

Using GELLO online with a RESTful web service

Overview

The Medical Objects RESTful web service is available. A POST is in json format and needs to include three things:

1. patient data in json (or cjson) format
2. Package data if needing an extension or separate package to the default HL7_v2_VMR
3. the gello code itself.

Information is available [here](#).

This service is a research and development site currently. We welcome feedback from users to the info@medical-objects... address. If you wish to use it for commercial/mainstream purposes, please contact us.

Use of SNOMED-CT

Use of SNOMED CT through this web service requires a license from snomed.org . Do not use this functionality if you are not licensed to do so.

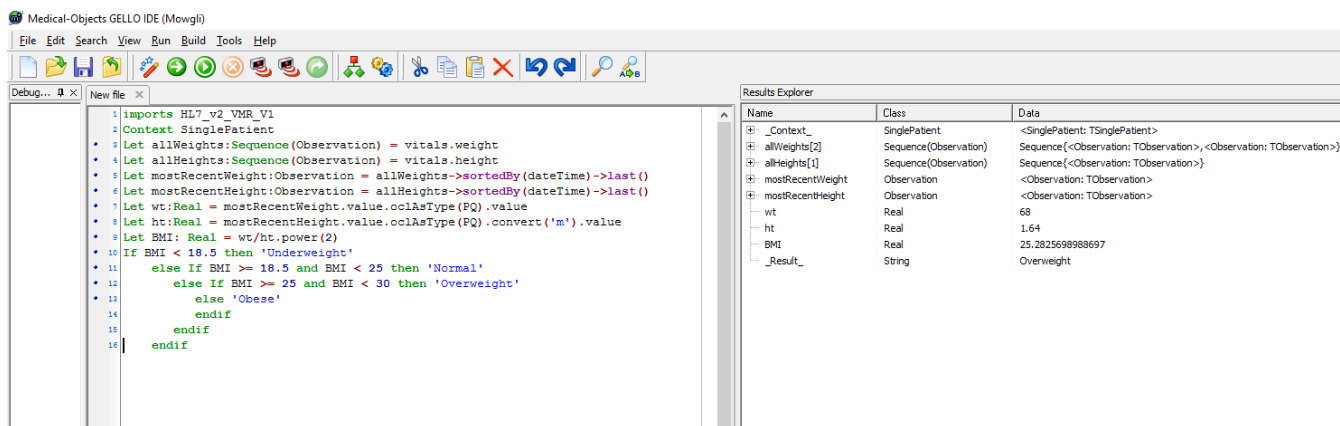
Worked example – Body Mass Index (BMI)

This worked example gets the latest height and weight for a patient and calculates the BMI on the MO web server, and then retrieves the result as json. The example makes use of the 'Postman – REST Client' chrome app.

Get it going in the editor first. Here is some GELLO code to calculate BMI:

```
imports HL7_v2_VMR_V1
Context SinglePatient
Let allWeights:Sequence(Observation) = vitals.weight
Let allHeights:Sequence(Observation) = vitals.height
Let mostRecentWeight:Observation = allWeights->sortedBy(dateTime)->last()
Let mostRecentHeight:Observation = allHeights->sortedBy(dateTime)->last()
Let wt:Real = mostRecentWeight.value.oclAsType(PQ).value
Let ht:Real = mostRecentHeight.value.oclAsType(PQ).convert('m').value
Let BMI: Real = wt/ht.power(2)
If BMI < 18.5 then 'Underweight'
else If BMI >= 18.5 and BMI < 25 then 'Normal'
else If BMI >= 25 and BMI < 30 then 'Overweight'
else 'Obese'
endif
endif
endif
```

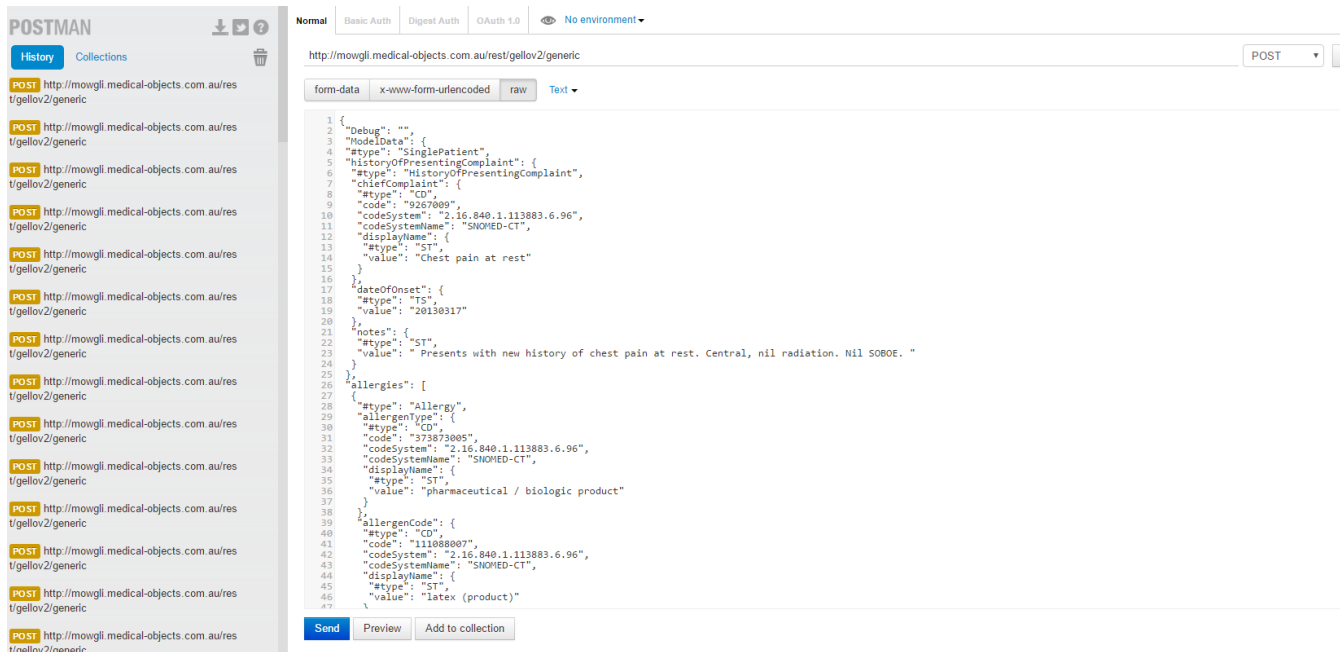
Using *fifthTest.xml*, here is the result:



The url string for the POST in the REST client is: <http://mowgli.medical-objects.com.au/rest/gellov2/generic>

The json code for a POST is [here](#) and it is named POST.txt. Instance data for the data model starts on line 3. The Gello query itself is on 3604.

Here is a screen shot of Postman showing some of the POST json text:



and the result itself:

Send

Preview

Add to collection

Body

Cookies (5)

Headers (7)


STATUS 200 OK


TIME 852 ms

Pretty

Raw

Preview





JSON

XML

```
1 {
2   "Results": [
3     {
4       "requestID": "1",
5       "wasSuccessful": true,
6       "result": "Overweight"
7     }
8   ]
9 }
```

In the POST text note how the GELLO section is now written with new line characters at the start of each line.

If we want to see the variables (say for debugging - although you would do this more in the editor), add 'V,1' to line 2, to make it:

```
"Debug": "V,1",
```

You can also return the result as a tuple instead of a single typed value (which in this case is simply a string).