



GENESIS64 REST API Integration



APPLICATION NOTE

October 2015

Description: An introduction on how to use the GENESIS64 REST API integration for reading data in and out of the system

Introduction

REST (Representational State Transfer) is a software architecture style consisting of guidelines and best practices for creating scalable web services. REST has gained widespread acceptance across the World Wide Web as a simpler alternative to SOAP and WSDL-based web services. RESTful systems typically, but not always, communicate over the Hypertext Transfer Protocol with the same HTTP verbs (GET, POST, PUT, DELETE, etc.) used by web browsers to retrieve web pages and send data to remote servers.

GENESIS64 support data accessibly through REST API. It allows users and third party applications to easily interface with the system. GENESIS64 exposes ICONICS runtime data with ODataConnector that can be marked as *REST OUT*. GENESIS64 can read REST data with Web Services Point Manager that can be marked as *REST IN* as shown below.

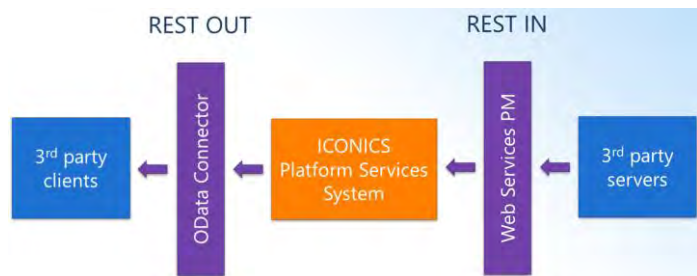


Figure 1 - REST Support in Platform Services

Complete architecture of REST API integration:

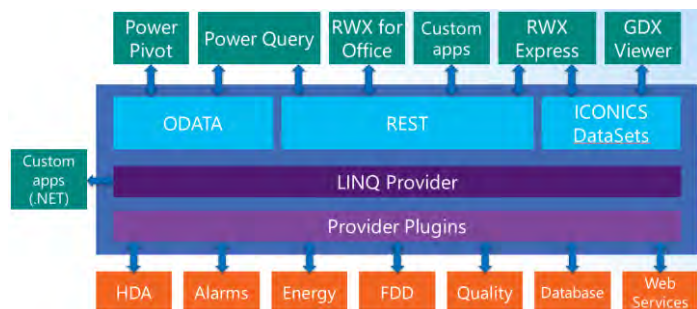


Figure 2 - Architectural Diagram

NOTE: LINQ provider can be used as an alternative to FWX Client from custom .NET apps

REST IN

Most Services on the web expose a REST API. As an example programmableweb.com page has a list of more than 13000 REST APIs. GENESIS64 can connect to the REST services in Workbench Desktop. We support services that return JSON data.

NOTE: In this example, the Web Service used is from Weather Underground. This service requires that you create a account before using it.

<http://www.wunderground.com/weather/api/d/pricing.html>

1. In Workbench Desktop browse to **Data Connectivity** → **Web Services** → **My Web Services**
2. Right click and select **Add REST Service**

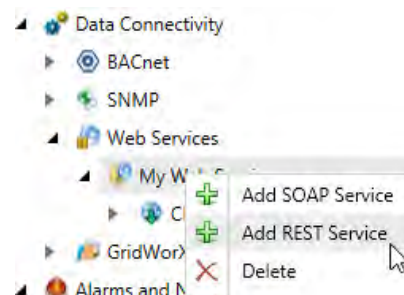


Figure 3 - Adding a REST service

3. Enter a **REST Service Name** and the web service **Base URL** such as `http://api.wunderground.com/api/{key}/` shown in the figure below

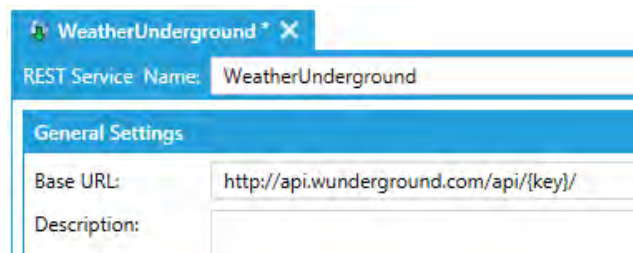


Figure 4 - Name and Base URL

4. Specify the URL **parameter** with a *URL segment Kind* and a Value which is the Key you received from Weather Underground (not shown below).

Default Parameters		
Name	Value	Kind
+ Click here to add new item		
key	[REDACTED]	URL Segment

Figure 5 - Default Parameters

- Select **none authentication** and click **Apply** to save the changes
- Right click on newly added Web Service and select **Add Web Method**

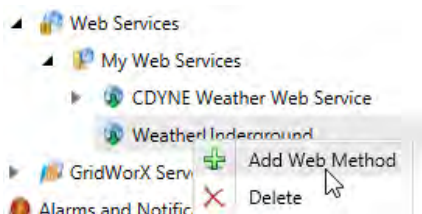


Figure 6 - Adding a web method

- Enter a **Name** for the method such as *Condition* and use **Relative URL** such as *conditions/q/{state}/{city}.json* shown in the figure below

WeatherUnderground fix Condition * X

Web Method

General Settings

Name:

Description:

Relative URL:

Http Method:

Refresh Type:

Refresh Period: (seconds)

Figure 7 - Web Method dialog

- Enter the method parameters with a **Kind** of URL Segment as shown in the figure below

Parameters		
Name	Value	Kind
+ Click here to add new item		
city	Foxboro	URL Segment
state	MA	URL Segment

Figure 8 - Entering URL Segment parameters

- Click on **Load Schema** button to retrieve the Return Schema and click **Apply** to save the changes
- In **GraphWorX64** you can now browse to *My Computer* → *Data Sources* → *Web Services* → *My Web Services* → *WeatherUnderground* → *Condition* → *Return Value* → *current_observation* → *current_observation.temp_f* from a Process Point to view the current temperature as shown below in runtime

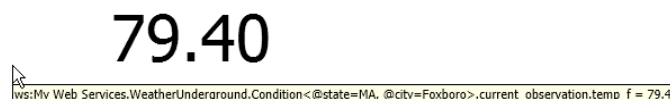


Figure 9 - Result from the Weather Underground service

REST OUT

ODataConnector is installed with GENESIS64. There is and *ICONICS OData Connector Point Manager* service with automatic startup type that enables REST and OData endpoints. All data are exposed as JSON. Interface is currently read-only.

NOTE: OData endpoints are specialized REST Endpoints

The easiest way to validate if the ODataConnector is running is to browse to following address in web browser:

<http://localhost/ODataConnector/rest/Endpoints>

NOTE: FireFox or Chrome will get the JSON response directly, Internet Explorer allows you to save or open the .json file.

REST Endpoints in the ODataConnector are available for:

- Real-time Data
- Historical Data
- Historical Alarms
- Facility A
- Quality AnalytiX
- Energy AnalytiX



GENESIS64 REST API Integration



APPLICATION NOTE

October 2015

NOTE: Complete endpoints description and syntax available in the help located at <http://localhost/ODataConnector/rest/Help/Html>

Below is an example of the REST API *RealTimeData* endpoint in Web Browser that will get the current value from the *svrsim:sine float -100 100* simulation signal.

1. Send the request for REST data by entering following URL into the browser:

<http://localhost/ODataConnector/rest/RealtimeData?PointName=svrsim:sine float -100 100>

2. The response from the ODataConnector should be similar to the figure below.

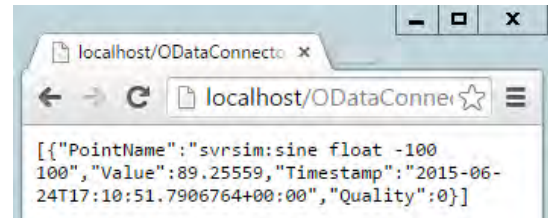


Figure 10 - Response from the ODataConnector