



AWQMS

Ambient Water Quality Monitoring System

RESTful Post and Put Web Services User Guide

Contents

Overview	2
How to Call Web Services	3
Using custom application code to create/update activities and add new or refresh results.....	3
Using Fiddler (for web service testing)	5
JSON Members:.....	9
Output JSON:.....	11

Overview

RESTful POST/PUT web services are used to update/insert activities and add new results to an activity in AWQMS.

The body of the service request describes the activity and its results as JSON (Java Script Object Notation).

The POST service is used to create a new activity and results.

The PUT service is used to update an activity and replace or add new results to that activity.

- If the service is called with only the activity and organization identifiers populated (on the activity) and includes one or more results, then the activity will not be updated. But the results will be added to the existing activity. If the boolean parameter "DeleteExistingResults" = "true" then all existing results (on that activity) will be deleted before adding any new results that are provided.
- If the service is called with any other activity fields populated (other than the activity and organization identifiers), then the activity fields will be updated. See the GET service call for the list of activity fields that can be updated. All other activity-level fields, that are not part of the GET service, will be ignored.

These services can only be called successfully from authenticated users. No public versions of these services exist. An AWQMS administrator will need to setup an account for use with these services. To POST or PUT activities and results to an organization, the caller must include an encrypted 88-character key in the header (of the request) which will identify a user account that has rights to that organization's private data.

Output from a web service call is JSON. If the call is successful (i.e., no errors), then the activity ID will be sent back in JSON and the header location will be the URI for the GET request for the activity and results. If there are errors, a list of error messages in JSON will be returned in the body.

The web services have the following pattern:

Post: [AWQMS URL]/api/ResultsVer1/

Put: [AWQMS URL]/api/ResultsVer1/[Activity Identifier]

NOTE: Avoid '/' in the activity identifier

For Example, the AWQMS Cloud Environment is available at (<http://awqms.goldsystems.com>). So an example of a web service call (to that environment) with an activity identifier "03-14-00-2715-PCB-1999FALL" is as follows:

`http://awqms.goldsystems.com/api/ResultsVer1/03-14-00-2715-PCB-1999FALL`

How to Call Web Services

This section outlines some of the methods for calling/testing the AWQMS web services:

Using custom application code to create/update activities and add new or refresh results

You will need the following two pieces of information:

1. Encryption Key: This 88-character string is a unique identifier for an individual user account within the AWQMS system. It used to encrypt the header data being sent. This is used to determine the rights that a specific web service caller has to private data within AWQMS. This Encryption Key should be kept private and only used by a specific user (or group of users) that share a set of rights within AWQMS.
For AWQMS Administrators, the steps to create an encryption key for a user are as follows:
 - a. *On the User Detail page (for a particular user) find the "Private Encryption Key:" label on the page. If a key already exists, click the "Show" button to reveal the key (and provide this key to a user who needs it). Otherwise click the "Create New Key" button, then "Ok" to generate a new key.*
2. User Login Name: This is encrypted in the header and will be used for authentication.

You can obtain a Private Encryption Key for the AWQMS-Cloud Environment (managed by Gold Systems) by sending a request to awqms@goldsystems.com.

The custom header elements are as follows:

1. X-UserID
 - a. The caller's "User Login Name" to determine rights to private data
2. X-Stamp
 - a. Timestamp when the caller made the request. This must be UTC Time (i.e. Greenwich Mean Time), so you must convert from your local time zone.
 - b. Format is (mm/dd/yyyy hh:mi:ss AM)
3. X-Signature
 - a. Signature is made up of the following pieces of information (concatenated together as a single string value):
 - i. User Login Name
 - ii. Timestamp
 - iii. URI
 - iv. Request Method/Action
 - b. Signature is then encrypted using the HMAC-SHA256 encryption algorithm and placed in the header element.

The following example demonstrates how to call an AWQMS Web Service with encrypted header information to access private data. This example uses C#, but you can use any language that provides an HMAC-SHA256 encryption algorithm:

```
using System;
```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Security.Cryptography;
using System.Net;

namespace WebServiceConsole {

    class Program {

        static void Main (string[] args) {
            // The two keys below need to be obtained from Gold Systems. These keys are for illustration purposes only.
            string userLoginName = "jdoe";
            string encryptionKey = "XyUjxhyrLzkczlNaEsgtgB/6RWuxNNfGQTBXVakF730k6IZkIXcAbccA+i9uE1HcP2leWqKcvuJENJCDgN/g==";

            using (WebClient client = new WebClient()) {
                try {
                    // The URI of the result webservice with parameters
                    Uri uri = new Uri("http://awqms.goldsystems.com/api/ResultsVer1/");
                    string activity = "{\"ActivityIdentifier\":\"00-TEST\",\"ActivityType\":\"Sample-
Routine\",\"BottomDepthHeight\":\"34\",\"BottomDepthHeightUnit\":\"ft\",\"DepthAltitudeReferencePoint\":\"Ref
Point\",\"DepthHeight\":\"123\",\"DepthHeightUnit\":\"\",\"MediaName\":\"Water\",\"MediaSubdivisionName\":\"Groundwater\",
\"MonitoringLocationIdentifier\":\"CHARM-G2\",\"OrganizationIdentifier\":\"TESTRICK\",
\"Projects\":{\"ProjectIdentifier\":\"CHARM
01\"}},\"Results\":{\"AnalyticalMethodContext\":\"TESTRICK\",\"AnalyticalMethodIdentifier\":\"(101)TEST\",\"Characteristi
cName\":\"(101) char used\",\"Comment\":\"Result
Comment\",\"DataLoggerLine\":\"14\",\"DepthHeight\":\"12\",\"DepthHeightUnit\":\"ft\",\"DetectionCondition\":\"\", \"Method
Speciation\":\"as H\",\"ParticleSizeBasis\":\"Size
Basis\",\"QualifierCode\":\"BJHQC\",\"ResultUnit\":\"mg/l\",\"ResultValue\":\".0025\",\"SampleFraction\":\"Total\", \"Stati
sticalBaseCode\":\"15 Pctl\",\"Status\":\"Final\", \"TemperatureBasis\":\"43 deg C\",
\"TimeBasis\":\"1
Day\", \"ValueType\":\"Actual\", \"WeightBasis\":\"Wet\"},{\"AnalyticalMethodContext\":\"TESTRICK\", \"AnalyticalMethodIdenti
fier\":\"Aug-12method\", \"CharacteristicName\":\"Fluoride\",
\"Comment\":\"\", \"DataLoggerLine\":\"\", \"DepthHeight\":\"\", \"DepthHeightUnit\":\"\", \"DetectionCondition\":\"\", \"Metho
dSpeciation\":\"\", \"ParticleSizeBasis\":\"\", \"QualifierCode\":\"HICC\", \"ResultUnit\":\"mg/l\",
\"ResultValue\":\".0023\", \"SampleFraction\":\"Fixed\", \"StatisticalBaseCode\":\"\", \"Status\":\"Final\", \"TemperatureBasi
s\":\"\", \"TimeBasis\":\"\", \"ValueType\":\"Actual\", \"WeightBasis\":\"\",
{\"AnalyticalMethodContext\":\"USEPA\", \"AnalyticalMethodIdentifier\":\"PCB-003\", \"CharacteristicName\":\"Polychlorinated
biphenyls\", \"Comment\":\"STORET ISN = 48485\", \"DataLoggerLine\":\"\",
\"DepthHeight\":\"\", \"DepthHeightUnit\":\"\", \"DetectionCondition\":\"\", \"MethodSpeciation\":\"\", \"ParticleSizeBasis\":
\"\", \"QualifierCode\":\"\", \"ResultUnit\":\"ug/l\", \"ResultValue\":\"1050.00\",
\"SampleFraction\":\"Total\", \"StatisticalBaseCode\":\"\", \"Status\":\"Final\", \"TemperatureBasis\":\"\", \"TimeBasis\":\"\",
\"ValueType\":\"Actual\", \"WeightBasis\":\"\", {\"AnalyticalMethodContext\":\"USEPA\",
\"AnalyticalMethodIdentifier\":\"61\", \"CharacteristicName\":\"Potassium\", \"Comment\":\"\", \"DataLoggerLine\":\"\", \"Dept
hHeight\":\"\", \"DepthHeightUnit\":\"\", \"DetectionCondition\":\"\", \"MethodSpeciation\":\"\",
\"ParticleSizeBasis\":\"\", \"QualifierCode\":\"\", \"ResultUnit\":\"mg/l\", \"ResultValue\":\".4\", \"SampleFraction\":\"Diss
olved\", \"StatisticalBaseCode\":\"\", \"Status\":\"Final\", \"TemperatureBasis\":\"\",
\"TimeBasis\":\"\", \"ValueType\":\"Actual\", \"WeightBasis\":\"\"}}, \"SampleCollectionEquipmentComment\":\"Equipment
Comment\", \"SampleCollectionEquipmentName\":\"Miscellaneous (Other)\",
\"SampleCollectionMethodIdentifier\":\"SP-001\", \"StartDate\":\"2017-06-27\", \"StartTime\":\"06:30:00
pm\", \"StartTimeZone\":\"MDT\", \"TopDepthHeight\":\"3\", \"TopDepthHeightUnit\":\"ft\", \"DeleteExistingResults\":\"true\"}";

                    client.Headers.Add("X-UserID", userLoginName);
                    string timeStamp = DateTime.UtcNow.ToString();
                    client.Headers.Add("X-Stamp", timeStamp);
                    // The encrypted data must be in this order and format
                    string data = String.Format("{0}{1}{2}{3}", userLoginName, timeStamp, uri.ToString(), "POST");
                    byte[] signature = Encoding.UTF8.GetBytes(data);
                    using (HMACSHA256 hmac = new HMACSHA256(Convert.FromBase64String(encryptionKey))) {
                        byte[] signatureBytes = hmac.ComputeHash(signature);
                        client.Headers.Add("X-Signature", Convert.ToBase64String(signatureBytes));
                    }
                    client.Headers.Add("Content-Type", "application/json");
                    // Process the response
                    var response = client.UploadString(uri, "POST", activity);
                    Console.WriteLine(response);
                    Console.Read();
                }
            }
            catch (WebException we) {
                string resp = new StreamReader(we.Response.GetResponseStream()).ReadToEnd();
                if (!string.IsNullOrEmpty(resp)) {

```

```
        Console.WriteLine(resp);
        Console.ReadKey();
    }
    else {
        if (we.InnerException != null) {
            Console.WriteLine(we.InnerException.Message);
        }
        Console.WriteLine(we.Message);
        Console.WriteLine(((HttpWebResponse) ((WebException) we).Response).StatusDescription);
        Console.Read();
    }
}
catch (Exception e) {
    if (e.InnerException != null) {
        Console.WriteLine(e.InnerException.Message);
    }
    Console.WriteLine(e.Message);
    Console.WriteLine(((HttpWebResponse) ((WebException) e).Response).StatusDescription);
    Console.Read();
}
}
```

Using Fiddler (for web service testing)

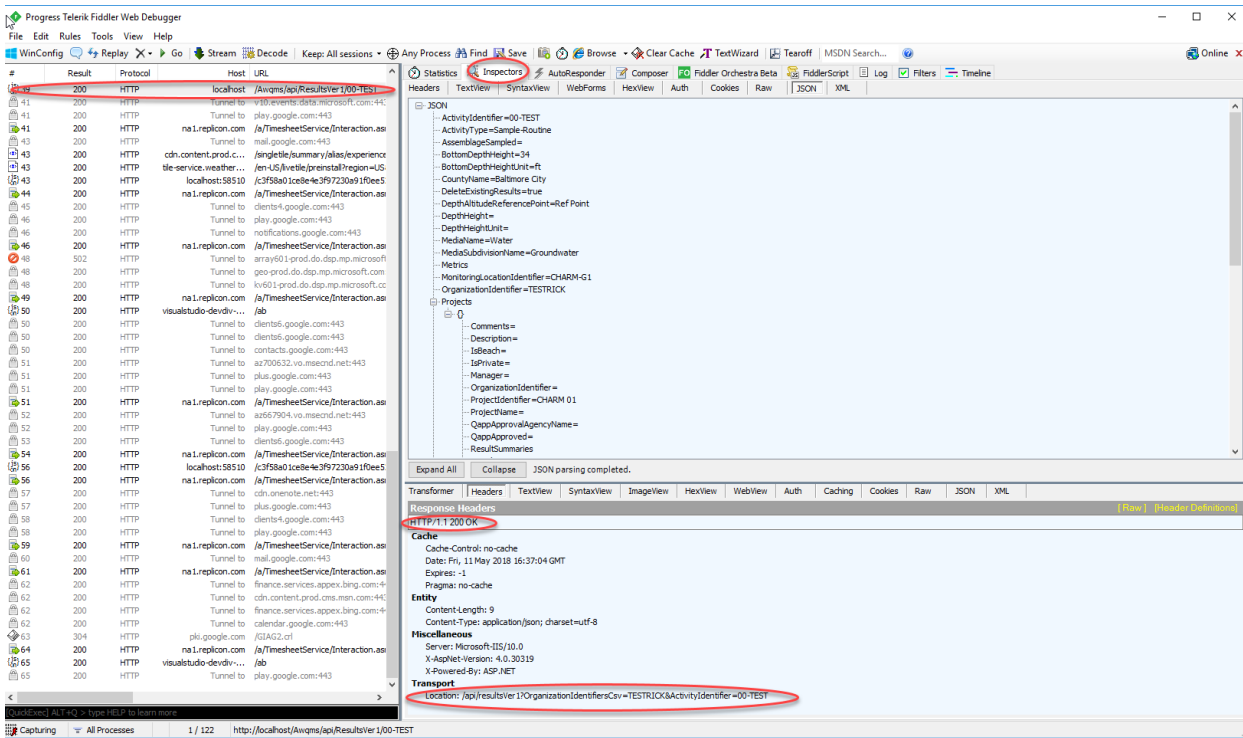
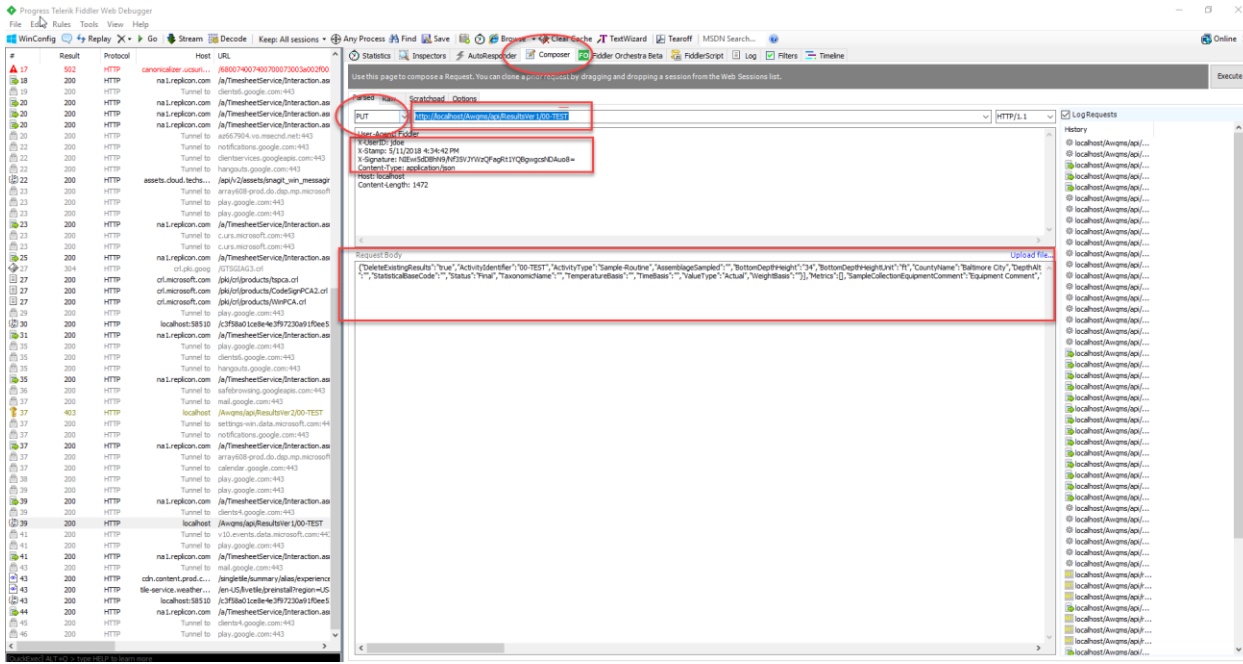
Another way to test the services is to use Fiddler www.fiddler2.com/fiddler2.

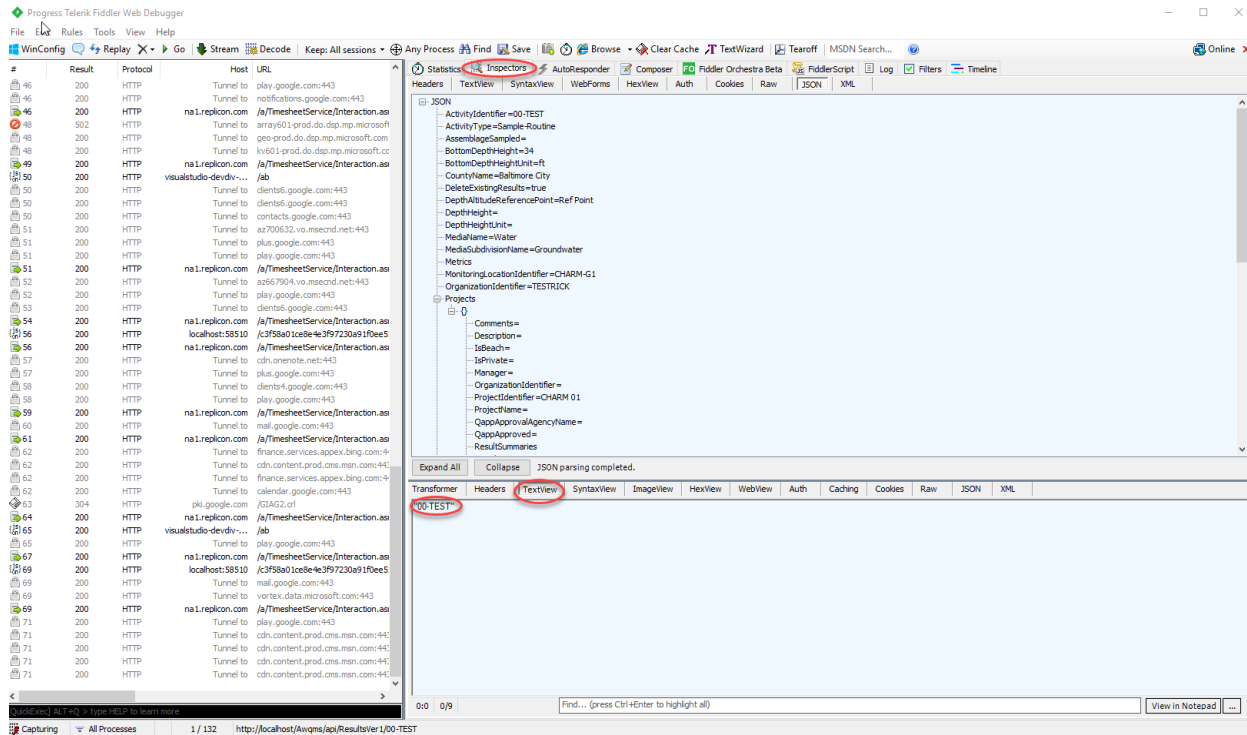
Using Fiddler:

- Start Fiddler
- Click Composer tab.
- Select POST/PUT in dropdown list.
- Copy/Paste URL (below) to test in dropdown list element.
- Select HTTP/1.1
- In header section, enter: “X-UserID, X-Stamp, X-Signature, Content-Type: text/json” (do not enter the quotes) for JSON
 - NOTE: It will be necessary to create an X-Signature outside of fiddler in some custom code similar to the above example code;
- Click Execute Button
- In left-pane, find the URL entry and double-click the entry.
- Output will display in the lower right-pane in the Request Body section.
- See screen shots below

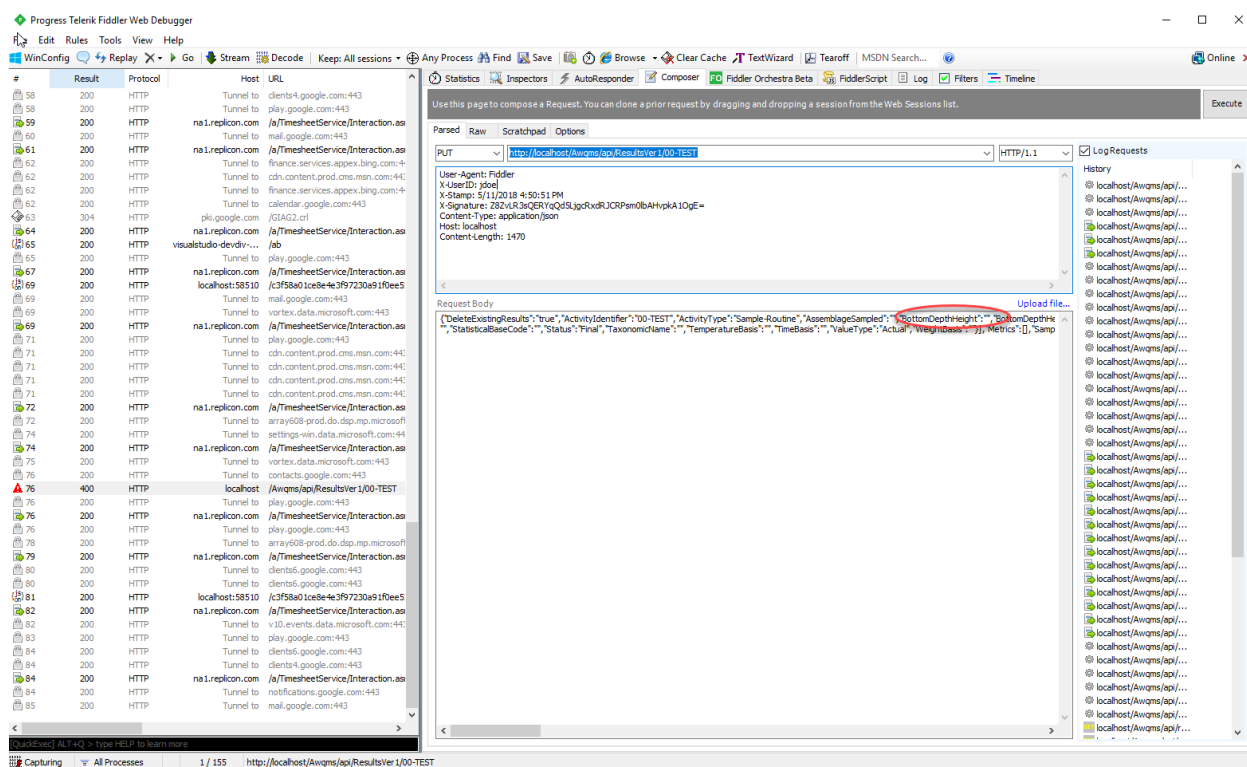
Note: PUT URL, HTTP/1.1, Content-Type text/json

Below is an example of a call with no errors:





The following is an example of an error in the activity (e.g. missing Bottom Depth Height measurement):



The screenshot shows the Progress Telerik Fiddler Web Debugger interface. The left pane displays a list of captured HTTP requests. The selected request is a 400 Bad Request from localhost to /Awqms/api/ResultsVer/100-TEST. The right pane shows the JSON response body, which contains an error message: "Bottom Depth/Height is required when a Top Depth/Height is provided".

#	Result	Protocol	Host	URL
53	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
54	200	HTTP	localhost:58510	/c3f58a01ce8e4e3f97230a91f0ee5
56	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
57	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
58	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
59	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
60	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
62	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
62	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
62	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
63	304	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
64	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
65	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
66	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
67	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
69	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
69	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
71	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
71	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
71	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
71	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
72	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
72	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
74	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
75	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
76	400	HTTP	localhost	/Awqms/api/ResultsVer/100-TEST
76	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
78	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
79	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
80	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
80	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
81	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
82	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
82	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as

JSON Response Body:

```
{
  "ActivityIdentifier": "00-TEST",
  "ActivityType": "Sample Routine",
  "AssemblageSampled": true,
  "BottomDepthHeight": "ft",
  "BottomDepthHeightUnit": "ft",
  "CountyName": "Baltimore City",
  "DeleteExistingResults": true,
  "DepthAltitudeReferencePoint": "Ref Point",
  "DepthHeight": "ft",
  "DepthHeightUnit": "ft",
  "MediaName": "Water",
  "MediaSubdivisionName": "Groundwater",
  "Metrics": {
    "MonitoringLocationIdentifier": "CHARM-01",
    "OrganizationIdentifier": "TETRUCK"
  },
  "Projects": {
    "Comments": " ",
    "Description": " ",
    "IsBeach": " ",
    "IsPrivate": " ",
    "Manager": " ",
    "OrganizationIdentifier": " ",
    "ProjectIdentifier": "CHARM 01",
    "ProjectName": " ",
    "QappApprovalAgencyName": " ",
    "QappApproved": " ",
    "ResultsSummaries": " "
  }
}
```

Error Message: [{"Message": "Bottom Depth/Height is required when a Top Depth/Height is provided", "Severity": "Error", "Context": ""}], [{"Message": "Bottom Depth/Height Value is required when Bottom Depth/Height Unit is provided", "Severity": "Error", "Context": ""}]]

The screenshot shows the Progress Telerik Fiddler Web Debugger interface. The left pane displays a list of captured HTTP requests. The selected request is a 400 Bad Request from localhost to /Awqms/api/ResultsVer/100-TEST. The right pane shows the JSON response body, which contains an error message: "Bottom Depth/Height is required when a Top Depth/Height is provided".

#	Result	Protocol	Host	URL
63	304	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
64	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
65	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
66	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
67	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
69	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
69	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
71	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
71	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
71	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
71	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
72	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
72	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
74	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
75	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
76	400	HTTP	localhost	/Awqms/api/ResultsVer/100-TEST
76	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
78	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
79	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
80	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
80	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
81	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
82	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
82	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
83	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
84	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
84	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
85	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
87	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
87	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
88	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
88	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
89	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
89	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
91	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as
91	200	HTTP	na1.replicon.com	/a/TimesheetService/Interaction.as

JSON Response Body:

```
{
  "ActivityIdentifier": "00-TEST",
  "ActivityType": "Sample Routine",
  "AssemblageSampled": true,
  "BottomDepthHeight": "ft",
  "BottomDepthHeightUnit": "ft",
  "CountyName": "Baltimore City",
  "DeleteExistingResults": true,
  "DepthAltitudeReferencePoint": "Ref Point",
  "DepthHeight": "ft",
  "DepthHeightUnit": "ft",
  "MediaName": "Water",
  "MediaSubdivisionName": "Groundwater",
  "Metrics": {
    "MonitoringLocationIdentifier": "CHARM-01",
    "OrganizationIdentifier": "TETRUCK"
  },
  "Projects": {
    "Comments": " ",
    "Description": " ",
    "IsBeach": " ",
    "IsPrivate": " ",
    "Manager": " ",
    "OrganizationIdentifier": " ",
    "ProjectIdentifier": "CHARM 01",
    "ProjectName": " ",
    "QappApprovalAgencyName": " ",
    "QappApproved": " ",
    "ResultsSummaries": " "
  }
}
```

Error Message: [{"Message": "Bottom Depth/Height is required when a Top Depth/Height is provided", "Severity": "Error", "Context": ""}], [{"Message": "Bottom Depth/Height Value is required when Bottom Depth/Height Unit is provided", "Severity": "Error", "Context": ""}]]

Cache:

- Cache-Control: no-cache
- Date: Fri, 11 May 2018 16:51:46 GMT
- Expires: -1
- Pragma: no-cache

Entity:

- Content-Length: 241
- Content-Type: application/json; charset=utf-8

Miscellaneous:

- Server: Microsoft-IIS/10.0
- X-AspNet-Version: 4.0.30319
- X-Powered-By: ASP.NET

POST/PUT Web Service

URL: [AWQMS base URL]/api/ResultsVer1

AWQMS-Cloud Example: <http://awqms.goldsystems.com/api/ResultsVer1>

JSON Members:

- ActivityIdentifier
 - Required
 - (e.g. "00-TEST")
- ActivityType
 - Required if creating new activity
 - (e.g. "Sample-Routine")
- BottomDepthHeight
 - Conditionally Required
 - (e.g. "34")
- BottomDepthHeightUnit
 - Conditionally Required
 - (e.g. "ft")
- DeleteExistingResults
 - Required
 - (e.g. "true/false")
- DepthAltitudeReferencePoint
 - (e.g. "Reference Point")
- DepthHeight
 - Conditionally Required
 - (e.g. "12.5")
- DepthHeightUnit
 - Conditionally Required
 - ("m")
- MediaName
 - Required if creating new activity
 - (e.g. Water)
- MediaSubdivisionName
 - (e.g. Groundwater)
- MonitoringLocationIdentifier
 - Conditionally Required
 - (e.g. LOCATION-G1)
- OrganizationIdentifier
 - Required
 - (e.g. TESTORG)
- Projects
 - ProjectIdentifier
 - Required
 - (e.g. PROJECT2018)

- Results
 - AnalyticalMethodContext
 - Conditionally Required
 - (e.g. USEPA)
 - AnalyticalMethodIdentifier
 - Conditionally Required
 - (e.g. "614.1")
 - CharacteristicName
 - Required
 - (e.g. "Conductivity")
 - Comment
 - DataLoggerLine
 - Conditionally Required
 - (e.g. "68-Line 68")
 - DepthHeight
 - Conditionally Required
 - (e.g. "76")
 - DepthHeightUnit
 - Conditionally Required
 - (e.g. "ft")
 - DetectionCondition
 - Conditionally Required
 - (e.g. "Not Detected")
 - MethodSpeciation
 - (e.g. "as N")
 - ParticleSizeBasis
 - (e.g. "< 0.0625 mm")
 - QualifierCode
 - (e.g. "CJQC")
 - ResultUnit
 - Conditionally Required
 - (e.g. "umho")
 - ResultValue
 - Conditionally Required
 - (e.g. "1")
 - SampleFraction
 - Conditionally Required
 - (e.g. "Total")
 - StatisticalBaseCode
 - (e.g. "Median")
 - Status
 - Required Member
 - (e.g. "Final")
 - TemperatureBasis
 - (e.g. "110 deg C")

- TimeBasis
 - (e.g. "98 Day")
- ValueType
 - Conditionally Required
 - (e.g. "Actual")
- WeightBasis
 - (e.g. "Ash Weight")
- SampleCollectionEquipmentComment
- SampleCollectionEquipmentName
 - (e.g. "Miscellaneous (Other)")
- SampleCollectionMethodIdentifier
 - Conditionally Required
 - (e.g. "SP-001")
- StartDate
 - Required
 - (e.g. "2017-06-27")
- StartTime
 - Conditionally Required
 - (e.g. "06:30:00 pm")
- StartTimeZone
 - Conditionally Required
 - (e.g. "MDT")
- TopDepthHeight
 - Conditionally Required
 - (e.g. "3")
- TopDepthHeightUnit
 - Conditionally Required
 - (ft)

Output JSON:

Success:

- "00-TEST"
- Response Header:
 - Location
 - /api/resultsVer1?OrganizationIdentifiersCsv=TESTRICK&ActivityIdentifier=00-TEST

Failure:

- [{"Message": "Bottom Depth/Height is required when a Top Depth/Height is provided", "Severity": "Error", "Context": ""}, {"Message": "Bottom Depth/Height Value is required when Bottom Depth/Height Unit is provided", "Severity": "Error", "Context": ""}]