

## GDP - Feature #108

### GDP read subscription service needed for Virtual Instrumentation

06/16/2017 11:32 AM - Rick Pratt

<b>Status:</b>	Closed	<b>Start date:</b>	06/16/2017
<b>Priority:</b>	High	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Description</b>			
<p>This requirement was originally appended by Christopher Brooks to Issue 95, "Working GDP RESTful Interface needed for Signpost":</p> <pre>One open issue is that we also need to be able to do subscription.</pre> <pre>We could do Long Polling (<a href="https://en.wikipedia.org/wiki/Push_technology#Long_polling">https://en.wikipedia.org/wiki/Push_technology#Long_polling</a>)</pre> <pre>Or, we could use a WebSockets interface. There is a WebSocketClient accessor (<a href="https://www.icyphy.org/accessors/library/index.html?accessor=net.WebSocketClient">https://www.icyphy.org/accessors/library/index.html?accessor=net.WebSocketClient</a>)</pre>			
<p>The above requirement is for a read subscription gateway or server deployment (whether RESTful or not) which might allow web browser application (Use Case: Virtual Instrumentation) software to operate without direct/local inclusion of GDP software.</p> <p>The RESTful option (Long Polling) is hereby shelved, as further investigation has shown that the available HTTP/HTML methods for Long Polling have at least two significant drawbacks:</p> <ul style="list-style-type: none"><li>• Clients on mobile/wireless infrastructure (e.g. laptops on Wifi) are slow -- O(minutes) -- to discover a session loss from intermittent connectivity.</li><li>• While Transfer-encoded chunked record delivery is possible, chunks can be buffered and coalesced by intermediate "transparent" firewalls/proxies, complicating support of Internet-accessible centralized deployment.</li></ul> <p>The non-RESTful WebSockets option is still on the table, as a potential technology on which to build and deploy a GDP read subscription service. The characteristics of a WebSocket upgrade imply a few operational constraints that limit the use cases:</p> <ul style="list-style-type: none"><li>• WebSocket clients are required to bitmask all client request frames, to protect against MiTM web proxy cache attacks, even if within an encrypted session, impacting power budgets on low power platforms.</li><li>• WebSocket upgrades are often blocked by domain firewalls and cloud providers as a security policy. A WebSocket-based GDP read subscription service (aka server deployment) on the Internet is likely to have mixed success driven by otherwise "transparent" network elements unique to a given user's network path.</li><li>• WebSockets provides a bi-directional communication upgrade from an HTTP/HTML session, and a generic message-based protocol is defined within a WebSocket, but a read subscription session protocol would need to be borrowed (or invented) on both clients and servers, somewhat retreating from the original goal to lighten the custom software burden on the client side.</li></ul>			
<b>Additional Notes:</b>			
<p>A WebSocket read subscription service should probably be network-local to the target userbase.</p> <p>A sensor gateway design pattern (Node.js/WebSockets/GDP Software on a BeagleBone) may be</p>			

one way to provide read subscription service in a way that is generally useful to all potential GDP users. A similar installation could be made on gdp-rest-01, with the understanding that it is primarily for internal UCB use and "best effort" for external clients, though support requests may arise regardless.

Nitesh has an active parallel investigation to determine whether this Issue can be satisfied by other means.

**Related issues:**

Related to GDP - Support #109: GDP with accessors (an umbrella item)

**New**

**History**

**#1 - 06/19/2017 12:11 PM - Nitesh Mor**

- Related to Support #109: GDP with accessors (an umbrella item) added

**#2 - 06/21/2017 05:21 PM - Eric Allman**

Another question I would ask is if WS is any faster to heal from failures than long poll. If there is a heartbeat (keepalive) on the WS then it's a potential power draw; if not it probably has to wait for the much longer timeout,  $O(\text{minutes})$  as with long poll. Maybe there's another way to get around it.

**#3 - 06/22/2017 06:06 AM - Anonymous**

Interesting question about healing of WebSockets. In the process of getting the GDPToWebSockets swarmlet deployed, I was restarting it every 15 seconds (in production, the restart time would be much greater). The test accessor that uses WebSockets to connect would get the first 15 seconds of data and then not get data after GDPToWebSockets restarted.

I believe that the client should define one or more reconnection policies.

One policy could be to reconnect if data is not received after a certain amount of time. This would work well for subscriptions to periodic log sources.

Another policy could be to reconnect after a certain amount of elapsed time.

Having a separate heartbeat signal is a possibility. I'm a little less concerned about power draw for subscription. Right now, the signpost is not subscribing to logs. The virtual instrument demo is subscribing, but it is assumed that the demo is running in the foreground.

One possibility is to have a swarmbox that has mains power subscribe to the log and do any necessary processing and then signal local low power nodes using any available protocol.

**#4 - 06/22/2017 10:23 AM - Nitesh Mor**

- Status changed from New to Feedback

I just pushed a Python based websocket subscriber gateway ([source:gdp-if|websocket-subscriber](https://github.com/gdp-if/websocket-subscriber)).

If the accessor based GDPToWebSockets swarmlet has to be restarted periodically, I will urge you to try out this Python based websocket instead. It is more stable and does not need to be restarted periodically (I hope). Further, it is easy to run this as a service; if there are platform compatibility issues, this websocket gateway can very well be distributed as a Docker image. The prerequisites and client interface is documented at [source:gdp-if|websocket-subscriber/README.md](https://github.com/gdp-if/websocket-subscriber/README.md).

Note that bug [#111](#), which causes extra communication between this gateway and gdplogd. However, this should not affect the websocket users in any way. I am marking this feature request as 'Feedback'. If this satisfies the demands of getting subscription data to limited environments (web-browser, for example), please mark it as closed.

**#5 - 06/22/2017 10:49 AM - Anonymous**

Excellent. I'm out of time, so if you want to install it on terra, send me your public key and I'll add it to sbuser's authorized\_keys2 file.

We are using ~sbuser/cg/node\_modules/.bin/pm2 to manage starting and restarting processes.

It would be helpful if the syntax between Edward's GDPToWebSocket.xml model and your model were compatible. I have no idea about this.

If they used the same syntax, then we could quickly see if the Python GDP/WebSocket interface has the same issue.

**#6 - 06/22/2017 11:04 AM - Nitesh Mor**

Christopher Brooks wrote:

Excellent. I'm out of time, so if you want to install it on terra, send me your public key and I'll add it to sbuser's authorized\_keys2 file.

We are using ~sbuser/cg/node\_modules/.bin/pm2 to manage starting and restarting processes.

Sending you the SSH key via email. I can give it a try, but I have no prior experience using pm2.

It would be helpful if the syntax between Edward's GDPToWebSocket.xml model and your model were compatible. I have no idea about this.

If they used the same syntax, then we could quickly see if the Python GDP/WebSocket interface has the same issue.

No, they don't use the same syntax. This Python gateway is significantly low-level in that it gives the users control over the number of records to fetch (as opposed to subscribe for eternity), as well as returns more meaningful errors/data to the user (including things such as timestamps, signatures, etc). I haven't looked at Edward's code quite yet (need to start up Ptolemy, etc), but it should be easy to make any syntax changes.

**#7 - 06/22/2017 12:51 PM - Nitesh Mor**

- Status changed from Feedback to In Progress

Marking it open again. I naively assumed JSON will deal well with binary data (ugh!!!). Need to put some fixes in to handle the most common cases. I will change the status once I have verified that it works.

**#8 - 06/22/2017 03:42 PM - Nitesh Mor**

- Status changed from In Progress to Resolved

**#9 - 06/22/2017 04:13 PM - Nitesh Mor**

I fixed the issues regarding the binary data; it's not a pretty fix but I hope it lets things get started. Logs with binary data as payload (as soon as folks start using encryption) will fail because JSON doesn't support binary data.

In order to test on your local machine, start the gateway by following [source:gdp-ij/websocket-subscriber/README.md](https://github.com/gdp-ij/websocket-subscriber/README.md). Open [source:gdp-ij/websocket-subscriber/test/test.html](https://github.com/gdp-ij/websocket-subscriber/test/test.html) in a web browser and specify the name of a log you'd like to subscribe to.

**#10 - 06/25/2018 11:48 AM - Eric Allman**

- *Status changed from Resolved to Closed*