



CONNECTING
THE WORLD



NIAGARA FORUM APAC | 9-10 SEPTEMBER 2025



MARTIN HUANG

APAC Tech Manager

Tridium

NF
25

RESTful API demo

– servlet and client + JSON



Servlet
Http Client

JSON & XML

01

Restful Introduction

02

JSON & XML

03

Servlet

04

Http Client Comm

A photograph of a modern building at dusk. The building features a large, dark, gabled roof and a central entrance with a wide staircase. The interior lights are on, and the entrance is brightly lit. The building is surrounded by lush tropical landscaping, including palm trees and other plants. The sky is a deep blue, and the overall atmosphere is serene and inviting.

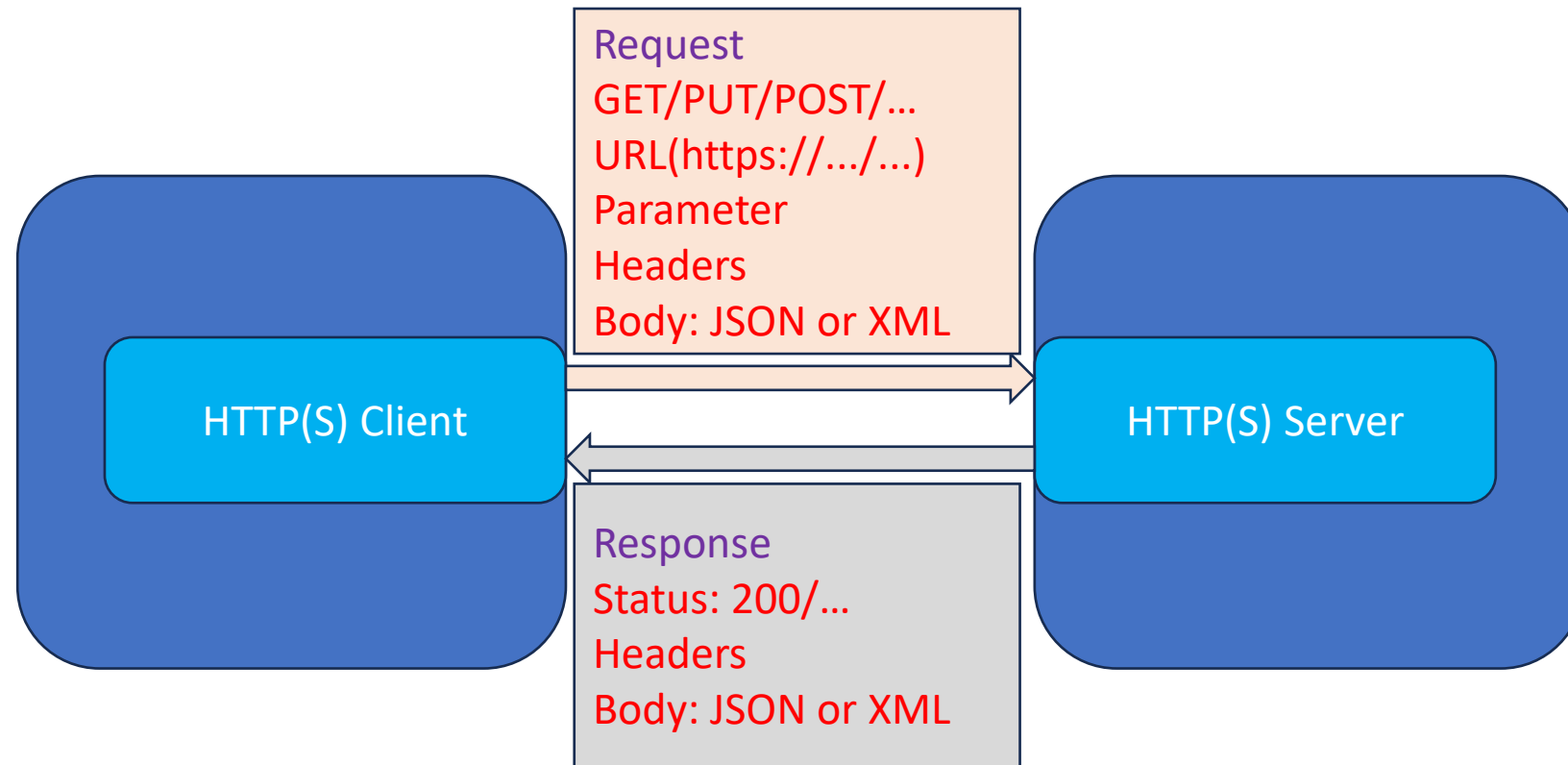
Restful Introduction

NIAGARA FORUM
CONNECTING
THE WORLD

RESTful Introduction



What is RESTful



RESTful features

Standardization
+ flexibility

Supported by
native languages
or 3rd party libs

Easy to
troubleshoot

Cross-platform
and
interoperability

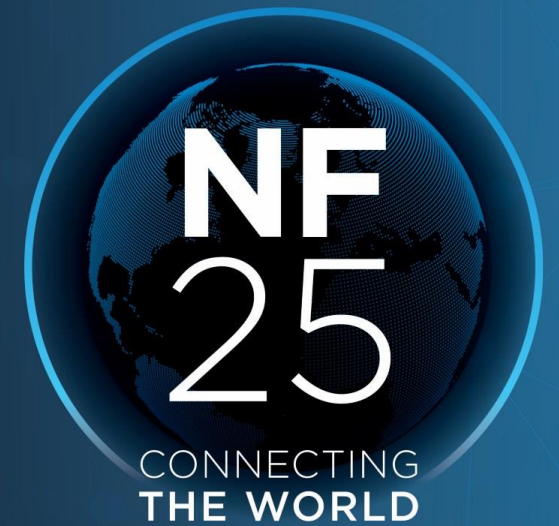
Secure
supported

Support cloud
and IoT

JSON & XML

JSON - JavaScript Object Notation

XML - Extensible Markup Language



JSON API in nre

- Key classes (Private API)
 - com.tridium.json.JSONObject

```
{  
  "roomTemp":21.3  
}
```

```
JSONObject jsonObject = new JSONObject();  
jsonObject.put("roomTemp", 21.3);
```

```
String jsonString = "{\"roomTemp\":21.3}";  
JSONObject jsonObject = new JSONObject(jsonString);  
double roomTemp = jsonObject.optDouble("roomTemp");
```

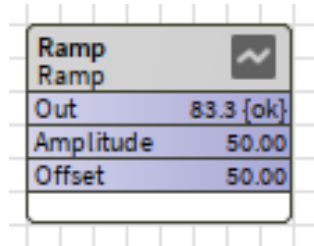
- com.tridium.json.JSONArray

```
[  
  "Item1",  
  "Item2",  
  123  
]
```

```
JSONArray jsonArray = new JSONArray();  
jsonArray.put("Item1").put("Item2").put(123);
```

```
String arrString = "[\"Item1\",\"Item2\",123]";  
JSONArray jsonArray = new JSONArray(arrString);  
String s1 = jsonArray.optString(0);  
String s2 = jsonArray.optString(1);  
int val = jsonArray.optInt(2);
```

JSONObject Encoding Example



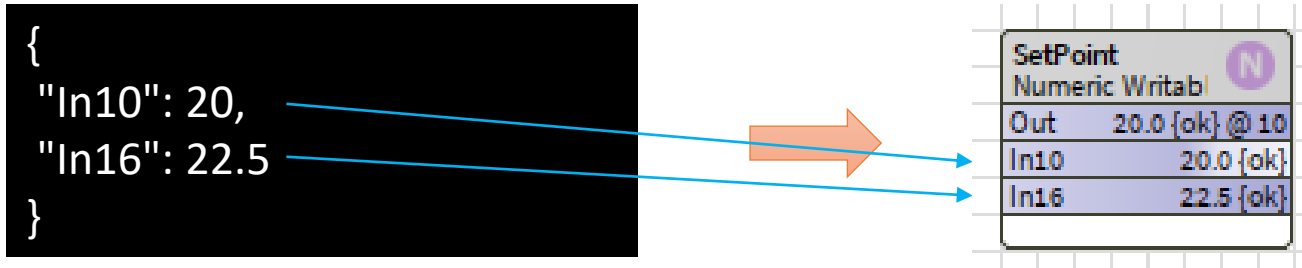
Ramp	
Out	83.3 {ok}
Amplitude	50.00
Offset	50.00



```
{  
  "name": "Ramp",  
  "Out": 83.3  
}
```

```
JSONObject jsonEncode(BNumericPoint ramp, Context cx)  
{  
    JSONObject jsonObject = new JSONObject();  
    jsonObject.put("name", ramp.getDisplayName(cx))  
        .put("Out", ramp.getOut().getValue());  
    return jsonObject;  
}
```

JSONObject Decoding Example



```
void jsonDecode(BNumericWritable setPoint, JSONObject jsonObject)  
{  
    setPoint.setIn10(new BStatusNumeric(jsonObject.optDouble("In10")));  
    setPoint.setIn16(new BStatusNumeric(jsonObject.optDouble("In16")));  
}
```

JSONArray Example

```
// encode example: Java String[]{"aaa","bbb"} -> JSON ["aaa","bbb"]
JSONArray JSONEncodeExample(String[] strArr) {
    JSONArray jsonArray = new JSONArray();
    for (String s : strArr) {
        jsonArray.put(s);
    }
    return jsonArray;
}

// decode example: JSON[1,2,3] -> Java int[]
int[] JSONDecodeExample(JSONArray jsonArray) {
    int[] values = new int[javascriptArray.length()];
    for (int i = 0; i < jsonArray.length(); i++) {
        values[i] = jsonArray.optInt(i);
    }
    return values;
}
```

3rd party JSON libs

- 3rd party libs
 - Jackson
 - <https://github.com/FasterXML/jackson>
 - Gson
 - <https://github.com/google/gson>
 - org.json (JSON-java)
 - <https://github.com/stleary/JSON-java>

Gson example

```
void gsonEncodeExample() {  
    JsonObject jsonObject= new JsonObject(); // Create a new JsonObject  
    jsonObject.addProperty("name","Martin"); // Add property to JsonObject  
    System.out.println(jsonObject);          // Print the JsonObject  
}  
  
String gsonDecodeExample() {  
    String jsonString = "{\"name\":\"Martin\"}"; // JSON string  
    // Parse the JSON string into a JsonObject  
    JsonObject jsonObject = JsonParser.parseString(jsonString).getAsJsonObject();  
    // Access data from the JsonObject  
    return jsonObject.get("name").getString();  
}
```

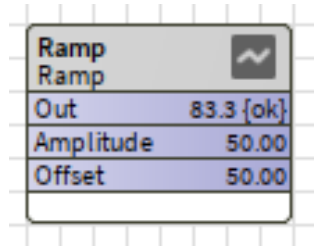
XML API in nre

- javax.baja.xml.XElem

```
<req>  
  <item ord="h:abc"/>  
  <item ord="slot:/data/Ramp"/>  
</req>
```

```
<res>  
  <item ord="h:abc" val="10.2" />  
  <item ord="slot:/data/Ramp" err="Point is Not found" />  
</res>
```

XML Encoding Example



Ramp Ramp	
Out	83.3 {ok}
Amplitude	50.00
Offset	50.00



<Ramp Out="83.3"/>

```
XElem xmlEncode(BNumericPoint ramp, Context cx)
{
    XElem xml = new XElem(ramp.getName());
    xml.addAttr("Out", String.valueOf(ramp.getOut().getValue()));
    return xml;
}
```

JSONObject Decoding Example

<Ramp In10="20" In16="22.5"/>



SetPoint	
Numeric Writable	N
Out	20.0 {ok} @ 10
In10	20.0 {ok}
In16	22.5 {ok}

```
void xmlDecode(BNumericWritable setPoint, XElement xml)
{
    setPoint.setIn10(new BStatusNumeric(xml.getd("In10")));
    setPoint.setIn16(new BStatusNumeric(xml.getd("In16")));
}
```

Servlet

Jetty Servlet

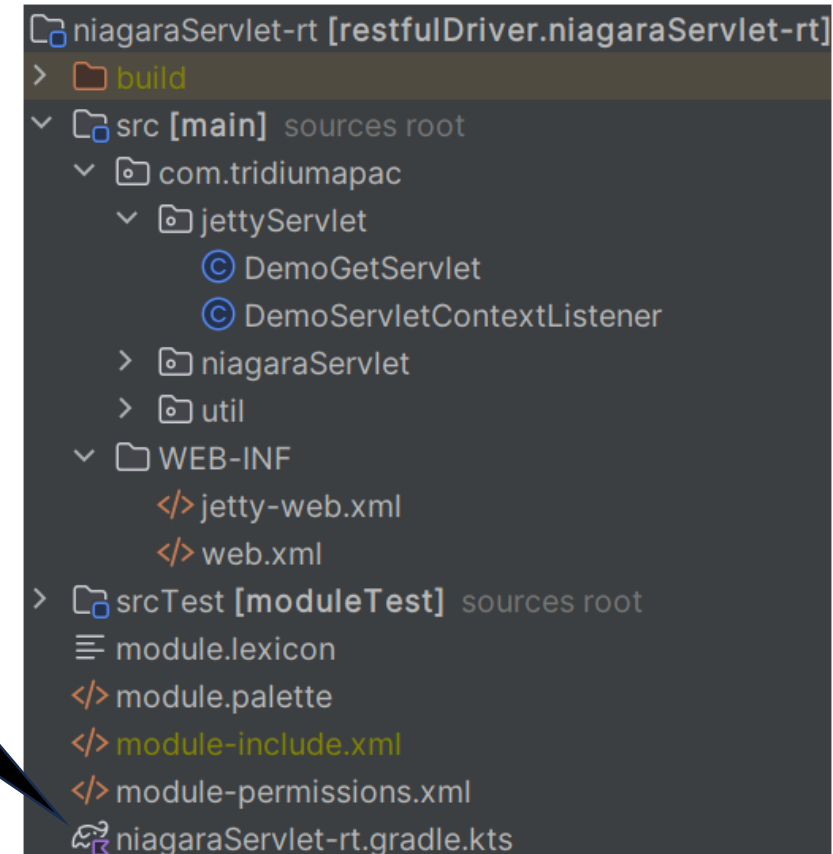
Component Servlet



Servlet – Building Environment

```
dependencies {  
    // NRE dependencies(Including JSON/XML lib)  
    nre(":nre")  
    // Niagara module dependencies  
    api(":baja")  
    api(":control-rt")  
    api(":web-rt")  
    // servlet lib  
    implementation(":javax.servlet-api-3.1.0")  
}
```

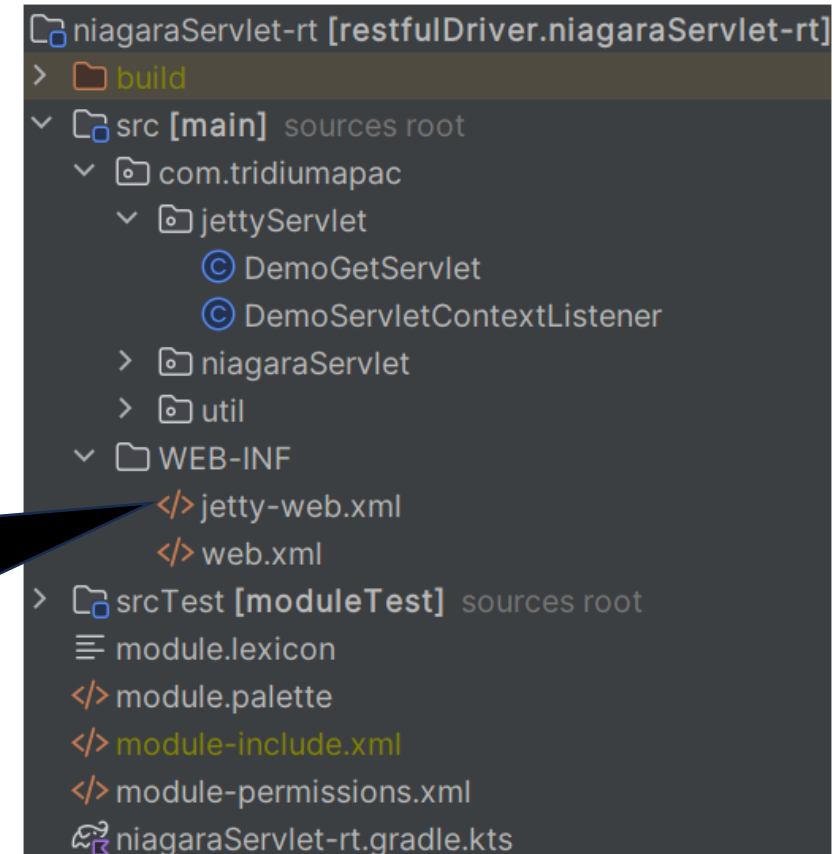
C:\Niagara\Niagara-4.15.0.130\bin\ext\javax.servlet-api-3.1.0.jar



Jetty Servlet

- Identify Base URL
 - e.g. `https://127.0.0.1/api/v1`

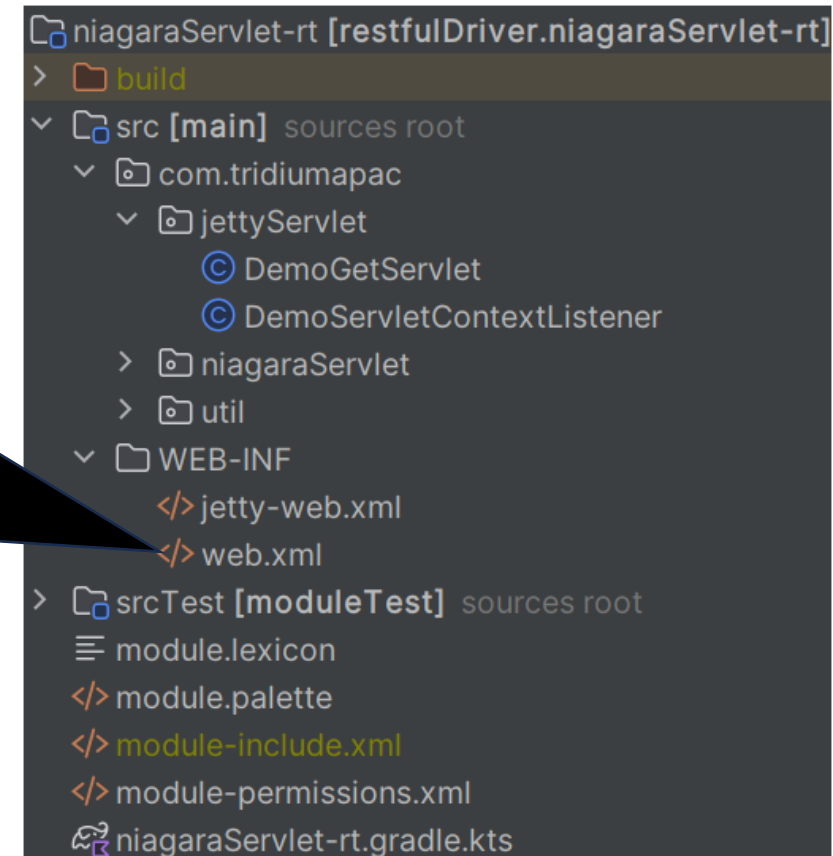
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN"
"http://www.eclipse.org/jetty/configure.dtd">
<Configure id="webApp" class="org.eclipse.jetty.webapp.WebAppContext">
  <!-- Change the Context Path from the module name to something else -->
  <Set name="contextPath">/api/v1</Set>
</Configure>
```



Jetty Servlet

- Map servlet
 - e.g. <https://127.0.0.1/api/v1/query>

```
<servlet>
  <servlet-name>DemoGetServlet</servlet-name>
  <servlet-class>com.tridiumapac.jettyServlet.DemoGetServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>DemoGetServlet</servlet-name>
  <url-pattern>/query</url-pattern>
</servlet-mapping>
```



```
niagaraServlet-rt [restfulDriver.niagaraServlet-rt]
> build
v src [main] sources root
  v com.tridiumapac
    v jettyServlet
      DemoGetServlet
      DemoServletContextListener
    > niagaraServlet
    > util
  v WEB-INF
    </> jetty-web.xml
    </> web.xml
> srcTest [moduleTest] sources root
  module.lexicon
  </> module.palette
  </> module-include.xml
  </> module-permissions.xml
  niagaraServlet-rt.gradle.kts
```

Jetty Servlet

- Demo

```
public class DemoGetServlet extends HttpServlet
{
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException
    {
        String pointOrd = req.getParameter("point");
        BControlPoint point = (BControlPoint)BOrd.make(pointOrd).get(Sys.getStation());
        JSONObject jRes = new JSONObject();
        jRes.put("value", point.getOutStatusValue().toString());
        resp.getWriter().write(jRes.toString());
    }

    private static final long serialVersionUID = 4341464327501795277L;
}
```

```
niagaraServlet-rt [restfulDriver.niagaraServlet-rt]
> build
  src [main] sources root
    com.tridiumapac
      jettyServlet
        DemoGetServlet
        DemoServletContextListener
      niagaraServlet
      util
    WEB-INF
      </> jetty-web.xml
      </> web.xml
  srcTest [moduleTest] sources root
    module.lexicon
    </> module.palette
    </> module-include.xml
    </> module-permissions.xml
    niagaraServlet-rt.gradle.kts
```

Servlet – Component Servlet

```
@NiagaraType
@NiagaraProperty(name = "servletName", type = "String", defaultValue = "query", override = true)
public class BBqlQueryServlet
    extends BWebServlet
{
    @Override
    public void doGet(WebOp op) throws Exception
    {
        HttpServletRequest request = op.getRequest();
        String sBql = request.getParameter("bql");

        op.getResponse().setContentType("application/json;charset=UTF-8");
        HtmlWriter writer = op.getHtmlWriter();
        JSONObject jRes = processQuery(BOrd.make(sBql)); // process bql query

        writer.write(jRes.toString());
        writer.flush();
    }
}
```

Context for Permission Checking

- Component Servlet
 - WebOp itself is a Context which including BUser
- Jetty Servlet
 - Get from attribute “niagara.context”

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException
{
    BasicContext cx = (BasicContext)req.getAttribute("niagara.context");
    String pointOrd = req.getParameter("point");
    BControlPoint point = (BControlPoint)BOrd.make(pointOrd).get(Sys.getStation(), cx);
    //.....
}
```

Tested using Postman

The screenshot shows the Postman interface for a GET request. The URL bar contains `https://127.0.0.1/query?bql=slot:/Data|bql:select slotPath,out.value from control:ControlPoint`. The 'Query Params' section shows a table with one parameter: `bql` with the value `slot:/Data|bql:select slotPath,out.value from control:ControlPoint`. The 'Body' tab is selected, showing a JSON response in 'Pretty' format. The response status is 200 OK, with a time of 8 ms and a size of 695 B.

HTTP test / 127.0.0.1 / n4Query

GET `https://127.0.0.1/query?bql=slot:/Data|bql:select slotPath,out.value from control:ControlPoint` Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
bql	slot:/Data bql:select slotPath,out.value from control:ControlPoint	

Body Cookies (1) Headers (8) Test Results

Status: 200 OK Time: 8 ms Size: 695 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": [
3     {
4       "out.value": 96.04666666666665,
5       "slotPath": "slot:/Data/Ramp"
6     },
7     {
8       "out.value": 0,
9       "slotPath": "slot:/Data/NumericWritable"
10    }
11  ]
12 }
```

Tested using curl

```
hwg@CH5OLTF2FL814:~$ curl -k -u api:XXXXXXXXXXXX15 https://172.26.32.1/query?bql=bql:select%20slotPath,out.value%20from%20control:ControlPoint  
{"result":[{"out.value":2.7066666666666666,"slotPath":"slot:/Data/Ramp"}, {"out.value":0,"slotPath":"slot:/Data/NumericWritable"}]}hwg@CH5OLTF2FL814:~$
```

HTTP Client Comm

- *HTTP Client in Ndriver*
- *OKHttp*



HttpComm API Process

```
// (1) Create NHttpRequest, including URL, header, body.  
NHttpRequest request = createRequest();  
  
// (2) Send Request and wait for response  
NHttpResponse httpResponse = getRestfulDriverNetwork().hcomm().sendRequest(request);  
  
// (3) Parse response JSON object  
String sRes = new String(httpResponse.getData());  
JSONObject jRes = new JSONObject(sRes);
```

Create NHttpRequest Example

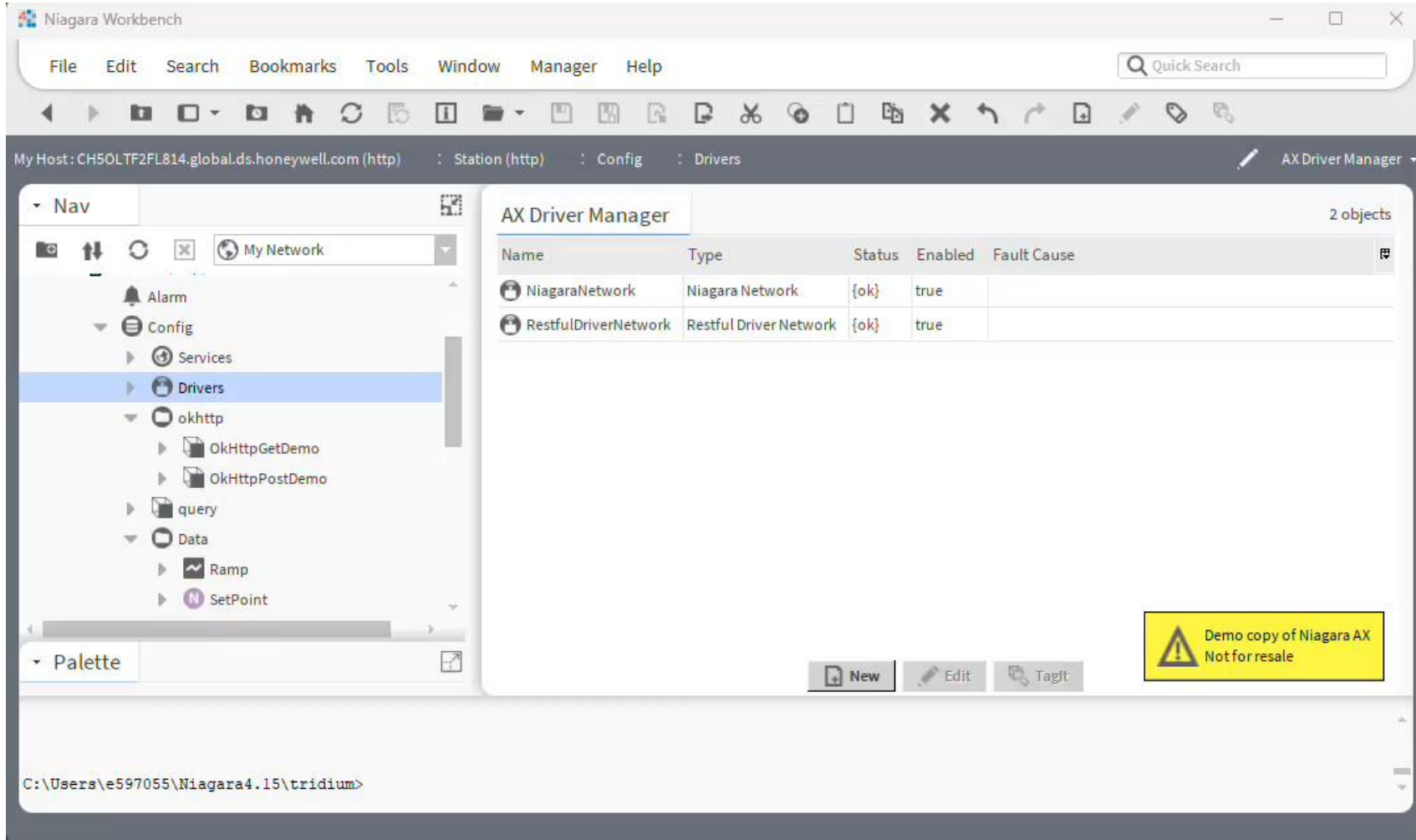
```
private NHttpRequest createRequest(BControlPoint[] controlPoints) throws Exception {  
    URLParser parser = new URLParser(getUrl());  
    getRestfulDriverNetwork().getHttpConfig().setUseTls(parser.isSecure());  
    BIpAddress ip = parser.getServer();  
    JSONArray jaReq = new JSONArray();  
    for (BControlPoint point : controlPoints) {  
        BRestfulDriverProxyExt pe = (BRestfulDriverProxyExt)point.getProxyExt();  
        jaReq.put(pe.getOrd());  
    }  
    NHttpRequest request = new NHttpRequest(ip, "POST", "/api/v1/read");  
    String pwd = AccessController.doPrivileged((PrivilegedAction<String>)getPassword()::getValue);  
    request.addBasicAuthorization(getUserName(), pwd);  
    request.setData(jaReq.toString());  
    request.setRetryCount(1);  
    request.setResponseTimeOut(30000);  
    return request;  
}
```

(1) URL

(2) Header

(3) Body

Http Client Driver Demo



The screenshot shows the Niagara Workbench interface. The left sidebar has a navigation tree with 'Drivers' selected. The main pane displays the 'AX Driver Manager' with a table of two objects:

Name	Type	Status	Enabled	Fault Cause
NiagaraNetwork	Niagara Network	{ok}	true	
RestfulDriverNetwork	Restful Driver Network	{ok}	true	

At the bottom right, a yellow warning box states: 'Demo copy of Niagara AX Not for resale'. The bottom status bar shows the file path: 'C:\Users\e597055\Niagara4.15\tridium>'.

- Discover:
 - GET
 - /api/v1/query
- Read:
 - POST
 - /api/v1/read
- Write:
 - PUT
 - /api/v1/write

OkHttpClient Example

```
OkHttpClient.Builder clientBuilder = new OkHttpClient.Builder();
// bind Niagara Certificate Store
clientBuilder.sslSocketFactory(SslUtil.getSocketFactory(), SslUtil.getTrustManager())
    .hostnameVerifier(new NoOpHostnameVerifier());

OkHttpClient client = AccessController.doPrivileged((PrivilegedAction<OkHttpClient>()) -> {
    return clientBuilder.build();
});

MediaType JSON = MediaType.get("application/json; charset=utf-8");
RequestBody body = RequestBody.create(getBody(), JSON);
Request request = new Request.Builder()
    .url(getUrl())
    .post(body)
    .header("Authorization", getBasicCredential())
    .build();

String sRes = sendRequest(client, request);
JSONObject jRes = new JSONObject(sRes);
```

THANK YOU

