

# Developer Series: Enhance Niagara Applications with JSON Toolkit

July 8, 2021

## Q&A

1. I have a need to send a batch of JSON data to the National Grid [UK Power Distribution]. The data is based on a large set of Modbus meter energy values, but the data has to have some simple arithmetic performed on it before being published via MQTT.

Since all data has to be sent every second, does this mean that a BoundQuery, triggered every second, is the way to go? Since the National Grid places limits on the amount of data that can be sent in one publish point, how do I ensure that the BoundQuery "slices" the data into manageable sizes?

For a time-bound application using Modbus, it is first imperative to establish if you can poll the number of points needed within the 1 second interval mentioned. This may depend on the bus type used and any device limitations (e.g. inter message delay or low baud rate). You should test carefully to see if polling; arithmetic; JSON output and onward transmission via MQTT occur within the time window specified. Performance during a station save or JVM garbage collection event may also need to be considered.

Assuming the result of the arithmetic is found in the component space (rather than the result of BQL query aggregating historical data, for example) then it sounds like it may be best to use a Bound Property to include the result of the arithmetic, and then use "On Demand Only" (rather than "COV") with a Time Trigger (or multi-vibrator) linked to the Generate JSON action of the Schema.

If you did need to access histories for the arithmetic result, then when the schema is in On Demand mode, it will run the query inline as part of the Generate action and cache the result for the building of the JSON string, so this is workable so long as the query is reliably fast.

Limiting size could be achieved various ways - limiting the number or scope of the bound properties, for BQL queries the where, top, and depth keywords. It would even be possible to paginate the schema output in a program object or substring component as it's ultimately just a String - this may be preferable to writing your own query output style as demonstrated in the presentation as these aren't really designed to output multiple times for a single "generate" action.

## 2. Is it possible to get a copy of the station?

There is not a single station containing all the examples in this presentation. The query output style example code is already within the developer documentation for the module, and the jsonToolkit palette itself contains many useful examples for getting started with building JSON schemas.

If you have a specific request for one of the examples featured in this presentation, please request this via your support channel.

## 3. Can I turn the whole station into a generic schema and then use that HTTP access as a rudimentary REST API for all of Niagara?

You could create a JSON schema which targets points, histories or alarms within the station, and then either make use of the schemas exporter URL to access the JSON via the station Web Service, or link that Schema to an HTTP component which could push the message to an external endpoint.

oBIX is another option to consider for RESTful access to points, alarms and histories with minimal configuration on the station side. JSON Toolkit allows customization of the payload contents, but requires engineering to do so.

We advise caution if a schema targets an entire station, particularly when using a controller rather than a supervisor due to the controller's performance constraints. While the JSON schema queries and string generation occurs on background threads, it might adversely affect control functionality if your application requires very frequent publishing of data, or brings points into subscription that were not be used before the schema was in use.

For some cases the Relative JSON Schema performs much better as only points that change will be published - this approach is usually preferred when sending data to the cloud via MQTT for example.

## 4. Is this feature licensed? I think a slide showed something about licensed in a production environment.

JSON Toolkit is free to evaluate on engineering or demo hosts, production use requires a paid for license feature and active software maintenance (SMA). Using JSON Toolkit does not consume global capacity from the stations license.

## 5. What happens if the Software Maintenance Agreement (SMA) expires?

When the maintenance expires, JSON Schemas, Routers, Selectors and Handlers will report a fault status and cease to create, or process, JSON strings on the next reboot of that station.

The JsonSchemaService allows you to configure alarms to alert you when this expiration is approaching, and a warning appears on the user login page.



## 6. Is there a way to have histories send a group of data in one message?

You could use multiple JSON Schema Queries to combine several histories within a single JSON schema.

As the JSON Schema Queries support any query that returns a BITable, you might also consider using a Series Transform (e.g. transform:slot/AggregateTransform) which would allow use of a rollup for shared timestamps.

It's even possible to mix different sources in a single schema, such as live point values from the component space combined with the most recent alarm and history records relating to that point by using bindings and query results together.