

# QGIS Application - Feature request #10764

## Python API to handle layer groups without iface

2014-07-01 02:53 AM - David AMAR

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Category:</b>	
<b>Pull Request or Patch supplied:</b>	<b>Resolution:</b> fixed/implemented
<b>Easy fix?:</b> No	<b>Copied to github as #:</b> 19147
<b>Description</b>	
<p>My intent is to automate production of raster files to use QgisMapServer as a WMS Server. In order to manage it, I use a stand-alone python script (without GUI) to handle QGS project files. See attached file modifyqgs.py.</p> <p>Up to now, I am able to add raster layers to an existing project using QgsMapLayerRegistry.</p> <p>Now, I want to organize layers using groups but I didn't find any API to acheive it using "headless" API - that is without <code>qgis.utils.iface.legendInterface()</code> component. Still undocumented <code>QgsLayerTreeGroup</code> used by <code>QgsProject</code> may be a future solution?</p> <p>Regards.</p>	

### History

#### #1 - 2014-07-01 07:04 PM - Martin Dobias

- Resolution set to fixed/implemented
- Status changed from Open to Closed

The `QgsLayerTree*` classes are exactly what you are looking for. Please see the doxygen documentation of the classes (especially `QgsLayerTreeNode`, `QgsLayerTreeLayer`, `QgsLayerTreeGroup`). I will add some more documentation to PyQGIS cookbook later. For now, your entry point is `QgsProject.instance().layerTreeRoot()` - from there you can start adding child groups and layers.

#### #2 - 2014-07-02 02:00 AM - David AMAR

Thank you Martin.

I have just tried these classes and I think there are some points to fix if you confirm them:

- `QgsLayerTreeNode`, `QgsLayerTreeLayer`, `QgsLayerTreeGroup` documentation is not generated by doxygen in official documentation

<http://qgis.org/api/2.4>.

- When I try `'print QgsProject.instance().layerTreeRoot().dump()'` in a stand-alone python script, I only see GROUP nodes. Same command in Python Console in QGIS Desktop displays GROUP and LAYER nodes.

Ex:

Results in stand-alone script

---

GROUP: visible=2 expanded=1

GROUP: myGroup visible=2 expanded=1

---

Results in Python Console

---

GROUP: visible=2 expanded=1

GROUP: myGroup visible=2 expanded=1

LAYER: myRaster visible=2 expanded=1 id=myRaster20140702045512566

---

Best Regards.

## Files

---

modifyqgs.py	1.75 KB	2014-07-01	David AMAR
--------------	---------	------------	------------