

SharePoint REST Basics

Contents

- REST and OData Overview
- OData Queries and Expanding
- Implementing CRUD Operation using REST
- Managing List & List Items

REST and OData Overview

REST

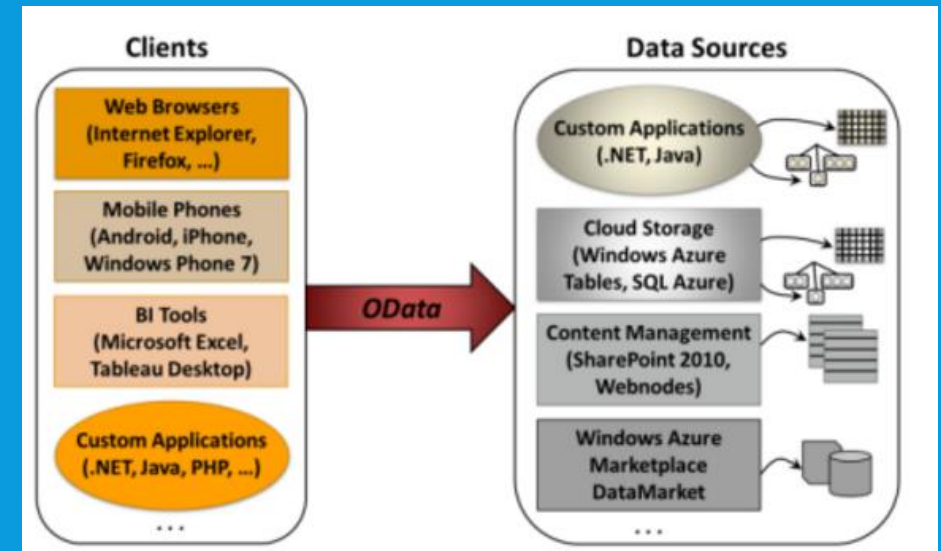
- REST is an architectural style, not a specification
- The Web is a graph of linked Resources
- Resources are identified by URI's
- Resources support a fixed set of operations
 - In practice, these are defined by HTTP
- Applications follow links to achieve late binding

HTTP Verbs

- RESTful (Representational State Transfer) web services use HTTP verbs to map CRUD operations to HTTP methods.
- RESTful web services expose either a collection resource (representational of a list) or an element resource (representational of a single item in the list)
- HTTP verbs are used as follows;
 - Create (POST) > create a new resource.
 - Read (GET) > retrieve one or many resources.
 - Update (PUT) > update an existing resource.
 - Delete (DELETE) > delete an existing resource.

oData

- Best practice for building and consuming RESTful APIs
- OData defines an abstract data model and a protocol that let any client access information exposed by any data source
- OData client chooses to have the data it receives serialized using Atom/AtomXML or JSON

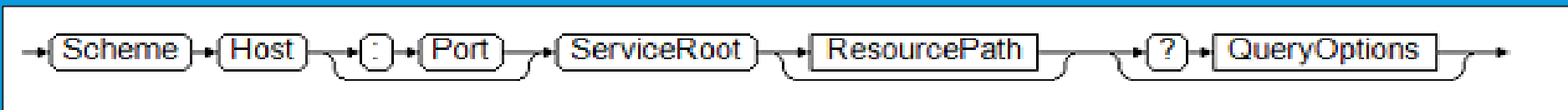


OData Terms and Concepts

Odata Resource	Is Described in an Entity Data Model by
Collection	Entity Set A navigation property on an entity type that identifies a collection of entities
Entry	Entity Type Note: Entity Types may be part of a type hierarchy
Property of an entry	Primitive or Complex Entity Type Property
Complex Type	Complex Type
Link	A Navigation Property defined on an Entity Type
Service Operation	Function Import

OData URIs

- URI has three significant parts
 - Service root URI
 - Resource path
 - Query string options



```
http://services.odata.org/OData/OData.svc/Category(1)/Products?$top=2&$orderby=name
```

Below the URI, three brackets indicate the corresponding parts:

- service root URI (under `http://services.odata.org/OData/OData.svc/`)
- resource path (under `Category(1)/Products`)
- query options (under `?$top=2&$orderby=name`)

REST Urls in SharePoint

- CSOM URLs can go through _api folder
 - Simplifies URLs that need to be built
 - Removes client.svc file name from URL
- You can replace this URL
 - `http://contososerver/_vti_bin/client.svc/web`
- With this URL
 - `http://contososerver/_api/web`

Mapping Objects to Resources

Example REST URLs targeting SharePoint sites

- `_api/web/lists`
- `_api/web/lists/getByTitle('Announcements')`
- `_api/web/getAvailableWebTemplates(lcid=1033)`

Returning ATOM XML vs. JSON

Response data format selected with ACCEPT header

- XML can be easier to deal with from managed code
- To get ATOM XML response use "application/atom+xml"
- JSON is easier to deal with when using JavaScript
- To get JSON response use "application/json;odata=verbose"

```
{
  "d": {
    "metadata": {
      "id": "d8d432f4-90a6-42d3-84c1-9f6b6a47f97f",
      "uri": "http://contososerver/_vti_bin/client.svc/web/Lists(guid'0",
      "etag": "\"0\"",
      "type": "SP.List"
    },
    "Id": "0a432ace-c6c8-4541-babc-e3c8bc3411a6",
    "Title": "Customers"
  }
}
```

OData Syntax

- \$metadata: Self describing metadata about the list
- \$select: Allows to select certain fields only
- \$filter=: Returns only entities that match the specified expression.
- \$orderby=: Orders results, in ascending or descending order.

OData Syntax

- \$top=n: Returns only the first n entities in an entity set
- \$skip=n: Skips the first n entities in an entity set.
- \$format: Determines whether data should be returned in JSON or the XML-based Atom/AtomPub format.
- \$expand: Expands associated entities
- Full syntax - <https://msdn.microsoft.com/en-us/library/office/fp142385.aspx>

\$expand

- Objects that represent complex types must be expanded in order to access the value
 - User fields (Author, Editor, ...)
 - Lookup
 - Geolocation
 - Link

```
/_api/lists/getByTitle('Assets')/Items?$select=LinkTitle,Owner/Name&$expand=Owner
```

```
1 {
2   "d": {
3     "results": [
4       {
5         "_metadata": {
6           "id": "Web/Lists(guid'3812a010-bced-4b5a-b4a9-123ac542f58f')/Items(1)",
7           "uri": "http://sp2013c/_api/Web/Lists(guid'3812a010-bced-4b5a-b4a9-123ac542f58f')/Items(1)",
8           "etag": "\"1\"",
9           "type": "SP.Data.AssetsListItem"
10        },
11        "Owner": {
12          "_metadata": {
13            "id": "98c337f0-2700-4860-88ab-98eadc6da573",
14            "type": "SP.Data.UserInfoItem"
15          },
16          "Name": "i:0#.w|spdom\\administrator"
17        },
18        "LinkTitle": "Chair"
19      },
20    ]
21  }
```

\$expand & user profile

- When extending a user field several user profile properties can be retrieved, but not all. The following user profile properties will be returned:
- Name
- Title
- Department
- JobTitle

Columns	
A column stores information about each item in the list. The following columns	
Column (click to edit)	Type
Title	Single line of text
Owner	Person or Group
Modified	Date and Time
Created	Date and Time
Created By	Person or Group
Modified By	Person or Group

UserProfile Properties using PeopleManager

- SP.UserProfiles.PeopleManager can be used to retrieve props

```
//Get all properties of current user
execLog("/_api/SP.UserProfiles.PeopleManager/GetMyProperties");
//Get single property of current user
execLog("_api/SP.UserProfiles.PeopleManager/GetMyProperties/PictureUrl");
//Get all properties of specific user
execLog("_api/SP.UserProfiles.PeopleManager/GetPropertiesFor(accountName=@v)?@v='sp\\administrator'");
execLog("_api/SP.UserProfiles.PeopleManager/GetPropertiesFor(accountName=@v)?@v='i:0%23.f|membership|you@yoursiteurl.onmicrosoft.com'");
```


REST / oData Tools

- Tools like CAML Designer help creating REST queries



CAML Designer 2013 - ENG

ViewFields Order By Where Query Options Test

These are the fields of the list

App Modified By

Attachments

These are the selected fields

Body contain mytext

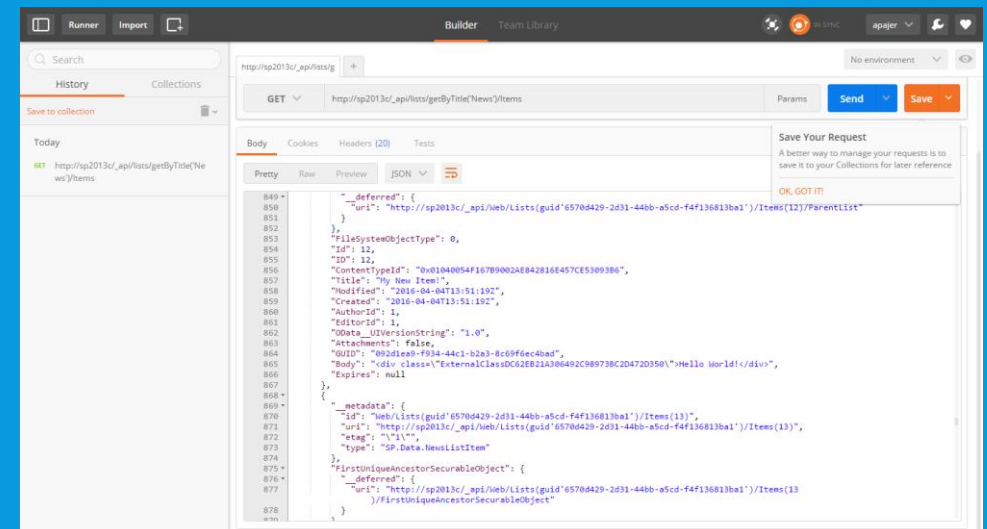
Caml ☒ Json ☐ Atom

```
$ajax({
  url: _spPageContextInfo.webAbsoluteUrl + "/_api/web/lists/getbytitle('News')/Items?$select=ID,Title&$filter=substringof('mytext',Body)",
  type: "GET",
  headers: {"accept": "application/json;odata=verbose"},
  success: function (data) {
    if (data.d.results) {
      // TODO: handle the data
      alert("handle the data");
    }
  },
  error: function (xhr) {
    alert(xhr.status + ': ' + xhr.statusText);
  }
});
```

CSOM Rest

Chrome Add-Ins that support REST

- Postman – Test Client for REST services
- Postman Interceptor – Passes Auth from Chrome to Postman
- Chrome Apps Launcher – Makes launching Postman easier
- JSON Editor visualizes JSON



Managing Lists

Request Digest

- When POSTing the request, you get a HTTP 403 Forbidden back in response from the Search service with an error message stating *"The security validation for this page is invalid and might be corrupted. Please use your web browser's Back button to try your operation again."*
- The reason for getting the 403 Forbidden status is mainly because you are missing an important header in your request which is the X-RequestDigest header.

Creating Lists with JavaScript and jQuery

Create appropriate JavaScript object and convert to JSON format for request body

Create request headers for HTTP Request

Send request to Web server using jQuery \$.ajax function

```
function onCreateList() {  
    var newList = {  
        "__metadata": { "type": "SP.List" },  
        "BaseTemplate": 104,  
        "Description": "A list created by REST",  
        "Title": $("#newListTitle").val()  
    }  
  
    var requestHeaders = {  
        Accept: "application/json;odata=verbose",  
        "X-RequestDigest": $("#__REQUESTDIGEST").val()  
    }  
  
    $.ajax({  
        type: "POST",  
        url: _spPageContextInfo.webAbsoluteUrl + "/_api/web/lists",  
        contentType: "application/json;odata=verbose",  
        data: JSON.stringify(newList),  
        headers: requestHeaders,  
        success: onDataReturned,  
        error: onError  
    });  
}
```

Get form digest value from control on page

Convert request body data to string-based JSON object

Create List

```
var body = JSON.stringify({
  '__metadata': { 'type': 'SP.List' },
  'AllowContentTypes': true,
  'BaseTemplate': 100,
  'ContentTypesEnabled': true,
  'Description': 'My REST List',
  'Title': 'MyRestList'
});
```

```
var digest = $("#__REQUESTDIGEST").val();
var url = _spPageContextInfo.webAbsoluteUrl +
  "/_api/web/lists";

jQuery.ajax({
  url: url,
  type: "POST",
  data: body,
  headers: {
    "accept": "application/json;odata=verbose",
    "content-type": "application/json;odata=verbose",
    "X-RequestDigest": digest
  },
  success: function () { console.log("list created") },
  error: function (msg) { console.log(msg) }
});
```

Update List

- "X-RequestDigest"
- "X-HTTP-Method":"MERGE",
- "IF-MATCH": "*"

```
jQuery.ajax({  
  url: _spPageContextInfo.webAbsoluteUrl + " /_api/web/lists/GetByTitle('MyRestList')",  
  type: "POST",  
  data: JSON.stringify({ '__metadata': { 'type': 'SP.List' }, 'Title': 'My Rest List' }),  
  headers: {  
    "X-HTTP-Method":"MERGE",  
    "accept": "application/json;odata=verbose",  
    "content-type": "application/json;odata=verbose",  
    "X-RequestDigest": $("#__REQUESTDIGEST").val(),  
    "IF-MATCH": "*"   
  },  
  success: function () { console.log("list renamed") },  
  error: function (msg) { console.log(msg) }  
});
```

Delete List

- "X-RequestDigest"
- "X-HTTP-Method":"DELETE",
- "IF-MATCH": "*"

```
$.ajax({  
  url: _spPageContextInfo.webAbsoluteUrl + "/_api/web/Lists/getbytitle('My Rest List')",  
  method: "POST",  
  headers: {  
    "accept": "application/json;odata=verbose",  
    "content-type": "application/json;odata=verbose",  
    "X-RequestDigest": $("#__REQUESTDIGEST").val(),  
    "IF-MATCH": "*",  
    "X-HTTP-Method": "DELETE"  
  },  
  success: function () { console.log("list deleted") },  
  error: function (msg) { console.log(msg) }  
});
```


REST CRUD

REST Query Using JavaScript & jQuery

```
$(onPageLoad);

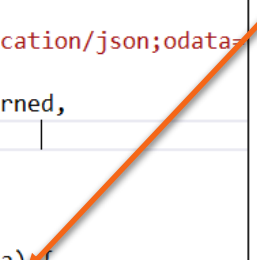
function onPageLoad() {
    $("#cmdGetSiteInfo").click(onGetSiteInfo);
}

function onGetSiteInfo(){
    var requestUri = _spPageContextInfo.webAbsoluteUrl + "/_api/web/?$select=Title";

    $.ajax({
        type: "GET",
        url: requestUri,
        contentType: "application/json",
        headers : {
            Accept : "application/json;odata=application/json"
        },
        success: onDataReturned,
        error: onError
    });
}

function onDataReturned(data){
    var odataResults = data.d;
    var site_title = odataResults.Title;
    $("#results").html("Title: " + site_title);
}

function onError(error) {
    $("#results").text("ERROR : " + JSON.stringify(error));
}
```



```
{
  "d": {
    "__metadata": {
      "id": "http://contoso-fb5f1425f041fd.contosoapps.com/sites/SpAppsCloudHosted/SharePointApp3/_api/Web",
      "uri": "http://contoso-fb5f1425f041fd.contosoapps.com/sites/SpAppsCloudHosted/SharePointApp3/_api/Web",
      "type": "SP.Web",
      "Title": "SharePointApp3"
    }
  }
}
```

Read List

- No Request Digest required because of HTTP GET

```
$.ajax({  
  url: _spPageContextInfo.webAbsoluteUrl + "/_api/lists/getByTitle('News')/Items?$select=Title",  
  type: "GET",  
  headers: {  
    "Accept": "application/json;odata=verbose",  
    "Content-Type": "application/json; odata=verbose"  
  },  
  success: function (data) {  
    if (data.d.results) {  
      // TODO: handle the data  
      console.log('handle the data');  
    }  
  },  
  error: function (msg) { console.log(msg) }  
});
```

Read Item

```
$.ajax({
  url: _spPageContextInfo.webAbsoluteUrl +
    "/_api/web/lists/getbytitle('MyRestList')/GetItems(query=@v1)?@v1={\"ViewXml\": \"<View>\" +
    <Query><OrderBy><FieldRef Name='Created' /></OrderBy></Query></View>\"}",
  type: "POST",
  headers: {
    "X-RequestDigest": $("#__REQUESTDIGEST").val(),
    "Accept": "application/json;odata=verbose",
    "Content-Type": "application/json; odata=verbose"
  },
  success: function (data) {
    if (data.d.results) {
      // TODO: handle the data
      console.log('handle the data');
    }
  },
  error: function (msg) { console.log(msg) }
});
```

Http Headers

- „Writing“ operations require special HTTP Headers

- Create

```
"X-RequestDigest": $("#__REQUESTDIGEST").val()
```

- Update

```
"X-RequestDigest": $("#__REQUESTDIGEST").val()  
"If-Match": "*",  
"X-HTTP-Method": "MERGE"
```

- Delete

```
"X-RequestDigest": $("#__REQUESTDIGEST").val()  
"IF-MATCH": "*",  
"X-HTTP-Method": "DELETE"
```

Create Item

```
var item = {  
  "__metadata": { "type": getListItemType("MyRestList") },  
  "Title": "CodedItem",  
  "My_x0020_Custom_x0020_Field": "My Custom Value"  
};
```

```
$.ajax({  
  url: _spPageContextInfo.webAbsoluteUrl + "/_api/web/lists/getbytitle('MyRestList')/items",  
  type: "POST",  
  contentType: "application/json;odata=verbose",  
  data: JSON.stringify(item),  
  headers: {  
    "Accept": "application/json;odata=verbose",  
    "X-RequestDigest": $("#__REQUESTDIGEST").val()  
  },  
  success: function () { console.log("item added to list") },  
  error: function (msg) { console.log(msg) }  
});
```

Update Item

```
function updateListItem() {  
    var itemProperties = { 'Title': 'The REST News' };  
    updateItem('News', 1, itemProperties, function () {  
        console.log('item has been updated');  
    }, function() {  
        console.log('error');  
    });  
}
```

```
function updateItem(listTitle, listItemId, itemProperties, success, failure) {  
    var listItemUri = _spPageContextInfo.webAbsoluteUrl + "/_api/web/lists/getbytitle('" + listTitle + "')/items(" + listItemId  
        + ")";  
    var itemPayload = {  
        '__metadata': { 'type': "SP.Data." + listTitle.charAt(0).toUpperCase() + listTitle.slice(1) + "ListItem" }  
    };  
    for (var prop in itemProperties) {  
        itemPayload[prop] = itemProperties[prop];  
    }  
    updateJson(listItemUri, itemPayload, success, failure);  
}
```

Update Item - Cont

```
function updateJson(endpointUri, payload, success, error) {  
    $.ajax({  
        url: endpointUri,  
        type: "POST",  
        data: JSON.stringify(payload),  
        contentType: "application/json;odata=verbose",  
        headers: {  
            "Accept": "application/json;odata=verbose",  
            "X-RequestDigest": $("#__REQUESTDIGEST").val(),  
            "X-HTTP-Method": "MERGE",  
            "If-Match": "*"   
        },  
        success: success,  
        error: error  
    });  
}
```


Delete Item

```
var id = 1;
var url = _spPageContextInfo.webAbsoluteUrl + "/_api/web/lists/GetByTitle('News')/items(" + id + ")";

$.ajax({
  url: url,
  type: "POST",
  headers: {
    "ACCEPT": "application/json;odata=verbose",
    "content-type": "application/json;odata=verbose",
    "X-RequestDigest": $("#__REQUESTDIGEST").val(),
    "IF-MATCH": "*",
    "X-HTTP-Method": "DELETE"
  },
  success: function (data) {
    console.log("Deleted Successfully.");
  },
  error: function (error) {
    console.log(JSON.stringify(error));
  }
});
```

JSOM vs REST

Technologies table:

	CSOM	SSOM	REST
JavaScript	✓		✓
C#	✓	✓	✓
From browser		✓	✓
From server	✓	✓	

Hosting table:

	CSOM	JSOM	SSOM	REST JS	REST C#
Sandboxed solution	✓	✓	✓(limited)	✓	✓
Farm solution	✓	✓	✓	✓	✓

SPRemoteAPIExplorer

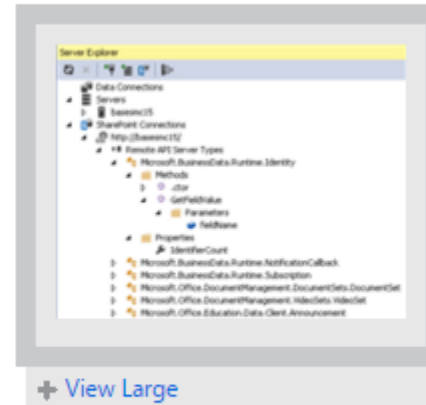
- Explore the remote API = available REST Endpoint
- Supports Office 365 and SharePoint 2016 ... and 2013 ☺

SPRemoteAPIExplorer Free

A Visual Studio server explorer extension to provide the ability to explore, discover and generate SharePoint 2016 or 2013 remote API code in REST and CSOM. Useful to see the new changes in SharePoint 2016. Exposes metadata for types, methods, parameters, properties and respo...

CREATED BY	Steve.Curran MVP (Steve Curran MVP)	UPDATED	4/4/2016
REVIEWS	★★★★★ (3) Review	VERSION	3.0.0
SUPPORTS	Visual Studio 2015, 2013, 2012	LICENSE	View
DOWNLOADS	Download (2,522)	SHARE	✉ t p g+ f
TAGS	tools, SharePoint Explorer, SharePoint Dev Tools, Visual Studio 2012, SharePoint 2013, App Model, Visual Studio 2013, Visual Studio 2015, SharePoint 2016		
		FAVORITES	Add to favorites

[Report abuse to Microsoft](#)



More From
Steve.Curran MVP

Explore the API ...

