Name="BathroomsTotalInteger" Type="Edm.Int32" />
      </EntityType>
      <Property Name="BedroomsTotal" Type="Edm.Int32" />
      <Property Name="City" Type="MyCompany.OData.Models.City" />
      <Annotation Term="MyCompany.OData.Metadata.RESO.TextMaxLength" Int="75" />
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>

```

```

        </Property>
        <Property Name="CoolingYN" Type="Edm.Boolean" />
        <Property Name="CountyOrParish" Type="MyCompany.OData.
Models.County">
            <Annotation Term="MyCompany.OData.Metadata.RESO.
TextMaxLength" Int="75" />
            </Property>
            <Property Name="HeatingYN" Type="Edm.Boolean" />
            <Property Name="ListAgentFirstName" Type="Edm.String"
MaxLength="50" />
            <Property Name="ListAgentKeyNumeric" Type="Edm.Int64" />
            <Property Name="ListAgentLastName" Type="Edm.String"
MaxLength="50" />
            <Property Name="ListingId" Type="Edm.String" Nullable="
false" MaxLength="255">
                <Annotation Term="Org.OData.Core.V1.Permissions">
                    <EnumMember>Org.OData.Core.V1.Permission/Read<
/EnumMember>
                </Annotation>
            </Property>
            <Property Name="ListingKeyNumeric" Type="Edm.Int64"
Nullable="false">
                <Annotation Term="Org.OData.Core.V1.Permissions">
                    <EnumMember>Org.OData.Core.V1.Permission/Read<
/EnumMember>
                </Annotation>
            </Property>
            <Property Name="ListPrice" Type="Edm.Decimal" Scale="2"
Precision="14" />
            <Property Name="ModificationTimestamp" Type="Edm.
DateTimeOffset" ConcurrencyMode="Fixed" Nullable="false">
                <Annotation Term="Org.OData.Core.V1.Permissions">
                    <EnumMember>Org.OData.Core.V1.Permission/Read<
/EnumMember>
                </Annotation>
            </Property>
            <Property Name="OriginalEntryTimestamp" Type="Edm.
DateTimeOffset">
                <Annotation Term="Org.OData.Core.V1.Permissions">
                    <EnumMember>Org.OData.Core.V1.Permission/Read<
/EnumMember>
                </Annotation>
            </Property>
            <Property Name="PostalCode" Type="Edm.String"
MaxLength="10" />
            <Property Name="PostalCodePlus4" Type="Edm.String"
MaxLength="4" />
            <Property Name="PropertySubType" Type="MyCompany.OData.
Models.PropertySubType">
                <Annotation Term="MyCompany.OData.Metadata.RESO.

```

```

TextMaxLength" Int="75" />
    </Property>
    <Property Name="PropertyType" Type="MyCompany.OData.
Models.PropertyType">
        <Annotation Term="MyCompany.OData.Metadata.RESO.
TextMaxLength" Int="75" />
            </Property>
            <Property Name="PublicRemarks" Type="Edm.String"
MaxLength="2000" />
            <Property Name="StandardStatus" Type="MyCompany.OData.
Models.Status">
                <Annotation Term="MyCompany.OData.Metadata.RESO.
TextMaxLength" Int="75" />
                    </Property>
                    <Property Name="StateOrProvince" Type="MyCompany.OData.
Models.States">
                        <Annotation Term="MyCompany.OData.Metadata.RESO.
TextMaxLength" Int="75" />
                            </Property>
                            <Property Name="StatusChangeTimestamp" Type="Edm.
DateTimeOffset" />
                                <Property Name="StreetName" Type="Edm.String"
MaxLength="50" />
                                    <Property Name="StreetNumber" Type="Edm.String"
MaxLength="50" />
                                        <Property Name="StreetSuffix" Type="MyCompany.OData.
Models.StreetSuffix">
                                            <Annotation Term="MyCompany.OData.Metadata.RESO.
TextMaxLength" Int="75" />
                                                </Property>
                                                ...
                                            <NavigationProperty Name="Media" Type="Collection
(MyCompany.OData.Models.Media)" />
                                                <NavigationProperty Name="OpenHouse" Type="Collection
(MyCompany.OData.Models.OpenHouse)" />
                                                ...
                                            </EntityType>
                                        <EnumType Name="City">
                                            <Member Name="_29P" Value="1">
                                                <Annotation Term="MyCompany.OData.Metadata.MLSName"
String="29 Palms" />
                                                    </Member>
                                                <Member Name="AB" Value="2">
                                                    <Annotation Term="MyCompany.OData.Metadata.MLSName"
String="Arrowbear" />
                                                        </Member>
                                                    ...
                                                </EnumType>
                                        ...
                                ...
                    ...
            ...
    ...

```

```

        </Member>
        ...
    </EnumType>

    <EnumType Name="PropertyType">
        <Member Name="Resi" Value="1">
            <Annotation Term="MyCompany.OData.Metadata.RESO.
StandardName" String="Residential" />
        </Member>
        <Member Name="Land" Value="2" />
        <Member Name="Comm" Value="3">
            <Annotation Term="MyCompany.OData.Metadata.RESO.
StandardName" String="Commercial Sale" />
        </Member>
        <Member Name="Manu" Value="4">
            <Annotation Term="MyCompany.OData.Metadata.RESO.
StandardName" String="Manufactured In Park" />
        </Member>
        <Member Name="Rinc" Value="5">
            <Annotation Term="MyCompany.OData.Metadata.RESO.
StandardName" String="Residential Income" />
        </Member>
        <Member Name="Busop" Value="6">
            <Annotation Term="MyCompany.OData.Metadata.RESO.
StandardName" String="Business Opportunity" />
        </Member>
        <Member Name="Rlse" Value="7">
            <Annotation Term="MyCompany.OData.Metadata.RESO.
StandardName" String="Residential Lease" />
        </Member>
        <Member Name="Clse" Value="8">
            <Annotation Term="MyCompany.OData.Metadata.RESO.
StandardName" String="Commercial Lease" />
        </Member>
        <Member Name="FARM" Value="9">
            <Annotation Term="MyCompany.OData.Metadata.RESO.
StandardName" String="Farm" />
        </Member>
    </EnumType>

    <EnumType Name="States">
        <Member Name="AK" Value="1" />
        <Member Name="AL" Value="2" />
        <Member Name="AR" Value="3" />
        <Member Name="AZ" Value="4" />
        <Member Name="CA" Value="5" />
        ...
    </EnumType>

    <EntityContainer Name="Container">

```



```

        <EntitySet Name="DataSystem" EntityType="MyCompany.
OData.Models.DataSystem" />
        <EntitySet Name="Property" EntityType="MyCompany.OData.
Models.Property">
            <NavigationPropertyBinding Path="BuyerAgent"
Target="Member" />
            <NavigationPropertyBinding Path="BuyerOffice"
Target="Office" />
            <NavigationPropertyBinding Path="CoBuyerAgent"
Target="Member" />
            <NavigationPropertyBinding Path="CoBuyerOffice"
Target="Office" />
            <NavigationPropertyBinding Path="CoListAgent"
Target="Member" />
            <NavigationPropertyBinding Path="CoListOffice"
Target="Office" />
            <NavigationPropertyBinding Path="ListAgent" Target="
Member" />
            <NavigationPropertyBinding Path="Media" Target="
Media" />
            <NavigationPropertyBinding Path="OpenHouse" Target="
OpenHouse" />
            <NavigationPropertyBinding Path="ListOffice"
Target="Office" />
            <Annotation Term="Org.OData.Core.V1.
OptimisticConcurrency">
                <Collection>
                    <PropertyPath>ModificationTimestamp<
/PropertyPath>
                </Collection>
            </Annotation>
        </EntitySet>

        <EntitySet Name="OpenHouse" EntityType="MyCompany.OData.
Models.OpenHouse">
            <NavigationPropertyBinding Path="Property" Target="
Property" />
            <Annotation Term="Org.OData.Core.V1.
OptimisticConcurrency">
                <Collection>
                    <PropertyPath>ModificationTimestamp<
/PropertyPath>
                </Collection>
            </Annotation>
        </EntitySet>
        <EntitySet Name="Member" EntityType="MyCompany.OData.
Models.Member">
            <NavigationPropertyBinding Path="Media" Target="
Media" />
            <NavigationPropertyBinding Path="Property" Target="

```

```

Property" />
    <NavigationPropertyBinding Path="Office" Target="
Office" />
    <Annotation Term="Org.OData.Core.V1.
OptimisticConcurrency">
        <Collection>
            <PropertyPath>ModificationTimestamp<
/PropertyPath>
            </Collection>
        </Annotation>
    </EntitySet>
    <EntitySet Name="Office" EntityType="MyCompany.OData.
Models.Office">
        <NavigationPropertyBinding Path="Member" Target="
Member" />
        <NavigationPropertyBinding Path="Media" Target="
Media" />
        <NavigationPropertyBinding Path="Property" Target="
Property" />
        <Annotation Term="Org.OData.Core.V1.
OptimisticConcurrency">
            <Collection>
                <PropertyPath>ModificationTimestamp<
/PropertyPath>
                </Collection>
            </Annotation>
        </EntitySet>
        <EntitySet Name="Media" EntityType="MyCompany.OData.
Models.Media">
            <Annotation Term="Org.OData.Core.V1.
OptimisticConcurrency">
                <Collection>
                    <PropertyPath>ModificationTimestamp<
/PropertyPath>
                    </Collection>
                </Annotation>
            </EntitySet>
        </EntityContainer>

        <Annotations Target="MyCompany.OData.Metadata.RESO.
StandardName">
            <Annotation Term="Org.OData.Core.V1.Description"
String="Original corresponding name defined in RESO Data Dictionary." />
        </Annotations>
        <Annotations Target="MyCompany.OData.Metadata.RESO.
TextMaxLength">
            <Annotation Term="Org.OData.Core.V1.Description"
String="The number of bytes needed to read the data column from
database for this Enum type OData entity property" />
        </Annotations>

```

```

        <Annotations Target="MyCompany.OData.Metadata.MLSName">
            <Annotation Term="Org.OData.Core.V1.Description"
String="The OData property name is not defined in RESO Dictionary, it
is defined by MLS." />
        </Annotations>

    </Schema>
</edmx:DataServices>
</edmx:Edmx>

```

5.2 Examples for Creating Data

The following examples illustrate how to add new data to a system. The examples are using POST only since PUT is really the same operation, with the exception that the PUT operation must happen with a specified key as part of the PUT request in the case that the resource being updated allows the client to control the key values being generated for new data records.

Example 1 - Request

```

POST serviceRoot/Property
OData-Version: 4.01
Content-Type: application/json;odata.metadata=minimal
Accept: application/json
{
  "BathroomsTotalInteger" : 3,
  "BedroomsTotal" : 3,
  "City" : "29 Palms",
  "CoolingYN" : true,
  "CountyOrParish" : "SanBernardino",
  "HeatingYN" : true,
  "ListAgentFirstName" : "Adam",
  "ListAgentKeyNumeric" : 1234,
  "ListAgentLastName" : "Smith",
  "ListPrice" : 799950,
  "PostalCode" : "12345",
  "PropertySubType" : "SFR",
  "PropertyType" : "Resi",
  "PublicRemarks" : "Vestibulum Hic, te potest imaginari feature gatum
praedium iactat omnis.",
  "StandardStatus" : "A",
  "StateOrProvince" : "CA",
  "StreetName" : "Main",
  "StreetNumber" : "123",
  "StreetSuffix" : "ST"
}

```

Example 1 - Response

```

HTTP/1.1 201 Created
OData-Version: 4.01
EntityId: "12345"
Location: serviceRoot/Property('12345')
Content-Length: N
Content-Type: application/json;odata.metadata=minimal;odata.streaming=true;IEEE754Compatible=false;charset=utf-8
{
  "@odata.context": "serviceRoot/$metadata#Property/$entity",
  "@odata.id": "serviceRoot/Property('12345')",
  "@odata.editLink": "serviceRoot/People('12345')",
  "@odata.etag": "W/\\"MjAxOC0wMS0yMlQwOD0lNjo0NS4yMi0wODowMA==\\"",

  "BathroomsTotalInteger" : 3,
  "BedroomsTotal" : 3,
  "City" : "29 Palms",
  "CoolingYN" : true,
  "CountyOrParish" : "SanBernardino",
  "HeatingYN" : true,
  "ListAgentFirstName" : "Adam",
  "ListAgentKeyNumeric" : 1234,
  "ListAgentLastName" : "Smith",
  "ListingId" : "12345",
  "ListingKeyNumeric" : 12345,
  "ListPrice" : 799950,
  "ModificationTimestamp" : "2017-12-05T12:34:34.357-08:00",
  "OriginalEntryTimestamp" : "2017-12-05T12:34:34.357-08:00",
  "PostalCode" : "12345",
  "PropertySubType" : "SFR",
  "PropertyType" : "Resi",
  "PublicRemarks" : "Vestibulum Hic, te potest imaginari feature gatum praedium iactat omnis.",
  "StandardStatus" : "A",
  "StateOrProvince" : "CA",
  "StatusChangeTimestamp" : null,
  "StreetName" : "Main",
  "StreetNumber" : "123",
  "StreetSuffix" : "ST"
}

```

Example 1 Notes (From OData Version 4.01. Part 1: Protocol)

8.3.2 Header EntityId (OData-EntityId)

A response to a create or upsert operation that returns 204 No Content MUST include an EntityId response header.

The EntityId header was named OData-EntityId in OData version 4.0. Services MUST return the OData-EntityId header for responses that have an OData.Version response header of 4.0 and the EntityId header for response with an OData.Version header of 4.01 or greater, and MAY return both EntityId and OData-EntityId headers in the same response.

8.3.4 Header Location

The Location header MUST be returned in the response from a Create Entity or Create Media Entity request to specify the edit URL, or for read-only entities the read URL, of the created entity, and in responses returning 202 Accepted to specify the URL that the client can use to request the status of an asynchronous request.

Example 2 - Request

```
POST serviceRoot/Property
OData-Version: 4.01
Prefer: return=minimal
Content-Type: application/json;odata.metadata=minimal
Accept: application/json
{
  "BathroomsTotalInteger" : 3,
  "BedroomsTotal" : 3,
  "City" : "29 Palms",
  "CoolingYN" : true,
  "CountyOrParish" : "SanBernardino",
  "HeatingYN" : true,
  "ListAgent" : { "@id" : "Member(1234)" },
  "CoListAgent" : { "@id" : "Member(7890)" },
  "ListPrice" : 799950,
  "PostalCode" : "12345",
  "PropertySubType" : "SFR",
  "PropertyType" : "Resi",
  "PublicRemarks" : "Vestibulum Hic, te potest imaginari feature gatum
praedium iactat omnis.",
  "StandardStatus" : "A",
  "StateOrProvince" : "CA",
  "StreetName" : "Main",
  "StreetNumber" : "123",
  "StreetSuffix" : "ST"
}
```

Example 2 - Response

```
HTTP/1.1 204 No Content
OData-Version: 4.01
EntityId: "12345"
Location: serviceRoot/Property('12345')
Preference-Applied: return=minimal
```

[Example 2 Notes \(From OData Version 4.01. Part 1: Protocol\)](#)

8.2.8.7 Preference return=representation and return=minimal

A preference of return=representation requests that the service invokes the request and returns the modified resource. The service MAY apply this preference by returning the representation of the successfully modified resource in the body of the response, formatted according to the rules specified for the requested format. In this case the service MAY include a Preference-Applied response header containing the return=representation preference.

5.3 Examples for Updating Data

The following examples illustrate how to add update data in a system. The examples are using PATCH only which is the partial update.

Example 1 - Request

```
PATCH serviceRoot/Property('12345')
OData-Version: 4.01
Content-Type: application/json;odata.metadata=minimal
Accept: application/json
If-Match: W/"MjAxOC0wMS0yMlQwOD01Njo0NS4yMi0wODowMA==\"
Prefer: return=representation

{
  "CoolingYN" : false
}
```

Example 1 - Response

```
HTTP/1.1 200 OK
Content-Length: N
Content-Type: application/json;odata.metadata=minimal;odata.streaming=true;IEEE754Compatible=false;charset=utf-8
EntityId: "12345"
Location: serviceRoot/Property('12345')
OData-Version: 4.01
{
  "@odata.context": "serviceRoot/$metadata#Property/$entity",
  "@odata.id": "serviceRoot/Property('12345')",
  "@odata.editLink": "serviceRoot/People('12345')",
  "@odata.etag": "W/\"ABCxOC0wMS0yMlQwOD01Njo0NS4yMi0wODowMA==\"",
  "BathroomsTotalInteger" : 3,
  "BedroomsTotal" : 3,
  "City" : "29 Palms",
  "CoolingYN" : false,
  "CountyOrParish" : "SanBernardino",
  "HeatingYN" : true,
  "ListAgentFirstName" : "Adam",
  "ListAgentKeyNumeric" : 1234,
  "ListAgentLastName" : "Smith",
  "ListingId" : "12345",
  "ListingKeyNumeric" : 12345,
```

```
"ListPrice" : 799950,
"ModificationTimestamp" : "2017-12-05T12:34:34.357-08:00",
"OriginalEntryTimestamp" : "2017-12-05T12:34:34.357-08:00",
"PostalCode" : "12345",
"PropertySubType" : "SFR",
"PropertyType" : "Resi",
"PublicRemarks" : "Vestibulum Hic, te potest imaginari feature gatum
praedium iactat omnis.",
"StandardStatus" : "A",
"StateOrProvince" : "CA",
"StatusChangeTimestamp" : null,
"StreetName" : "Main",
"StreetNumber" : "123",
"StreetSuffix" : "ST"
}
```

Example 2 - Request

```
PATCH serviceRoot/Property('12345')
OData-Version: 4.01
Prefer: return=minimal
Content-Type: application/json;odata.metadata=minimal
Accept: application/json
If-Match: W/\ "MjAxOC0wMS0yMlQwODo1Njo0NS4yMi0wODowMA==\"

{
  "CoolingYN" : false
}
```

Example 2 - Response

```
HTTP/1.1 204 No Content
OData-Version: 4.01
EntityId: "12345"
```

5.3 Examples for Deleting Data

The following examples illustrates data being deleted from a system.

Example 1 - Request

```
DELETE serviceRoot/Property('12345')
OData-Version: 4.01
If-Match: W/"MjAxOC0wMS0yMlQwODo1Njo0NS4yMi0wODowMA=="
```

Example 1 - Response

```
HTTP/1.1 204 No Content
OData-Version: 4.01
```

Further Notes

Certification Impact