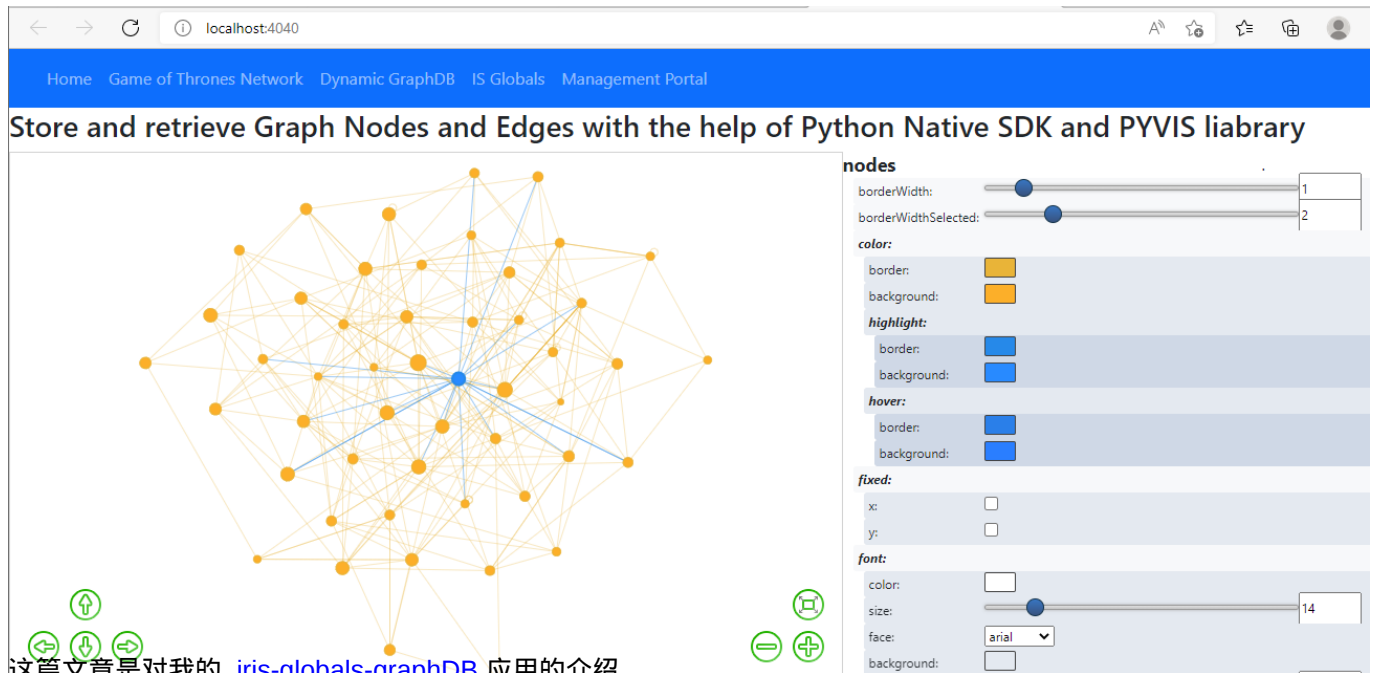


文章

[Michael Lei](#) · 四月 13, 2022 阅读大约需 7 分钟

[Open Exchange](#)

用Globals 作为图数据库来存储和抽取图结构数据



这篇文章是对我的 [iris-globals-graphDB](#) 应用的介绍。

在这篇文章中，我将演示如何在[Python Flask Web](#) 框架和[PYVIS交互式网络可视化库](#)的帮助下，将[图形数据](#)保存和抽取到[InterSystems Globals](#)中。

建议

- 阅读相关文档 [使用 Globals](#)
- [原生 SDK 介绍](#)
- [PYVIS 交互式网络可视化库](#)

第一步：通过使用Python 原生SDK建立与IRIS Globals的连接

```
#create and establish connection
if not self.iris_connection:
    self.iris_connection = irisnative.createConnection("localhost", 1972, "USER"
, "superuser", "SYS")

# Create an iris object
self.iris_native = irisnative.createIris(self.iris_connection)
return self.iris_native
```

第二步：使用 `iris_native.set()` 功能把数据保存到Globals 里

```
#import nodes data from csv file
isdefined = self.iris_native.isDefined("^g1nodes")
if isdefined == 0:
    with open("/opt/irisapp/misc/g1nodes.csv", newline='') as csvfile:

        reader = csv.DictReader(csvfile)
        for row in reader:
            self.iris_native.set(row["name"], "^g1nodes", row["id"])

#import edges data from csv file
isdefined = self.iris_native.isDefined("^g1edges")
if isdefined == 0:
    with open("/opt/irisapp/misc/g1edges.csv", newline='') as csvfile:
        reader = csv.DictReader(csvfile)
        counter = 0
        for row in reader:
            counter = counter + 1
            #Save data to globals
            self.iris_native.set(row["source"]+'-'+row["target"], "^g1edges", counter)
```

第三步: 使用`iris_native.get()` 功能把节点和边缘数据从Globals传递给PYVIS

```
#Get nodes data for basic graph
def get_g1nodes(self):
    iris = self.get_iris_native()
    leverl1_subscript_iter = iris.iterator("^g1nodes")
    result = []
    # Iterate over all nodes forwards
    for level1_subscript, level1_value in leverl1_subscript_iter:
        #Get data from globals
        val = iris.get("^g1nodes",level1_subscript)
        element = {"id": level1_subscript, "label": val, "shape":"circle"}
        result.append(element)
    return result

#Get edges data for basic graph
def get_g1edges(self):
    iris = self.get_iris_native()
    leverl1_subscript_iter = iris.iterator("^g1edges")
    result = []
    # Iterate over all nodes forwards
    for level1_subscript, level1_value in leverl1_subscript_iter:
        #Get data from globals
        val = iris.get("^g1edges",level1_subscript)
        element = {"from": int(val.rpartition('-')[0]), "to": int(val.rpartition(
            '-')[2])}
        result.append(element)
    return result
```

Step4: Use PYVIS Javascript to generate graph data

```
<script type="text/javascript">
  // initialize global variables.
  var edges;
  var nodes;
  var network;
  var container;
  var options, data;

  // This method is responsible for drawing the graph, returns the drawn network
  function drawGraph() {
    var container = document.getElementById('mynetwork');
    let node = JSON.parse('{{ nodes | tojson }}');
    let edge = JSON.parse('{{ edges | tojson }}');

    // parsing and collecting nodes and edges from the python
    nodes = new vis.DataSet(node);
    edges = new vis.DataSet(edge);

    // adding nodes and edges to the graph
    data = {nodes: nodes, edges: edges};

    var options = {
      "configure": {
        "enabled": true,
        "filter": [
          "physics", "nodes"
        ]
      },
      "nodes": {
        "color": {
          "border": "rgba(233,180,56,1)",
          "background": "rgba(252,175,41,1)",
          "highlight": {
            "border": "rgba(38,137,233,1)",
            "background": "rgba(40,138,255,1)"
          },
        },
        "hover": {
          "border": "rgba(42,127,233,1)",
          "background": "rgba(42,126,255,1)"
        }
      },
      "font": {
        "color": "rgba(255,255,255,1)"
      },
      "edges": {
        "color": {
          "inherit": true
        },
        "smooth": {
          "enabled": false,
          "type": "continuous"
        }
      },
      "interaction": {
        "dragNodes": true,
        "hideEdgesOnDrag": false,
        "hideNodesOnDrag": false,
```

```

        "navigationButtons": true,
        "hover": true
    },

    "physics": {
        "barnesHut": {
            "avoidOverlap": 0,
            "centralGravity": 0.3,
            "damping": 0.09,
            "gravitationalConstant": -80000,
            "springConstant": 0.001,
            "springLength": 250
        },

        "enabled": true,
        "stabilization": {
            "enabled": true,
            "fit": true,
            "iterations": 1000,
            "onlyDynamicEdges": false,
            "updateInterval": 50
        }
    }
}
// if this network requires displaying the configure window,
// put it in its div
options.configure["container"] = document.getElementById("config");
network = new vis.Network(container, data, options);
return network;
}
drawGraph();
</script>

```

第五步: 从app.py 主文件调用上面的代码

```

#Mian route. (index)
@app.route("/")
def index():
    #Establish connection and import data to globals
    irisglobal = IRISGLOBAL()
    irisglobal.import_g1_nodes_edges()
    irisglobal.import_g2_nodes_edges()

    #getting nodes data from globals
    nodes = irisglobal.get_g1nodes()
    #getting edges data from globals
    edges = irisglobal.get_g1edges()

    #To display graph with configuration
    pyvis = True
    return render_template('index.html', nodes = nodes, edges=edges, pyvis=pyvis)

```

下面是关于此项目的 介绍视频：

[//player.bilibili.com/player.html?aid=256588464&bvid=BV1LY411w7MU&cid=718938724&page=1](https://player.bilibili.com/player.html?aid=256588464&bvid=BV1LY411w7MU&cid=718938724&page=1)

[这是一个嵌入式链接，但由于您拒绝了访问嵌入式内容所需的

Cookie，您无法直接在网站上进行查看。要查看嵌入式内容，您需要在 Cookie **设置** 中接受所有 Cookie。]

欢迎大家来我们的 [Bilibili主页](#) 观看更多视频!

谢谢！

[#CSS](#) [#Python](#) [#数据库](#) [#数据模型](#) [#竞赛](#) [#Caché](#) [#InterSystems IRIS for Health](#) [#Open Exchange](#) [#VSCode](#)
[在 InterSystems Open Exchange 上检查相关应用程序](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%94%A8globals-%E4%BD%9C%E4%B8%BA%E5%9B%BE%E6%95%B0%E6%8D%AE%E5%BA%93%E6%9D%A5%E5%AD%98%E5%82%A8%E5%92%8C%E6%8A%BD%E5%8F%96%E5%9B%BE%E7%BB%93%E6%9E%84%E6%95%B0%E6%8D%AE>