|  |
| --- |
|  |
| Office 365: Reporting web service query tool code sample |

Summary: This application uses Windows Presentation Foundation (WPF) and simple HTTP request/response classes to create an interactive Office 365 Reporting web service query tool in Visual Studio 2012. Users must have Office 365 administrative privileges.

# Description of the sample

The sample is a Windows Presentation Foundation application written in C# and Visual Studio 2012. The sample allows an Office 365 user with administrator permissions to easily construct and modify queries for the Office 365 Reporting web service. The tool is intended for developers to:

* Learn the overall program flow for building REST URI requests for the Reporting web service.
* Understand the structure, parameters and options of the REST URI.
* Obtain usable query strings for use in static-reporting dashboards.
* Learn how report data is returned in Atom and JSON (JavaScript Object Notation) formats.
* Record the full results of the headers, request, and response to help in troubleshooting.

# Prerequisites

The sample requires the following:

* Visual Studio 2012 with the .NET Framework 4.5 installed on your computer.
* Office 365 Enterprise subscription.
* Office 365 account with administrator privileges of Global, Password, Service, or User management administrator.

# Key components of the sample

The sample is a self-contained Windows Presentation Foundation application.

# Build the sample

1. Open Visual Studio 2012.
2. On the **File** menu, click **Open**, click **Project/Solution** and then navigate to the folder where you unzipped the O365RWS\_QueryTool files. Locate and select the file O365RWS\_QueryTool.sln. Click **Open**.
3. On the **Build** menu, click **Build All**. No errors or warning should be reported. If you receive any, the most likely cause is some part of the .NET Framework is missing.

If the sample builds with no errors and warnings, it is ready to run. No further configuration is needed.

# Run and test the sample

1. In Visual Studio 2012, open the solution file and press **F5** to start the application. The application opens as shown in the following screen shot. Notice that only the **1) Enter administrator credentials** button is enabled. The application will enable the other steps as you enter and verify information required for the Reporting web service calls to succeed.

**Figure 1. Enter administrator credentials button.**

Click **1) Enter administrator credentials**. The credentials dialog displays, as shown in the following screen shot:

**Figure 2. Enter Office 365 Credentials dialog box.**

Enter your administrator user name and password and click **OK**.

1. The query tool will now have the parts used in step 2 enabled, as shown in the following screen shot:

**Figure 3. Step 2 Check Endpoint.**

Verify that the domain name reads reports.office365.com. There are rare cases when you might need to change this, for example when your subscription is in a preview datacenter deployment. But for now, leave this at the default value. Click **Check endpoint**. It normally takes a few moments for the query tool to obtain the service description document using the credentials and domain endpoint. When the Reporting web service replies, the result will be displayed to the right of the **Check endpoint** button. A detailed result status is available if you hover the cursor over the short status, as shown in the following screen shot:

**Figure 4. Detailed result status.**

1. If the query tool retrieved the service document, step 3 will be enabled. The drop-down list will be enabled, allowing you to select a report. For this procedure, select the **MailFilterList** report item, as shown in the following screen shot.

**Figure 5. Choose a report drop-down list.**

If your administrator account is one of the lesser-privileged accounts, you might see one or more of the report names in the drop-down list disabled, shown in a lighter text. The tool parses the service description document retrieved during the Check endpoint step 2, and disables any reports that are unavailable to your administrator account. The following screen shot shows an example of how this looks, with the **MailDetailDlpPolicy** report disabled.

**Figure 6. MailDetailDlpPolicy disabled in the drop-down list.**

1. After you've selected a report, the query tool enables the parts needed to set the query options. When you select a report, the options box shows the default options for that report. The default options provided are only defined in the tool, and are intended to provide a list of parameters and fields for that report type. Click **Clear** to empty the options box, and click **Defaults** to reset them. You can either leave the original default options in place, or replace the options as shown in the following screen shot.

**Figure 7. Default options.**

The Reporting web service can return results as either Atom or JSON (JavaScript Object Notation) format. Use Atom format if you are using an XML parser, and use JSON if you're using the report information directly in a JavaScript-driven web page or browser window.

1. After you have the query options as you want them, the tool should look like the following screen shot:

**Figure 8. Query options screen shot.**

Click **5) Generate REST URL**.

The beginning of the fully-formed URL that will be passed to the Reporting web service is displayed below the button, as shown in the following screen shot. Click the **Copy REST URL** button to copy the REST URL to the Windows clipboard.

**Figure 9. REST URL screen shot.**

1. You are now ready to submit the query to the Reporting web service. You've verified your credentials and that the service is available, you've set your options and format, and constructed the full REST URL. Click **6) Send Request**, as shown in the following screen shot.

**Figure 10. Send Request button.**

The report will take a few moments to return. Some of the delay is consumed by the tool formatting the Atom or JSON data for easier reading. If the request is successful, the top three lines of the display will look like the following screen shot (shown with the lines selected for clarity):

**Figure 11. Successful results screen shot.**

1. To append the results to a log file, click the **Append results** button at the bottom of the tool, as shown in the following screen shot.

**Figure 12. Append results button.**

1. To view the contents of the log file, navigate to the **My Documents** folder and locate the file named as shown on the button, with today's date at the end. The file contains all the results you added, in chronological order (earliest report is first in the file). There are five sections to each report record:
   * Basic information about the report request and whether it was successful.
   * The full request URL.
   * The headers sent in the HTTP request.
   * The HTTP headers returned by the Reporting web service.
   * The formatted results. Note that if you select JSON format, the formatting may have introduced syntactical errors into the data. To use the JSON data properly, you need to take the stream and use it in a browser.

The following is an example of the file, with lines removed for clarity.

|  |  |
| --- | --- |
|  |  |
| ==========================================================  ==== SUCCESSFUL ====  ==========================================================  Type: MailFilterList  Request sent: 2/5/2013 8:22:59 PM  Response received: 2/5/2013 8:23:22 PM  Response time: 23004 ms  User name: user@example.onmicrosoft.com  HTTP response: 200 OK  ==== REQUEST URL =========================================  https://reports.office365.com/ecp/reportingwebservice/reporting.svc/MailFilterList?  $select=Display,SelectionTarget&$filter=SelectionTarget%20eq%20'Actions'&$format=Atom  ==== HEADERS SENT ========================================  DataServiceVersion:2.0  MaxDataServiceVersion:2.0  Accept-Language:EN-US  X-RWS-Version:2013-V1  Authorization:Basic dXNlcnR3b0BleG9kZW1vMS5vbm1pY3Jvc29mdC5jb206ZXJkN0pVSjA=  Host:reports.office365.com  ==== HEADERS RCVD ========================================  Pragma:no-cache  request-id:2365e97e-f42b-4774-b238-5e673c857df8  X-CalculatedBETarget:server.domain.com  X-Content-Type-Options:nosniff  DataServiceVersion:2.0;  X-RWS-Version:2013-V1  X-DiagInfo:SN2PR03MB030  X-BEServer:SN2PR03MB030  X-UA-Compatible:IE=10  X-FEServer:BLUPR03CA003  Content-Length:14402  Cache-Control:no-cache, no-store  Content-Type:application/atom+xml;type=feed;charset=utf-8  Date:Wed, 06 Feb 2013 04:23:21 GMT  Expires:-1  Set-Cookie:X-BackEndCookie=S-1-5-21-2412119505-3560892057-1500426294-....  Server:Microsoft-IIS/7.5  X-AspNet-Version:4.0.30319  X-Powered-By:ASP.NET  ==== RESULTS DOCUMENT ====================================  <?xml version="1.0" encoding="utf-8"?>  <feed xml:base="https://reports.office365.com/ecp/ReportingWebService/Reporting.svc/"  xmlns="http://www.w3.org/2005/Atom" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">  <id>https://reports.office365.com/ecp/reportingwebservice/reporting.svc/MailFilterList</id>  <title type="text">MailFilterList</title>  <updated>2013-02-06T04:23:22Z</updated>  <link rel="self" title="MailFilterList" href="MailFilterList" />  <entry>  <id>https://reports.office365.com/ecp/ReportingWebService/Reporting.svc/MailFilterList('Actions')</id>  <category term="TenantReporting.MailFilterListReport"  scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" />  <link rel="edit" title="MailFilterListReport" href="MailFilterList('Actions')" />  <title />  <updated>2013-02-06T04:23:22Z</updated>  <author>  <name />  </author>  <content type="application/xml">  <m:properties>  <d:SelectionTarget>Actions</d:SelectionTarget>  <d:Display>AddBccRecipient</d:Display>  </m:properties>  </content>  </entry>  [[ LINES REMOVED FOR CLARITY ]]  <entry>  <id>https://reports.office365.com/ecp/ReportingWebService/Reporting.svc/MailFilterList('Actions')</id>  <category term="TenantReporting.MailFilterListReport"  scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" />  <link rel="edit" title="MailFilterListReport" href="MailFilterList('Actions')" />  <title />  <updated>2013-02-06T04:23:22Z</updated>  <author>  <name />  </author>  <content type="application/xml">  <m:properties>  <d:SelectionTarget>Actions</d:SelectionTarget>  <d:Display>StopRuleProcessing</d:Display>  </m:properties>  </content>  </entry>  </feed>  ==========================================================  ========================================================== | |

# Troubleshooting

1. Entering your user credentials
   * The credentials dialog box displays an error message box if the user name is not detected as a standard SMTP email address, also known as a User Principal Name (UPN). International characters are not supported for user names in this sample.
   * The user name and password fields may not be left blank.
2. Checking the Reporting web service endpoint
   * The status label, just to the right of the Check endpoint button provides additional information about the successful or failed request. Hover the cursor over the label to read the additional details.
   * If the endpoint check fails, make sure the user name and password are correct, and that the account has the required permissions. The best way to check this is to log onto the admin center at <https://portal.microsoftonline.com/default.aspx> and verify that account can access the reports interactively.
   * If no response is received, it is possible that the Reporting web service is temporarily unavailable.
   * Remember to read the tooltip on the status label; it contains specific results information that may help you solve the problem.
3. Selecting the report
   * If the **MailDetailDlpPolicy** is disabled, but your account is a Global or Compliance administrator, double-check the user name and password, and verify you can see that report in the admin console.
   * Note that although the **MailDetail** report is present in the drop-down list, the report will always return no results. As of early February 2013, it is disabled in the datacenter, but the service description document erroneously reports that it is present.
4. Setting the report options
   * Many failed report requests are likely caused by incorrectly setting the report options. The syntax of complex $filter clauses is sometimes difficult to get right.
   * Check that you are using the legal ODATA comparison operators (for example, "eq" for equals, "lt" for less than, "ge" for greater than or equal, and so on).
   * The only places where the equals sign (=) is allowed are immediately after the option name (for example, $filter=, $select=).
   * Be sure you have the column names entered correctly.
   * If you're using StartDate and EndDate in a $filter option to restrict the reporting duration, use the long-date format, and remember to cast it into the datetime type. For example: $filter=StartDate eq datetime'2013-01-01T00:00:00Z' and EndDate eq datetime'2013-01-31T11:59:59T'
   * Put each ODATA query option on a separate line; the application will combine them and add the & separators when it builds the REST URL.
   * There is no need to include the StartDate and EndDate fields in the $select= option. The sample includes them so you have a complete and accurate list of the field names.
5. Generate the REST URL
   * The displayed string is cut off at the right. Use the Copy REST URL button to paste the URL into a browser or editor to view the full REST request URL. The full REST URL is included in the log file.
6. Send the request
   * The sample catches exceptions thrown by the HTTP request, but network conditions can also cause the sample to act unexpectedly. But normally the results of failed requests are properly recorded in the transcript area.
   * The most common reasons for failed requests involve the options. Check, double-check and triple check the options. By saving the results to the log file, you can view the full REST URL, including the URI encoding performed on the options.
7. Append the results to the log file
   * Errors can occur if you have no disk space or do not have permissions to write in the My Documents folder.
8. View the results log file
   * If results are not added to the log file, check whether you have the file open in an editor that locks the file. Windows Notepad locks files in this manner. Other editors do not lock the file, and so results can be added even while you're reading the file.

# Change log

|  |  |
| --- | --- |
| **Version** | **Date** |
| First version | February, 28 2013 |

# Related content