

QGIS Application - Bug report #18781

QgsProject.instance().geometry().asPolyline() not working on QGIS 3

2018-04-20 07:39 PM - Roger Gonzalez

Status:	Closed	
Priority:	Normal	
Assignee:		
Category:	Geometry	
Affected QGIS version:	3.0.1	Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: invalid
Crashes QGIS or corrupts data:	No	Copied to github as #: 26668
Description		
The next code only works on 2.xx versions of QGIS, returning empty list when it should export Polyline coordinates like in QGIS 2.xx <pre>def getAttributeIndex(aLayer, attrName): if len(attrName) > 10 and aLayer.storageType() == 'ESRI Shapefile': iface.messageBar().pushMessage('Error', 'For ESRI Shapefiles, the maximum length of any attribute name is 10. Please choose a shorter attribute name.', level=QgsMessageBar.CRITICAL) return -3 AttrIdx = aLayer.dataProvider().fieldNameIndex(attrName) if AttrIdx == -1: caps = aLayer.dataProvider().capabilities() if caps & QgsVectorDataProvider.AddAttributes: res = aLayer.dataProvider().addAttributes([QgsField(attrName, QVariant.String)]) AttrIdx = aLayer.dataProvider().fieldNameIndex(attrName) aLayer.updateFields() if AttrIdx == -1: iface.messageBar().pushMessage('Error', 'Failed to create attribute!', level=QgsMessageBar.CRITICAL) return -1 else: iface.messageBar().pushMessage('Error', 'Failed to add attribute!', level=QgsMessageBar.CRITICAL) return -1 return AttrIdx layer_name= 'L_MT_IND' #layerMT1=QgsMapLayerRegistry.instance().mapLayersByName(layer_name)[0] ## for QGIS 2 layerMT1=QgsProject.instance().mapLayersByName(layer_name)[0] ## for QGIS 3 idx = getAttributeIndex(layerMT1, "DSSName") lineasMT = layerMT1.getFeatures() for lineaMT in lineasMT: line = lineaMT.geometry().asPolyline() # Read line geometry print(line)</pre>		

History

#1 - 2018-04-21 12:39 AM - Nyal Dawson

- Status changed from Open to Closed

- Resolution set to invalid

That's 2.x API - you need to change your code to use the 3 API

#2 - 2018-04-23 03:28 PM - Roger Gonzalez

Nyall Dawson wrote:

| *That's 2.x API - you need to change your code to use the 3 API*

That code uses 3 API...

#3 - 2018-04-23 03:34 PM - Jürgen Fischer

- *Description updated*

#4 - 2018-04-24 09:41 PM - Roger Gonzalez

- *Status changed from Closed to Reopened*

#5 - 2018-05-01 07:05 AM - Nyall Dawson

- *Status changed from Reopened to Feedback*

Can you share your data?

#6 - 2018-05-02 03:23 PM - Roger Gonzalez

- *File Test.zip added*

- *File Error.png added*

Nyall Dawson wrote:

| *Can you share your data?*

Sure, here is it

And this is the necessary code

```
layer_name= 'Test'
#layerMT1=QgsMapLayerRegistry.instance().mapLayersByName(layer_name)[0] ## for QGIS 2
layerMT1=QgsProject.instance().mapLayersByName(layer_name)[0] ## for QGIS 3

lineasMT = layerMT1.getFeatures()
for lineaMT in lineasMT:
    line = lineaMT.geometry().asPolyline()
    # Read line geometry
    print(line)
```

Ah - these are multipolyline geometries in 3.0, not polylines.

So your code needs to be:

```
for lineaMT in lineasMT:
    line = lineaMT.geometry().asMultiPolyline()[0]
# Read line geometry
print(line)
```

QGIS 3.0 always pulls in shapefiles using multi-geometry types for line or polygon layers. This change was made due to shapefiles not having strict constraints on the single/multi type feature geometries for lines and polygons -- so a layer reporting that it contained LineString geometries may have MultiLineString geometries mixed in there too. This breaks many assumptions in QGIS code, so for consistency it was decided to always return multi-geometries for lines/polygons for shapefiles.

Files			
Test.zip	5.72 KB	2018-05-02	Roger Gonzalez
Error.png	343 KB	2018-05-02	Roger Gonzalez