

QGIS Application - Bug report #10890

Unable to add non-cached WFS layer through Python API

2014-07-15 12:53 AM - Neil Benny

Status:	Closed		
Priority:	Severe/Regression		
Assignee:			
Category:	Python plugins		
Affected QGIS version:	2.4.0	Regression?:	No
Operating System:	Windows	Easy fix?:	No
Pull Request or Patch supplied:	No	Resolution:	worksforme
Crashes QGIS or corrupts data:	No	Copied to github as #:	19257
Description			
<p>Adding a WFS layer (with iface.addVectorLayer) works without any problems when adding a cached WFS to the map canvas. However, using the BBOX parameter in the vector layer uri will add the layer to the TOC but no features are shown.</p> <p>I am using the BBOX parameter in the uri:</p> <pre>c_extent = self.canvas.extent() str_c_extent = c_extent.toString() str_c_extent = str_c_extent.replace(" : ", ",") url = site_url + "wfs.ashx?id=" + token + "&SRSNAME=EPSG:27700&SERVICE=WFS&VERSION=1.0.0&REQUEST=GetFeature&TYPENAME=" + layer + "&BBOX=" + str_c_extent iface.addVectorLayer(url, title, 'WFS')</pre> <p>printing url gives:</p> <p>http://myurl.co.uk/data_services/wfs.ashx?id=lalalala~&SRSNAME=EPSG:27700&SERVICE=WFS&VERSION=1.0.0&REQUEST=GetFeature&TYPENAME=The_Layer_I_Want&BBOX=279661.8626045954297297,692934.9235554961487651,280059.7859282681602053,693175.8901554960757494</p> <p>The code I am using was working OK in 2.2.</p>			
Related issues:			
Related to QGIS Application - Bug report # 10413: non-cached WFS won't load		Closed	2014-06-02

History

- #1 - 2014-07-17 05:31 AM - Giovanni Manghi
- Priority changed from High to Severe/Regression
 - Target version deleted (Future Release - High Priority)

#2 - 2014-09-22 04:49 AM - Neil Benny

I know we're a few weeks away from the lock, but I was wondering if anyone has been able to do anything with this, or could suggest a workaround?

- #3 - 2014-10-19 07:13 AM - Jürgen Fischer
- Status changed from Open to Feedback

Works fine for me. E.g.

```
iface.addVectorLayer("http://osgeolive7:8082/geoserver/ows?SERVICE=WFS&VERSION=1.0.0&REQUEST=GetFeature&TYPENAME=sf:roads  
&SRSNAME=EPSG:26713&BBOX=455273.50764834566507488,5780335.53749999962747097,520717.52360165433492512,5817545.962500000372529  
0000037252903","test","WFS")
```

Any messages in the log?

#4 - 2014-10-20 03:00 AM - Neil Benny

Hi Jurgen,

I am able to add WFS layers to the map canvas with `iface.addVectorLayer` (with the BBOX parameter in the URI) without any errors, as you have demonstrated. If the added layer has less features than the 'Geoserver maximum number of features' then all of the features are displayed without issue.

However, if the layer has more features than the maximum allowed for a single request, then the behaviour I would expect (and was experiencing in 2.2) is that on each pan/zoom/canvas refresh QGIS would request features up to the maximum, for the new extent. What I am instead experiencing is that no features are requested and a pan/zoom does not cause features to be requested again.

Adding the same layer in using the QGIS gui (and unchecking 'cache feature') results in the expected behaviour of features been requested on a pan/zoom.

#5 - 2014-10-20 05:23 AM - Jürgen Fischer

Neil Benny wrote:

However, if the layer has more features than the maximum allowed for a single request, then the behaviour I would expect (and was experiencing in 2.2) is that on each pan/zoom/canvas refresh QGIS would request features up to the maximum, for the new extent. What I am instead experiencing is that no features are requested and a pan/zoom does not cause features to be requested again.

Hm, not sure what the "maximum allowed for a single request" is, but

```
iface.addVectorLayer("http://192.168.0.156:8082/geoserver/ows?SERVICE=WFS&VERSION=1.0.0&REQUEST=GetFeature&TYPENAME=tiger  
:tiger_roads&SRSNAME=EPSG:4326&BBOX=-73.99393627054513445,40.71424130427189425,-73.99188234547472121,40.7148503992927715","test",  
715","test","WFS")
```

works for me too as it invokes new requests when panning.

#6 - 2014-10-20 08:33 AM - Neil Benny

By "maximum allowed for a single request" I mean the maximum number of features that Geoserver will return in a single wfs request, this is a setting which can be configured in Geoserver for a workspace.

For example, if a wfs layer contains 1,000 features and the max feature limit is set to 10,000 then the layer will load without issue and make new requests on each extent change.

If a wfs layer contains 20,000 features and the max feature limit is set to 10,000 then an extent change does not invoke a new request via python, but works fine via qgis gui.

Using debug view I have found a difference in the way QGIS handles layers added with Python Vs QGIS GUI.

Using QGIS GUI:

```
00000165 10.67942429 [9460] c:\src\qgis\src\gui\qgsmapcanvas.cpp(397) : (QgsMapCanvas::setLayerSet) Layers changed to: layer_1, layer_2
00000166 10.67949295 [9460] c:\src\qgis\src\gui\qgsmapcanvas.cpp(431) : (QgsMapCanvas::setLayerSet) Layers have changed, refreshing
00000167 10.67953396 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(984) : (QgsMapRenderer::setLayerSet) Entering: layer_1, layer_2
00000168 10.67958164 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(909) : (QgsMapRenderer::updateFullExtent) called.
00000169 10.67961884 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(919) : (QgsMapRenderer::updateFullExtent) Layer count: 2
00000170 10.67965317 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(929) : (QgsMapRenderer::updateFullExtent) Updating extent using
workspace:layer_1
00000171 10.67970562 [9460] c:\src\qgis\src\core\qgsvectorlayer.cpp(830) : (QgsVectorLayer::extent) Extent of layer:
227656.2723751600424293,674379.0292924284003675 : 289409.2569257552386262,738982.7003754294710234
00000172 10.67975140 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(930) : (QgsMapRenderer::updateFullExtent) Input extent:
227656.2723751600424293,674379.0292924284003675 : 289409.2569257552386262,738982.7003754294710234
00000173 10.67981434 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(784) : (QgsMapRenderer::layerExtentToOutputExtent) proj extent =
227656.2723751600424293,674379.0292924284003675 : 289409.2569257552386262,738982.7003754294710234
00000174 10.67986488 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(942) : (QgsMapRenderer::updateFullExtent) Output extent:
227656.2723751600424293,674379.0292924284003675 : 289409.2569257552386262,738982.7003754294710234
```

Using iface.addvectorlayer:

```
00000054 1.51108801 [9460] c:\src\qgis\src\gui\qgsmapcanvas.cpp(397) : (QgsMapCanvas::setLayerSet) Layers changed to: layer_1, layer_2
00000055 1.51113713 [9460] c:\src\qgis\src\gui\qgsmapcanvas.cpp(431) : (QgsMapCanvas::setLayerSet) Layers have changed, refreshing
00000056 1.51117253 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(984) : (QgsMapRenderer::setLayerSet) Entering: layer_1, layer_2
00000057 1.51120412 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(909) : (QgsMapRenderer::updateFullExtent) called.
00000058 1.51123714 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(919) : (QgsMapRenderer::updateFullExtent) Layer count: 2
00000059 1.51126993 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(929) : (QgsMapRenderer::updateFullExtent) Updating extent using layer_1
00000060 1.51130259 [9460] c:\src\qgis\src\core\qgsvectorlayer.cpp(830) : (QgsVectorLayer::extent) Extent of layer: Empty
00000061 1.51133454 [9460] c:\src\qgis\src\core\qgsmaprenderer.cpp(930) : (QgsMapRenderer::updateFullExtent) Input extent: Empty
```

I don't know to much about QGIS/C++ but it seems the new extent is not being picked up maybe?

If it would help to demonstrate the issue I can setup a WFS layer for you to test?

#7 - 2014-10-20 08:44 AM - Jürgen Fischer

Neil Benny wrote:

I don't know to much about QGIS/C++ but it seems the new extent is not being picked up maybe?

Both layers have the same data source uri (in layer properties or in the layer tooltip in the layer tree)?

If it would help to demonstrate the issue I can setup a WFS layer for you to test?

I suppose that would at least make it clearly reproducible.

#8 - 2015-02-15 05:27 AM - Jürgen Fischer

- *Resolution set to worksforme*
- *Status changed from Feedback to Closed*

closing for the lack of feedback.