

MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DE TURPIAL AUDITOR UI

Esta librería proporciona una forma de visualizar los datos generados por la librería **turpial_auditor**. Tiene un módulo "Reportes" donde se puede listar las auditorías y generar informes de PDF y CSV; y un módulo "Estadísticas" para visualizar los datos por medio de gráficos y conocer el comportamiento de su aplicación.

REQUISITOS

- Python 2.7
- Django 1.10
- django-datatables-view 1.14
- turpial_auditor

INSTALACIÓN

Agregue "turpial_auditor_ui" a su configuración `INSTALLED_APPS` de esta manera:

```
INSTALLED_APPS = (  
    ...  
    'turpial_auditor_ui.auditor_commons',  
    'turpial_auditor_ui.auditor_reports',  
    'turpial_auditor_ui.auditor_statistics',  
    ...  
)
```

Nota: Puede no incluir **auditor_statistics** o **auditor_reports** si no desea las “Estadísticas” o “Reportes”, respectivamente.

MÓDULO ESTADÍSTICAS

Para incluir el módulo "Estadísticas" cree una plantilla, la llamaremos "yourTemplate.html", esta plantilla puede incluir que provee la librería, "turpial_statistics.html", que brinda toda la información relacionada con las estadísticas y su presentación:

yourTemplate.html

```
...
<body>
    ...

    {% include "turpial_statistics.html" %}

    ...
</body>
...
```

Luego, herede la vista **FilterView** en su archivo views.py y habilite una URL que llame a dicha vista:

views.py

```
from turpial_auditor_ui.auditor_statistics.views import FilterView

class YourView(FilterView):

    template_name = 'yourTemplate.html'
```

url.py

```
url(r'^url$', YourView.as_view(), name="other_name"),
```

Incluir gráficos

Puede que no desee utilizar la plantilla que proporcionamos en esta librería, pero es posible que desee utilizar los gráficos sin tener que lidiar con la lógica que lo respalda. Puede incluir solo los gráficos que son útiles para su negocio haciendo algo como:

```
{% include 'turpial_chart.html' with chart=< chart_data > %}
```

donde: <chart_data> es el JSON con la información de datos.

Las variables proporcionadas por la biblioteca son:

- line_chart_data.
- action_pie_chart_data.
- model_pie_chart_data.
- author_pie_chart_data.
- mod_act_chart_data.
- aut_act_chart_data.

Si desea agregar sus propios datos de gráficos, consulte la sección, estas variables deben heredar de ChartData "Crear sus propias variables de datos para gráficos".

Por ejemplo:

```
{% include 'turpial_chart.html' with chart_data=line_chart_data %}
```

Crea tus propios gráficos

Para agregar su propia variable de gráfico de datos, importe la clase **ChartData** y anule el método **get_context_data** agregando una nueva variable que contenga el objeto de su gráfico:

views.py

```
from turpial_auditor_ui.auditor_statistics.utils import ChartData

class YourView(FilterView):

    template_name = 'yourTemplate.html'

    def get_context_data(self, **kwargs):
        context = super(YourView, self).get_context_data(**kwargs)
        stats = build_stats()
        context['your_chart'] = ChartData(stats)
```

Nota: la variable **stats** puede ser de cualquier tipo que se pueda convertir a un JSON válido y que coincida con el **dataProvider** de Amcharts.

Si desea crear un gráfico más complejo, herede la clase ChartData. Asegúrese de ajustar el *type*, *id* y *theme* de su gráfico.

LISTA DE PLANTILLAS DISPONIBLES

- turpial_base.html - Ubicación: commons.
- turpial_statistics.html - Ubicación: estadísticas.
- turpial_chart.html - Ubicación: estadísticas
- turpial_card_stats.html - Ubicación: estadísticas.
- turpial_card.html - Ubicación: estadísticas.
- turpial_report.html - Ubicación: reportes.

Nota: No nombre sus archivos html con ninguno de estos nombres.