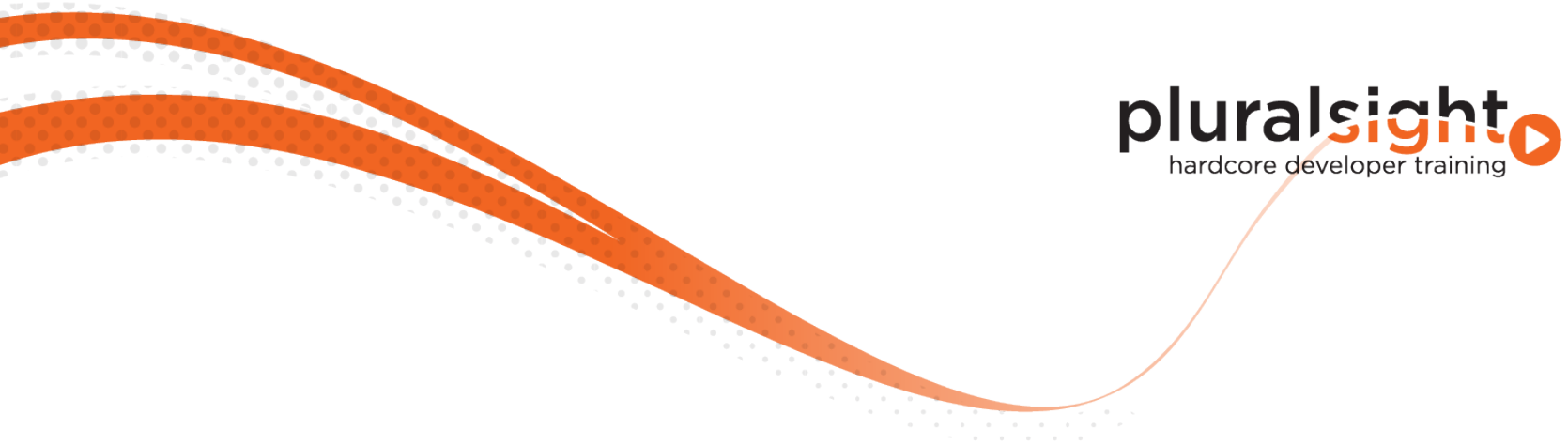
SharePoint 2013 Dev with CSOM and REST – Programming with the REST API

Rob Windsor
@robwindsor



pluralsight
hardcore developer training

Outline

- Things to Consider
- Data-binding
- Creating a List
- Creating List Items
- Updating List Items
- Web Proxy
- Connecting to the Host Web

REST API – Things to Consider

- **REST API does not completely conform to OData spec**
  - No metadata
- **OData does not support batched requests**
- **OData provider may limit number of items in response**
  - Need to check for continuation

OData Queries

- **Queries represented by query strings added to resource URL**

Option	Example
$select	_api/Web/Lists?$select=Title,ItemCount
$filter	_api/Web/Lists?$filter=(Hidden eq false)
$orderby	_api/Web/Lists?$orderby=ItemCount desc
$skip, $top	_api/Web/Lists?$skip=25&$top=10
$expand	_api/Web/Lists?$expand=Fields

Full documentation: http://www.odata.org/documentation/odata-v2-documentation/uri-conventions/#4_Query_String_Options (http://bit.ly/10dqevp)

Data-Binding

- Similar to the CSOM example… but easier
- REST API returns JSON objects
- JavaScript templating frameworks designed to work with JSON

```
<script id="products-template" type="text/x-jsrender">
       <ul>
           {{for #data}}
           <li>
               <b>{{>Title}}</b>
               <br />
               {{>UnitsInStock}} in stock at {{>UnitPrice}}
           </li>
           {{/for}}
       </ul>
   </script>
```

```
var template = jQuery("#products-template");
message.append(template.render(data.d.results);
```

Creating a List

- **Send POST to /_api/Web/Lists**
- **Message body has SP.List object with properties**
  - Fills same role as SP.ListCreationInformation object in CSOM
- **Must include Form Digest in headers**
  - Protects against replay attacks
  - Value of Form Digest available in hidden field on SharePoint page

```
var call = jQuery.ajax({
    url: _spPageContextInfo.webAbsoluteUrl + "/_api/Web/Lists",
    type: "POST",
    data: JSON.stringify({
        "__metadata": { type: "SP.List" },
        BaseTemplate: SP.ListTemplateType.tasks,
        Title: "Tasks"
    }),
    headers: {
        Accept: "application/json;odata=verbose",
        "Content-Type": "application/json;odata=verbose",
        "X-RequestDigest": jQuery("#__REQUESTDIGEST").val()
    }
});
```

Creating List Items

- Post to /_api/Web/Lists/getByTitle('<List Name>')/Items
- Type name is SP.Data.<List Name>ListItem

```
var call = jQuery.ajax({
    url: _spPageContextInfo.webAbsoluteUrl + "/_api/Web/Lists/getByTitle('Tasks')/Items",
    type: "POST",
    data: JSON.stringify({
        "__metadata": { type: "SP.Data.TasksListItem" },
        Title: "Sample Task",
        AssignedToId: userId,
        DueDate: due
    }),
    headers: {
        Accept: "application/json;odata=verbose",
        "Content-Type": "application/json;odata=verbose",
        "X-RequestDigest": jQuery("#__REQUESTDIGEST").val()
    }
});
```

Updating List Items

- **Send to /_api/Web/Lists/getByTitle('<List>')/Items(<Item Id>)**
- **Request type (X-Http-Method)**
  - Can update by sending PUT
    - All writable field values must be specified
  - Can update by sending POST
    - Set X-Http-Method to PATCH or MERGE
    - Only send field values that are changing
- **Concurrency (IF-MATCH)**
  - Item metadata includes etag which represents the version
  - Set IF-MATCH in header to etag value
    - Update will fail if item has been updated since read
  - SET IF-MATCH in header to *
    - Update will overwrite changes (if any)

Updating List Items

```
var call = jQuery.ajax({
    url: _spPageContextInfo.webAbsoluteUrl +
        "/_api/Web/Lists/getByTitle('Tasks')/Items(" + item.Id + ")",
    type: "POST",
    data: JSON.stringify({
        "__metadata": { type: "SP.Data.TasksListItem" },
        Status: "In Progress",
        PercentComplete: 0.10
    }),
    headers: {
        Accept: "application/json;odata=verbose",
        "Content-Type": "application/json;odata=verbose",
        "X-RequestDigest": jQuery("#__REQUESTDIGEST").val(),
        "IF-MATCH": item.__metadata.etag,
        "X-Http-Method": "PATCH"
    }
});
```

Web Proxy

- Use /_api/SP.WebProxy.invoke to make requests to external services
- Request made by SharePoint
- Response forwarded to calling code
- Apps must register target site as remote endpoint in app manifest
- Response size must not exceed 200 Kb

```
var call = jQuery.ajax({
    url: _spPageContextInfo.webAbsoluteUrl + "/_api/SP.WebProxy.invoke",
    type: "POST",
    data: JSON.stringify(
        {
            "requestInfo": {
                "__metadata": { "type": "SP.WebRequestInfo" },
                "Url": "http://services.odata.org/Northwind/Northwind.svc/Categories/?$format=json",
                "Method": "GET"
            }
        }),
    headers: {
        "Accept": "application/json;odata=verbose",
        "Content-Type": "application/json;odata=verbose",
        "X-RequestDigest": $("#__REQUESTDIGEST").val()
    }
});
```

Connecting to the Host Web

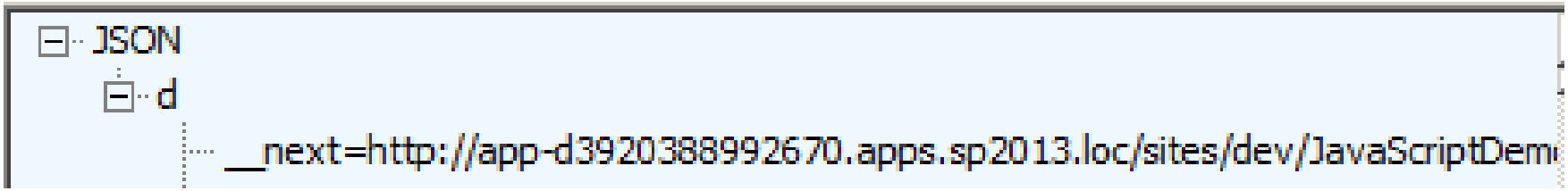
- Use SP.RequestExecutor to make request
  - Works like jQuery.ajax
- Use SP.AppContextSite to use host web context
- App must request permissions to read/write resources

```
var scriptbase = hostUrl + "/_layouts/15/";
jQuery.getScript(scriptbase + "SP.RequestExecutor.js", getItems);

function getItems() {
    var executor = new SP.RequestExecutor(appUrl);
    var url = appUrl + "/_api/SP.AppContextSite(@target)/Web/Lists/getByTitle(
        'Order Details')/Items?$select=Title&@target='" + hostUrl + "'";
    executor.executeAsync(
        {
            url: url,
            ...
            success: function (data) {
                var response = JSON.parse(data.body);
                message.append(String.format("Retrieved {0} items", response.d.results.length));
            }
        }
    );
}
```

OData Continuations

- OData provider may limit number of item in response
- Need to check for __next (JSON) or link element (AtomPub)
- Use URL to get next set of results




```
<link rel="next"
  href="http://sp2013found/sites/dev/_api/Web/Lists/getByTitle
  ('Order%20Details')/Items?%24skiptoken=Paged%3dTRUE%26p_ID%
  3d100" />
```

Summary

- Things to Consider
- Data-binding
- Creating a List
- Creating List Items
- Updating List Items
- Web Proxy
- Connecting to the Host Web