Django Graphos Documentation

Release 0.0.2a0

Agiliq

Aug 21, 2017

Contents

Contents:

Django-graphos is a tool to create Graphs. (doh). There are two things which Graphos gives you over a low level graph manupulation.

It provides various data sources.

- SimpleDataSource - Use a Python list
- ModelDataSource
- MongoDataSource

It provides various renderers.

- Flot
- Google charts
- YUI
- Morris.js
- (And more)

Graphos makes it very easy to switch between different data source and renderers.

Are you building your charts with Flot but would like to later switch to Gchart? In many cases, it might be as easy as switching an import statement.

CHAPTER 1

Intro to Django-graphos

CHAPTER 2

Using flot with Django-graphos

Include the js in your html:

```
<script src="{% sttaic 'js/jquery.flot.js' %}"></script>
```

Create a data source.:

```
from graphos.sources.model import ModelDataSource
queryset = Account.objects.all()
data_source = ModelDataSource(queryset,
                              fields=['year', 'sales'])
```

Pass the `data_source` to a flot Chart:

```
from graphos.renderers import flot
chart = flot.LineChart(data_source)
```

You can render this chart in the template by `{{ point_chart.as_html }}`.

Supported chart types

- Line
- Bar
- Point

CHAPTER 3

Using Google chart api with graphos

Include the JS in the template:

```
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
</script>
```

Create a data source.:

```
from graphos.sources.model import ModelDataSource
queryset = Account.objects.all()
data_source = ModelDataSource(queryset,
                                  fields=['year', 'sales'])
```

Pass the `data_source` to a *gchart*:

```
from graphos.renderers import gchart
chart = gchart.LineChart(data_source)
```

You can render this chart in the template by `{{ point_chart.as_html }}`.

Supported chart types

- Area chart
- Bar chart
- Candlestick charts
- Column chart
- Line chart
- Pie chart

CHAPTER 4

Doing Ajax with Graphos

Graphos plays well with ajax interactions. There are two ways you can replace a graph object.

1. Render `chart.as_html` in the views. Return and replace the `DOM`.
2. Calculate the `chart.get_data`, return the JSON. Redraw the chart using `$.plot` or equivalent.

CHAPTER 5

Creating a data source

If you need your chart to get data from a data source we do not natively support, writing a custom data source is easy. Once you do that, the data source can be used in any `renderer`.

To create a new data source

1. Create a class which extends `BaseDataSource` or `SimpleDataSource`
2. Make sure your class has implementation of `get_data`, `get_header` and `get_first_column`
3. `get_data` Should return a NxM matrix (see example data below).

Example Data:

```
data = [
       ['Year', 'Sales', 'Expenses', 'Items Sold', 'Net Profit'],
       ['2004', 1000, 400, 100, 600],
       ['2005', 1170, 460, 120, 310],
       ['2006', 660, 1120, 50, -460],
       ['2007', 1030, 540, 100, 200],
       ]
```

Creating custom charts

You may need to create custom charts in two scenarios:

1. You want to use a charting libary we do not support.
2. You need more control over the html than `chart.as_html` provides.

To customize html for an existing chart type, you will generally create a new template.:

```
from graphos.renderers import gchart

class CustomGchart(gchart.LineChart):
    def get_template(self):
        return "demo/gchart_line.html"
```

To create a chart for a new charting backend, create a new class extending `BaseChart`. This class needs to return the rendrered htmls from `as_html` method.

However in most of the cases you will override the `get_templates` method.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search