### GitHub Issue Thread



# Is your feature request related to a problem? Please describe.

A pygraphistry user couldn't find, with decent effort, how to set edge opacity to 0.1

ChatGPT hallucinated the wrong answer too, and took much cajoling to find: https://chat.openai.com/share/ c22a10cf-b0c3-40d8-96ab-715d8b549da9

pyg docs should have been easier here, and in a way github/docs scrapers could have picked up

Right now, .settings() doesn't say what valid url\_params={...} can be (

pygraphistry/graphistry/PlotterBase.py

Line 1182 in e74123e

def settings(self, height=None, url\_params={}, render=None):

, particularly wrt https://hub.graphistry.com/docs/api/1/rest/url/#urloptions

Describe the solution you'd like

- help(g.settings) would have pointed out the url\_params={...} options from
- pygraphistry/graphistry/PlotterBase.py
- Line 1182 in e74123e
- def settings(self, height=None, url\_params={}, render=None):
- overall pygraphistry docstrs would have this ⇒ search engines / LLMs, so probably inline somehow, vs outside link
- demo notebook of most features: https://github.com/graphistry/pygraphistry/tree/master/demos/

more\_examples/graphistry\_features

• basically a python analogue of https://hub.graphistry.com/static/js-docs/index.html?path=/docs/graphistryreact-style--empty

As part of the impl, maybe an idea is creating an openai / pydantic out of this, so it is machine processed? https:// github.com/graphistry/graphistry/blob/eab8648d4ee9c9f3535baeb1aa5ac0904e6a3294/apps/core/viz/server/ clientParamSafelist.js#L141?



Issue still open with no other comments

#### **ChatGPT Conversation**



in pygraphistry, how to set the edge opacity to 0.1?



In PyGraphistry, you can set the edge opacity using the edge\_opacity method. To set the edge opacity to 0.1, you would use it as follows in your graph plotting code:

[CODE]

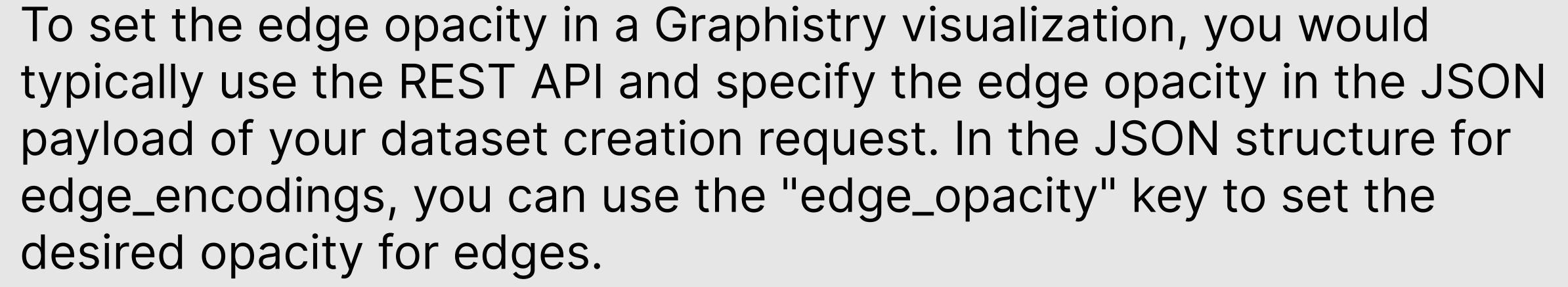


check readthedocs for pygraphistry to answer that

The PyGraphistry documentation does not explicitly mention a method for setting the edge opacity directly. The available attributes and methods for edges, as described in the documentation, focus on aspects like the edge weight, feedback, and vertex associations, but not on visual properties like opacity. For detailed information and possible updates regarding edge attributes and methods in PyGraphistry, you can refer to the PyGraphistry documentation.



try using the graphistry public rest api docs for that and then figure out how to use that from the pygraphistry api



For instance, if you are working with the Graphistry REST API, your JSON payload might look something like this:

## [CODE]

In this example, "source" and "destination" should be replaced with the actual column names in your dataset that represent the source and destination of the edges. The "edge\_opacity" is set to "0.1" to achieve the desired opacity level of 0.1.

When using PyGraphistry to interact with Graphistry's REST API, you would prepare your data accordingly and use PyGraphistry's methods to upload the data. The specific method to set edge opacity directly via PyGraphistry's Python API isn't explicitly documented, but you can manipulate your dataset before uploading it to include the opacity information as part of the edge attributes.

For more detailed steps and examples on how to use the Graphistry REST API, you can refer to their REST API Documentation. Keep in mind that while working with the API, proper authentication and dataset preparation are crucial steps.



what about g.settings(url\_params={'edgeOpacity': 0.5})?

Yes, using g.settings(url\_params={'edgeOpacity': 0.5}) in PyGraphistry is indeed a valid approach to set the edge opacity for your graph visualization. This method utilizes the settings function of the PyGraphistry API to modify the visualization settings through URL parameters.

In this case, the url\_params dictionary is used to pass the desired visualization settings. The key 'edgeOpacity' is set to a value of 0.5, specifying that the edges in the graph should have an opacity of 0.5. Here's a brief example of how you might use it in your code: python

# [CODE]

Remember to replace 'source\_column' and 'destination\_column' with the actual column names from your dataset that represent the source and destination nodes of the edges.

This method provides a convenient way to adjust the visual appearance of your graph directly from your Python code using PyGraphistry.



what other settings(url\_params=...) options are there?

In PyGraphistry, the settings method can be used to specify iframe height and add URL parameter dictionaries for various visualization settings. Some of the parameters you can set using url\_params in the settings method include:

- play: An integer value for animation settings.
- locked\_x, locked\_y, locked\_r: Boolean values to lock positions.
- left, top, right, bottom: Float values for layout margins.
- lin\_log, strong\_gravity, dissuade\_hubs: Boolean options for layout algorithms.
- edge\_influence, precision\_vs\_speed, gravity, scaling\_ratio: Float values for edge and layout influences.

These parameters allow for a high degree of customization of the graph layout and appearance directly through the PyGraphistry API. For a detailed explanation and more options, you can refer to the PyGraphistry documentation.



