RCP - WEBAPI-032 Lookup and RelatedLookup Resources for Lookup Metadata

Submitter Name	Josh Darnell (**Currently Editing**)
Submitter Organization	RESO
Submitter Email	josh@reso.org
Submitter Name	Ryan Yates
Submitter Organization	Rapattoni Corporation
Submitter Email	ryates@rapattoni.com
Co-submitter Name	Sergio Del Rio
Co-submitter Organization	Templates 4 Business Inc.
Co-submitter Email	Sergio.Del.Rio@t4bi.com
Co-submitter Name	Rob Larson
Co-submitter Email	rob@larson.cc
Co-submitter Name	Paul Stusiak
Co-submitter Organization	Falcon Technologies
Co-submitter Email	pstusiak@falcontechnologies.com

Document Name	Web API Core
Document Version	1.0.2+
Requirement Level	**MAY**
Date Submitted	August 25, 2020
Status	DRAFT
Status Change Date	Feb 1, 2021
Related RCPs	RCP-031, RCP-033
Related Documents	Data Dictionary 1.7.0 Spreadsheet Data Dictionary 1.7.0 Specification ORIGINAL PRESENTATION

Synopsis

The goal of this proposal is to create a RESO standard resource called ***Lookup*** that can be used to convey metadata about enumerations available on a given server.

Additionally, support for relationships between lookups is included in this proposal by an optional resource called ***RelatedLookup****.*

Rationale

Lookup Resource

In systems that cover large geographic areas, the amount of metadata available on a given server can grow quite large. This is due to the fact that there are lookups for cities, counties, subdivisions, etc. for each of the areas a given vendor covers, which causes a lot of overhead in replicating and processing the entire OData metadata document each time it changes.

Prior to Data Dictionary 1.7, vendors had been allowed to provide documentation for their externally enumerated fields and then transport them as strings rather than declaring them statically in their OData XML Metadata using OData lookups.

However, the Transport, Certification, and Common Schema groups, as well as others in the community, have been working on viable alternatives to OData lookups that would allow Data Dictionary `String List, Single` and `String List, Multi` enumerations to be expressed using OData `Edm.String` and `Collection(Edm.String)` types rather than OData `Edm.EnumType` lookups. This is mainly due to the fact that OData doesn't allow special characters in the values it uses for enumerations, requiring special values instead and making things difficult for data consumers; rather than being able to use the value that comes across in the payload, additional work is currently needed to decode values in the server metadata to figure out what should be displayed to downstream consumers of the data. This process should be made easy for data consumers.

The Lookup resource allows human-friendly display names to be used in transport, both for queries and payloads, while providing consumers with a way to retrieve metadata as needed rather than all at once.

RelatedLookup Resource (Optional)

There is also a general business need to convey relationships between lookup items, geographically related or otherwise, which the optional *RelatedLookup* resource outlined in this proposal addresses.

By adding support for related lookups, this proposal meets an important need for data consumers: there is currently no mechanism to convey which location-based lookups are related to each other, which is crucial for winnowing select lists (picklists) on both search and add/edit forms. For example, if a consumer is looking for all subdivisions in a given city. This becomes even more important for add/edit, as users need to be able to select valid choices for their inputs in order to pass business rules.

Proposal

This proposal requests the creation of a new resource called ***Lookup*** to provide a more streamlined way to deliver enumeration metadata to the consumer.

It also provides an optional resource called ***RelatedLookup*** which data providers can use to transport information about relationships between lookups, such as in the case of geographically related items.

Lookup Resource

A new resource called `Lookup` will be created with the following structure:

Field	Data Type	Sample Value	Nullable	Description

LookupKey	Edm.String	"ABC123"	false	The key used to uniquely identify the Lookup entry.
LookupName	Edm.String	"ListingAgreementType"	false	The name of the enumeration. This is the ***LookupField*** in the adopted Data Dictionary 1.7 spreadsheet. It is called a "LookupName" in this proposal because more than one field can have a given lookup, so it refers to the name of the lookup rather than a given field. For example, Listing with CountyOrParish and Office with OfficeCountyOrParish having the same CountyOrParish LookupName. This MUST match the Data Dictionary definition for in cases where the lookup is defined. Vendors MAY add their own enumerations otherwise. The LookupName a given field uses is required to be annotated at the field level in the OData XML Metadata, as outlined later in this proposal.
LookupValue	Edm.String	"Seller Reserve"	false	The human-friendly display name the data consumer receives in the payload and uses in queries. This MAY be a local name or synonym for a given RESO Data Dictionary lookup item.
StandardLookupValue	Edm.String	"Exclusive Agency"	true	The Data Dictionary ***LookupDisplayName*** of the enumerated value. This field is required when the LookupValue for a given item corresponds to a RESO standard value, meaning a standard lookup display name, known synonym, local name, or translation of that value. Local lookups MAY omit this value if they don't correspond to an existing RESO standard lookup value.
LegacyODataValue	Edm.String	"ExclusiveAgency"	true	The Legacy OData lookup value that the server vendor provided in their OData XML Metadata. This value is optional, and has been included in order to provide a stable mechanism for translating OData lookup values to RESO standard lookup display names, as well as for historical data that might have included the OData value at some point, even after the vendor had converted to human friendly display names.
ModificationTimestamp	Edm.DateTimeOffset	"2020-07-07T17:36:14+00:00"	false	The timestamp for when the enumeration value was last modified. This is used to help rebuild caches when metadata items change so consumers don't have to re-pull and reprocess the entire set of metadata when only a small number of changes have been made.

Required Annotation

For any `String List, Single` or `String List, Multi` field using the Lookup resource, an annotation similar to the following MUST be present in the server metadata:

Example

```
<!-- OData annotation for String List, Single field -->
<Property Name="OfficeCountyOrParish" Type="Edm.String">
  <Annotation Term="RESO.OData.Metadata.LookupName" String="CountyOrParish" />
</Property>

<!-- OData annotation for String List, Multi field -->
<Property Name="ExteriorFeatures" Type="Collection(Edm.String)">
  <Annotation Term="RESO.OData.Metadata.LookupName" String="ExteriorFeatures" />
</Property>
```

Where:

- **Term** uses the required namespace `RESO.OData.Metadata.LookupName`
- **String** indicates the LookupName in the Lookup Resource in which the given field's lookups are defined.

Notes:

- The referenced *LookupName* MUST be a standard lookup name for items currently defined by the RESO Data Dictionary.
- Data providers MAY add additional *LookupName* entries when not already defined by the Dictionary.
- The underlying type for the lookup-based field MUST either be `Edm.String` or `Collection(Edm.String)`, depending on whether the given field is String List, Single or Multi in the Data Dictionary, respectively.

Example: GET Lookup items for a given LookupName

The following request shows a sample query to the **Lookup** resource for the lookups having a *LookupName* of *CountyOrParish*:

```
GET /Lookup?$filter=LookupName eq 'CountyOrParish'
{
 "@odata.context": "https://your.resoapi.com/Lookup?$filter=LookupName eq 'CountyOrParish'",
 "value": [{
   "LookupKey": "ABC123",
   "LookupName": "CountyOrParish",
   "LookupValue": "Los Angeles County",
   "StandardLookupValue": null,
   "ModificationTimestamp": "2020-07-07T17:36:14+00:00"
 }, {
   "LookupKey": "BCD124",
   "LookupName": "CountyOrParish",
   "LookupValue": "Ventura County",
   "StandardLookupValue": null,
   "ModificationTimestamp": "2020-07-07T17:36:15+00:00"
 }, {
   "LookupKey": "CDE125",
   "LookupName": "CountyOrParish",
   "LookupValue": "Contra Costa County",
   "StandardLookupValue": null,
   "ModificationTimestamp": "2020-07-07T17:36:16+00:00"
 }, ...]
}
```

The *Lookup* resource provides the ability to retrieve all items with *LookupName* of 'CountyOrParish' or LookupValue of 'Los Angeles County', but doesn't contain enough information to fetch the CountyOrParish values for a given StateOrProvince, for example.

This is where the optional *RelatedLookup* resource outlined in the next section comes into play.

Optional Resource: RelatedLookup

The optional **RelatedLookup** resource has been included to model relationships between Lookup resource items.

A *RelatedLookup* can be thought of a directed edge from *LookupKeyA* to *LookupKeyB*, with a *ModificationTimestamp* field for replication purposes.

It has the following structure:

Field	Data Type	Sample Value	Nullable	Description
LookupKey	Edm.String	"ABC123"	false	The unique key of the Lookup resource item the relationship originates from.
RelatedLookupKey	Edm.String	"X22"	false	The unique key of the Lookup resource item the relationship "points to" or terminates at.
ModificationTimestamp	Edm.DateTimeOffset	"2020-07-07T17:36:14+00:00"	false	The timestamp for when the relationship was last modified.

A common use case would be that a data consumer would already know the *LookupKey* they are interested in, and would then retrieve *RelatedLookupKey* items from the *RelatedLookup* resource.

Example: GET RelatedLookup entries for a given Lookup

Assuming "ABC123" is the set of related lookups for the CountyOrParish with value "Los Angeles County" above, the related lookups resource might contain records similar to the following:

```
GET /RelatedLookup?$filter=LookupKey eq 'ABC123'
{
 "@odata.context": "https://your.resoapi.com/RelatedLookup?$filter=LookupKey eq 'ABC123'",
 "value": [{
   "LookupKey": "ABC123",
   "RelatedLookupKey": "X22",
   "ModificationTimestamp": "2020-07-07T17:36:14+00:00"
 }, {
   "LookupKey": "ABC123",
   "RelatedLookupKey": "Y33",
   "ModificationTimestamp": "2020-07-07T17:36:15+00:00"
 }, {
   "LookupKey": "ABC123",
   "RelatedLookupKey": "Z44",
   "ModificationTimestamp": "2020-07-07T17:36:16+00:00"
 }]
}
```

In the above example:

- "X22" is a *StateOrProvince* lookup key for "Los Angeles County" (LookupKey "ABC123").
- "Y33" is a *City* lookup key for a given City in "Los Angeles County".
- "Z44" is a *Subdivision* lookup key belonging to "Los Angeles County".

Using queries alone, the next step would be to pull from the *Lookup* resource where the *RelatedLookupKey* equals retrieved from the *RelatedLookup* resource in the previous example.

Example: GET Lookup entries for given RelatedLookup Items

Once related lookups have been retrieved, the data consumer would then need to requery the Lookup resource for those items.

```
GET /Lookup?$filter=LookupKey eq 'ABC123' or LookupKey eq 'X22' or LookupKey eq 'Y33' or LookupKey eq 'Z44'
{
 "@odata.context": "https://your.resoapi.com/Lookup?$filter=LookupKey eq 'ABC123' or LookupKey eq 'X22' or Looku
 "value": [{
   "LookupKey": "ABC123",
   "LookupName": "CountyOrParish",
   "LookupValue": "Los Angeles County",
   "DisplayName": null,
   "ModificationTimestamp": "2020-07-07T17:36:14+00:00"
 }, {
   "LookupKey": "X22",
   "LookupName": "StateOrProvince",
   "LookupValue": "CA",
   "StandardLookupValue": "CA",
   "ModificationTimestamp": "2020-07-07T17:36:15+00:00"
 }, {
   "LookupKey": "Y33",
   "LookupName": "City",
   "LookupValue": "City of Los Angeles",
   "StandardLookupValue": null,
   "ModificationTimestamp": "2020-07-07T17:36:15+00:00"
 }, {
   "LookupKey": "Z44",
   "LookupName": "Subdivision",
   "LookupValue": "Sawtelle",
   "StandardLookupValue": null,
   "ModificationTimestamp": "2020-07-07T17:36:16+00:00"
 }]
}
```

Example: GET all 'City' values for a given 'CountyOrParish'

Suppose we want to retrieve all the *City* lookups for the *CountyOrParish* of "Los Angeles County" with *LookupKey* "ABC123" that we retrieved in a previous example.

In this case, we can add `LookupName eq 'City'` to the query in order to filter only *City* lookups for the *LookupKey* items retrieved in the *RelatedLookup* query:

```
GET /Lookup?$filter=LookupName eq 'City' and (LookupKey eq 'X22' or LookupKey eq 'Y33' or LookupKey eq 'Z44')
{
 "@odata.context": "https://your.resoapi.com/Lookup?$filter=LookupName eq 'City' and (LookupKey eq 'X22' or Look
 "value": [{
   "LookupKey": "Y33",
   "LookupName": "City",
   "LookupValue": "City of Los Angeles",
   "StandardLookupValue": null,
   "ModificationTimestamp": "2020-07-07T17:36:15+00:00"
 }]
}
```

Optional Section: Navigation Property Path Queries

In the previous examples, multiple queries were required to fetch related lookups.

For those using OData navigation property paths, *Lookup* and *RelatedLookup* items can be fetched in a single query.

Example: GET Lookup Items and Related Lookup Items for a given Lookup Item

```
GET /Lookup?$filter=LookupKey eq 'ABC123' or (RelatedLookup/any(a:a/RelatedLookupKey eq 'ABC123') and LookupName
{
  "@odata.context": "https://your.resoapi.com/Lookup?$filter=LookupKey eq 'ABC123' or (RelatedLookup/any(a:a/Rel
  "value": [{
    "LookupKey": "ABC123",
    "LookupName": "CountyOrParish",
    "LookupValue": "City of Los Angeles",
    "StandardLookupValue": null,
    "ModificationTimestamp": "2020-07-07T17:36:15+00:00"
  }, {
    "LookupKey": "ab8298",
    "LookupName": "City",
    "LookupValue": "City of Santa Monica",
    "StandardLookupValue": null,
    "ModificationTimestamp": "2020-07-07T17:36:15+00:00"
  }, {
    "LookupKey": "23oerr2",
    "LookupName": "City",
    "LookupValue": "City of Westwood",
    "StandardLookupValue": null,
    "ModificationTimestamp": "2020-07-07T17:36:16+00:00"
 }]
}
```

Note: the following ***StandardRelationship*** will need to be added to the *Lookup* resource to support the above navigation property path queries:

Lookup has many Lookup as *RelatedLookup* through *RelatedLookupKey*

Impact

Vendors

- This is a MAY proposal for Web API Core 1.0.2 and Data Dictionary 1.6.0+.
- Vendors MAY continue to use `OData Edm.EnumType` and `Collection(Edm.EnumType)` for Data Dictionary String List, Single or Multi enumerations.
- Lookups, whether OData lookups or those in the Lookup resource, will not be tested in Data Dictionary 1.7. Additional requirements MAY be added to the Data Dictionary specification, as necessary.
- Data Type Mappings will be updated in RCP-031 and the Data Dictionary 1.7 Testing Specification to reflect support for Edm.String and Collection(Edm.String) to be allowed types for Data Dictionary String List, Single and Multi items.
- Fields using the Lookup resource MUST use the annotation format described in this proposal.
- Servers MUST return a non 200 HTTP response if they do not support the `Lookup` resource.

Compatibility

- Web API Core 1.0.2+
- Data Dictionary 1.6.0+

Certification Impact

- Upon approval, acceptance tests need to be updated to allow either `OData Edm.EnumType` or `Edm.String` for Single enumerations and `Collection(Edm.EnumType)` or `Collection(Edm.String)` for Multiple enumerations and to check for a Lookup resource defined according to this specification.

- Additional acceptance tests will need to be added to ensure that the annotation format and requests and responses behave as intended for Lookup items.
- Once there are two implementations, an endorsement and testing tools can be created to ensure compliance with the specification. The *Lookup* or *RelatedLookup* resource are not required for Data Dictionary 1.7.
- If the *RelatedLookup* resource is used, it should follow the appropriate standard navigation property path definition outlined in the Data Dictionary specification.