

Módulo web_graph_highcharts para la generación de gráficas Highcharts en aplicaciones OpenERP 6.1

Autores

Ing. Liber F. Matos Martín <lfmatos@gmail.com>

Resumen

OpenERP es un sistema de código abierto y multiplataforma para la planificación de los recursos empresariales (ERP, Enterprise Resources Planning, por sus siglas en inglés). El mismo cuenta con un mecanismo para la generación de gráficas a través de la obtención de datos proveniente de la base de datos. Este mecanismo en ocasiones no cumple con todos los posibles gráficos que se desean mostrar en la aplicación. Los tipos de gráficas son principalmente de pastel y de barras, las cuales no permiten la interacción con los usuarios y visualmente no son muy vistosas. Con el objetivo de proporcionar otro mecanismo independiente al ya existente para la generación de gráficos, se ha creado el módulo `web_graph_highcharts`, el cual no sustituye al ya existente sino que se encarga de establecer la relación del framework de desarrollo de OpenERP 6.1 con el framework JavaScript para generación de gráficos, Highcharts. De esta forma se logra la creación de gráficas más vistosas, de diversos tipos, que brinden más información así como interacción visual con los usuarios.

Contenido

1. Módulos disponibles	1
2. Requerimientos	1
3. Proceso de instalación.....	1
4. Forma de uso.....	2
4.1. Vista.....	2
4.2. Modelo	2
5. Aspectos a tener en cuenta.....	6
6. Uso de la opción de exportar a PDF o PNG las gráficas generadas.....	7
7. Extensiones extras adicionadas en el módulo web_graph_highcharts	8
7.1. Traducción	8
7.2. Vista search	8
7.3. Dashboards.....	8
8. Conclusiones.....	9

1. Módulos disponibles

- `web_graph_highcharts`: Módulo encargado de mostrar gráficas utilizando Highcharts. Se adiciona un nuevo tipo de vista llamado `graph_highcharts`.
- `web_graph_highcharts_export`: Módulo que posibilita exportar a formato *png* o *pdf* los gráficos generados. Tiene dependencia con las librerías `pyCairo` y `CairoSVG`.
- `demo_graph_highcharts`: Módulo que muestra el funcionamiento de los módulos anteriores. Muestra algunos [ejemplos](#) existentes del [sitio](#) oficial de Highcharts. Los datos utilizados son estáticos, con el fin solo de mostrar el funcionamiento.

2. Requerimientos

- OpenERP 6.1
- Ejecutarlas en navegadores de versiones recientes y sobre todo que sean compatibles con HTML5 y gráficos SVG.
- `web_graph_highcharts`:
 - Highcharts JS v4.0.1. Incluido en el módulo.
- `web_graph_highcharts_export`:
 - CairoSVG. Incluido en el módulo, fue necesario realizarle algunos cambios para lograr la compatibilidad con los fragmentos de *svg* generados por Highcharts.
 - Librería `pyCairo`:
 - `freetype6`
 - `libcairo-2`
 - `libcairo-gobject-2`
 - `libcairo-script-interpreter-2`
 - `libexpat-1`
 - `libfontconfig-1`
 - `libpng14-14`
 - `zlib1`

Para Windows estás dependencias, son *.dll* y están incluidas en el módulo para su instalación. En el caso de GNU Linux estas dependencias se encuentran en los repositorios. Ver proceso de instalación.

3. Proceso de instalación

El único módulo que tiene alguna peculiaridad en cuanto a su instalación es `web_graph_highcharts_export`. Los otros módulos se instalan de la forma tradicional, no es necesario la instalación ni configuración de ninguna otra dependencia o librería. Para la instalación de `web_graph_highcharts_export` es necesario tener instalado la librería de python `pyCairo`, la cual se encarga de hacer *bindings* a la librería Cairo. Para sistemas operativos basados en GNU Linux, estás librerías se encuentran en los repositorios. Es por eso que en este documento solo se mostrará el procedimiento de instalación en Windows.

- En la ruta `web_graph_highcharts_export\common` se encuentra un archivo llamado `cairo.zip`. Se debe extraer su contenido en alguna ruta válida del `PYTHON_PATH`. En el caso de Windows pudiera ser en `C:\Python27\Lib\site-packages`. Dentro se encuentran las *dlls* necesarias.

- Instalar el módulo `web_graph_highcharts_export`.

4. Forma de uso

4.1. Vista

El módulo adiciona un nuevo tipo de vista llamado `graph_highcharts`. Por lo tanto la sintaxis para su inclusión en una acción es la misma que la de tipo `form`, `tree`, `graph`, etc...

```
<record model="ir.actions.act_window" id="action_graph_hc">
  <field name="res_model">demo.graph.hc</field>
  <field name="view_type">form</field>
  <field name="view_mode">tree, form, graph_highcharts</field>
</record>
```

Ahora se muestra la especificación de la vista:

```
<record model="ir.ui.view" id="view_area_basic_hc_graph">
  <field name="name">demo.graph</field>
  <field name="model">demo.graph.hc</field>
  <field name="type">graph_highcharts</field>
  <field name="arch" type="xml">
    <graph_highcharts string="Título" method="method_exec"/>
  </field>
</record>
```

Es opcional la especificación de campos (`fields`) dentro de `graph_highcharts`. Es decir esto también sería válido:

```
<graph_highcharts string="Título" method="method_exec">
  <field name="campo1"/>
  <field name="campo2"/>
  <field name="campo3"/>
</graph_highcharts>
```

En la definición de la vista para la gráfica aparece el atributo *string*, el cual es opcional y se encarga de adicionar un título al gráfico. Esta posibilidad también puede lograrse retornando un *title* en el diccionario de configuración desde la función del modelo en cuestión. En este ejemplo sería una función llamada *method_exec* perteneciente al modelo `demo.graph.hc`.

4.2. Modelo

De esta forma solo queda definir la parte del negocio que proporcionará los datos a graficar. A la función especificada en la vista se le pasarán parámetros útiles que pueden ser utilizados en el negocio. Esta función se vuelve a ejecutar siempre que se elige un criterio de búsqueda en esa vista, lo que permite recalcular los datos mostrados en las gráficas y estas se actualizan y se repintan con los cambios dados.

Diferentes formas de retornar los datos para graficar

La parte principal que se debe asegurar en la función es el retorno de un diccionario en Python con los aspectos de configuración, representación y datos de la gráfica. Las diferentes opciones y variantes dependen solamente del framework Highcharts, para configurar correctamente y retornar los valores necesarios es aconsejable dirigirse a la [documentación](#) de Highcharts. El módulo no adiciona ninguna opción nueva que pueda ser utilizada en el diccionario retornado. Hay que tener en cuenta que el diccionario retornado es utilizando código Python, los tipos de datos deben mantenerse fieles al lenguaje. El módulo internamente se encarga de transformar este código a JavaScript para que sea válido para el framework Highcharts. A continuación se presentan los diferentes atributos que se pasan a la función y las 3 variantes de retorno correctas que debe tener la función.

Los atributos de la función son los siguientes:

- **cr:** Cursor de conexión.
- **uid:** ID del usuario en ejecución.
- **chart_title:** Título de la gráfica dado en la vista xml.
- **fields:** Campos del modelo especificados en la vista xml.
- **domain:** Lista de condiciones utilizadas en la vista de búsqueda.
- **group_by:** Valores utilizados en los filtros de la vista de búsqueda.
- **context:** Diccionario con los valores tradicionales; además contiene *graph_container*, el cual es el ID auto generado del contenedor del cliente en el que se mostrará el gráfico. Esto puede ser útil en el uso de JavaScript desde Python.

La implementación de la función del modelo (demo.graph.hc) siguiendo el ejemplo sería:

1. Primera variante: Se retorna un solo elemento en la lista que sería el diccionario con los datos y configuraciones de la gráfica. Ver módulo demo_graph_highcharts.

```
def method_exec(self, cr, uid, chart_title, fields, domain, group_by, context=None):
    return [{
        'chart': {
            #Opciones
        },
        'exporting': {
            #Opciones
        },
        'title': {
            #Opciones
        },
        'xAxis': {
            #Opciones
        },
        'yAxis': {
            #Opciones
        }
    }
    ]
```

2. Segunda variante: Se retorna como primer elemento de la lista el diccionario con los datos y configuraciones de la gráfica. En caso de que en la lista se utilice código perteneciente a funcionalidades de Highcharts o código JavaScript, entonces se especifican la ruta completa donde se encuentra en el diccionario con el valor que este tendría. Esto se especificará en una lista de diccionarios como segundo elemento en la lista retornada. Esto se realiza con el objetivo de saber que se va interpretar en JavaScript y donde se ubicará en el diccionario de configuración. Es posible especificar *js_code* como *key* de uno de esos diccionarios, lo que posibilita que el valor no sea añadido al diccionario de configuración en tiempo de ejecución sino que se trata de código JavaScript que será ejecutado. Esto es muy útil para la definición de funciones en JavaScript pero desde Python, un ejemplo pudiera ser los manejadores de eventos de la gráfica en cuestión o funciones alternativas para ser ejecutadas en el cliente. Ver módulo `demo_graph_highcharts`.

```
def method_exec(self, cr, uid, chart_title, fields, domain, group_by, context=None):
    return [ {
        # Diccionario de definición. Ver ayuda de Highcharts.
    },
    [
        { 'key': '', 'value': '' }, #Opcional. Enlace de código
        { 'key': '', 'value': '' } #Opcional. Enlace de código
        { 'key': 'js_code', 'value': '' } #Opcional. Código JS
    ]
    ]
```

Un ejemplo de cómo puede quedar el segundo elemento de la lista retornada pudiera ser este:

```
[
    {
        'key': 'series[0].pointStart',
        'value': 'Date.UTC(2006, 0, 01)'
    },
    {
        'key': 'plotOptions.area.fillColor.stops[0]',
        'value': '[0, Highcharts.getOptions().colors[0]]'
    },
    {
        'key': 'plotOptions.area.fillColor.stops[1]',
        'value': '[1, Highcharts.Color(Highcharts.getOptions().colors[0]).setOpacity(0).get("rgba")]'
    }
]
```

De esta forma significa que existen entradas del diccionario, ubicadas en las rutas especificadas en los *keys* a las cuales se le deben adicionar en tiempo de ejecución el código JavaScript especificado en los *values*. Hay que tener en cuenta que ambos valores son cadenas de texto.

3. Tercera variante: Se retorna *False* como primer elemento de la lista y un diccionario de segundo elemento, el cual tiene un *key* con valor *js_code* y un *value* que representa una cadena que solo contiene código JavaScript. Ver módulo `demo_graph_highcharts`.

```
[ False, {
```

```

'key': 'js_code',
'value': ""
    $('#'+context['graph_container']+'').highcharts({
        chart: {
            // Código JavaScript del diccionario de configuración de Highcharts.
        },
        // Función que se ejecuta después que se genera la gráfica
        function(chart) {
            //Por parámetro se pasa el objeto de la gráfica generada

        }
    });

    function Muestra(){
        // Se pueden especificar otras funciones que pueden ser utilizadas
    }
    ""
}
]

```

En el ejemplo anterior se muestra como se puede utilizar el valor de *graph_container* pasado por el *context*.

Con el objetivo de extender las funcionalidades de Highcharts, así como brindar la posibilidad de utilizar dentro del mismo toda la parte del OpenERP del cliente; se ha añadido un *mixin* al objeto Highcharts llamado *openerp*. Esto permite que dentro del Highchart sea posible el uso de cualquier funcionalidad perteneciente al framework web de OpenERP. Un ejemplo de esto pudiera ser el uso de llamadas asíncronas a funciones pertenecientes a modelos a través de XmlRPC, como se muestra a continuación:

```

Highcharts.openerp.rpc( '/web/dataset/call', {
    model : "MODELO",
    method : "FUNCION",
    args: args //Datos extras }).then( function(result) {

        //CODIGO PARA MANEJAR LA RESPUESTA
    });

```

De esta forma se hace una llamada asíncrona a FUNCION perteneciente a MODELO, se le puede adicionar parámetros extras así como manejar la respuesta obtenida. Esto es solo un ejemplo para mostrar el funcionamiento, lo cierto es que a través de este objeto añadido a Highcharts se puede tener acceso a cualquier funcionalidad. Para obtener documentación sobre el framework web de OpenERP si es necesario remitirse a su documentación oficial o al código fuente. Este documento no abarca ese aspecto. Todo esto se especifica desde el modelo, no desde un archivo con código JavaScript. Además, se le ha añadido al objeto Highcharts la propiedad *graph_container* la cual tiene el mismo valor que el contenido en el *context*. Esta propiedad representa un *ID* auto generado del contenedor del cliente en el que se mostrará el gráfico. De esta forma el desarrollador puede utilizar esta propiedad o la que contiene el *context* de la función. La diferencia es que la propiedad solo es accesible desde

código JavaScript y el contenido en el *context* puede ser utilizado en Python o concatenarlo a la cadena del valor retornado en la función.

5. Aspectos a tener en cuenta

- Evitar llamadas a funciones como *alert*, *confirm*, *prompt* desde Python. Estas funciones detienen la ejecución del código JavaScript hasta obtener respuesta de alguna acción por parte del usuario. Si no se usa consecuentemente pudieran surgir comportamientos no deseados.
- Evitar el uso de funciones para la ejecución automática de código JavaScript, como lo es *setInterval* o cualquier otra variante utilizada. Debido a las características asíncronas de la interfaz OpenERP es probable que el código automático continúe ejecutándose aun cuando la gráfica que lo utilizaba ya no se esté mostrando. Si no se usa consecuentemente pudieran surgir comportamientos no deseados.
- Cuando se usan funciones asíncronas como *rpc* o cualquier otra variante, la ejecución principal no se detiene en espera de la respuesta sino que su ejecución continua. Es aconsejable no esperar respuestas que influyan en el flujo principal debido a que no se asegura en el momento que llegue. Además la intervención del usuario también puede en algún momento causar comportamiento no deseado.
- En el archivo `__openerp__.py` del módulo `web_graph_highcharts` se encuentran con comentarios entradas de archivos `.js` y `.css` que pueden ser utilizados por los desarrolladores. Incluidos se encuentran algunos que tal vez tampoco sean necesarios utilizarlos, en cualquiera de los dos casos el desarrollador puede adaptarlo según su propósito. La inclusión de los archivos que realmente sean necesarios mejora el rendimiento de la aplicación, en ningún caso es aconsejable tener habilitados archivos que no se utilicen. Entre los archivos `.js` que se encuentran deshabilitados están los pertenecientes a los diferentes temas utilizados en Highcharts. Además de algunos módulos de extensión que su uso no es frecuente. En cuanto a los `.css` se encuentran los usados por OpenERP para representar la vista de tipo `graph_highcharts`. Existe además la diferenciación de estilos del botón de la vista según el tema utilizado, en estos momentos solo tiene rojo y azul (que está habilitado).

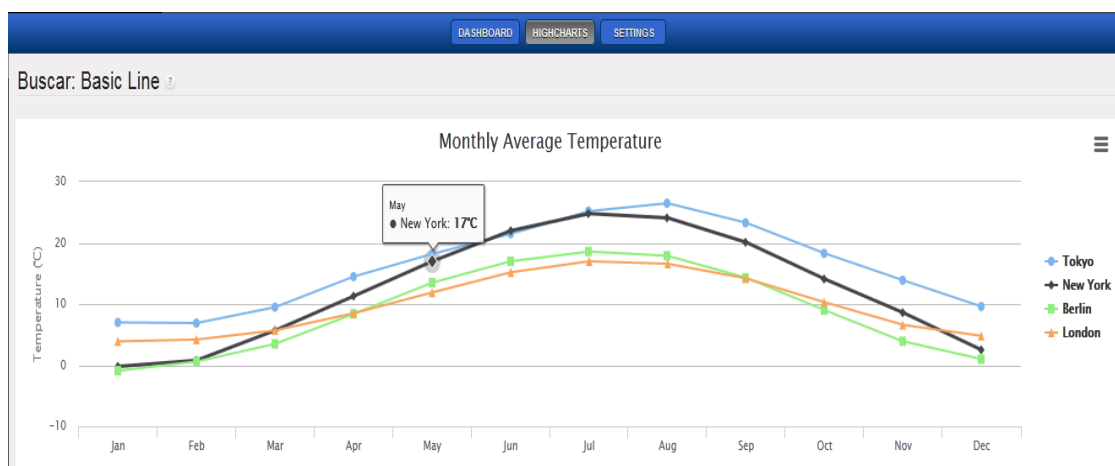


Figura 1. Gráfica Basic Line de demo_graph_highcharts

6. Uso de la opción de exportar a PDF o PNG las gráficas generadas

Originalmente el framework Highcharts tiene un componente que le adiciona este comportamiento; pero solo aporta el botón en la interfaz web y dentro, los enlaces de las opciones de conversión que tiene el usuario. Además se incluyen algunas implementaciones que pudieran ser utilizadas según el lenguaje utilizado del lado del servidor. Por defecto dirige la solicitud para el servidor donde está alojado Highcharts, esto implica que la disponibilidad de internet es necesaria.

El módulo `web_graph_highcharts_export` es el encargado de hacer esta tarea. Se ha utilizado como base el módulo existente en Highcharts pero se le ha añadido la posibilidad de utilizar el servidor OpenERP para la conversión de *svg* a *PDF* o *PNG*, utilizando funciones del framework web y servidor para lograrlo. Se ha mantenido la misma apariencia visual de Highcharts para esto y se le han eliminado algunas opciones de conversión que por el momento no están incluidas. Para la correcta instalación y funcionamiento de este módulo es necesario tener configuradas las dependencias necesarias, especificadas en los [Requerimientos](#). Su correcta configuración está descrita en el [Proceso de instalación](#) del módulo.

Por defecto Highcharts tiene habilitado el uso del exportar. En el módulo se ha invertido esta característica y por defecto está deshabilitado. De esta forma para utilizarlo es necesario especificar en el diccionario de configuración retornado desde el modelo la siguiente entrada:

```
'exporting': {
    'enabled': True,
    'filename': 'archivo',
    'sourceWidth': 800,
    'sourceHeight': 600,
    'url': 'Especificar la url (_cp_path) de un controlador nuevo'
}
```

Como se muestra las posibles opciones son las siguientes:

- **enabled:** Habilita el botón para exportar. Por defecto es *false*.
- **filename:** Nombre del archivo exportado. Por defecto es *chart*.
- **sourceWidth:** Ancho de la imagen exportada. Por defecto es 600.
- **sourceHeight:** Alto de la imagen exportada. Por defecto es 400.
- **url:** Url (*_cp_path*) del controlador que procesará la imagen. Por defecto es */web/export/svg*. Esto indica que es posible implementar otro controlador o redefinir el existente haciendo uso de la herencia.

Una de las librerías utilizadas por este módulo es CairoSVG, la misma se encuentra dentro del módulo porque ha sido necesario hacerle algunas transformaciones para lograr la compatibilidad con el código *svg* que se genera en las gráficas Highcharts y que en ocasiones no siguen el estándar y tipos de datos válidos. Es posible que en algún momento surja otro error no detectado. Si ocurre alguno se puede poner en contacto.

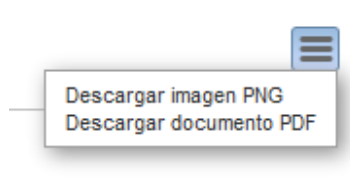


Figura 2. Exportar del ejemplo Basic Line en demo_graph_highcharts

7. Extensiones extras adicionadas en el módulo web_graph_highcharts

7.1. Traducción

En el módulo se utiliza el mecanismo de traducción presente en OpenERP para la traducción de las partes de las gráficas donde se hace necesario, lo que hace posible tener estas interfaces en varios idiomas. Actualmente solo presenta traducción para el idioma inglés y español.

7.2. Vista search

En las vistas *Search* es posible especificar opciones adicionales para lograr comportamiento que no existen originalmente en OpenERP. Estas opciones son las siguientes:

- **searchable:** Oculta o muestra el botón Buscar.
- **clearable:** Oculta o muestra el botón Limpiar.
- **filterable:** Oculta o muestra el combo box Filtrar.

A continuación un ejemplo de su uso:

```
<record id="demo_search" model="ir.ui.view">
  <field name="name">demo.search</field>
  <field name="model">demo.graph.hc</field>
  <field name="type">search</field>
  <field name="arch" type="xml">
    <search options='{ "searchable": false, "clearable": false, "filterable": false}'>
      </search>
    </field></record>
```

7.3. Dashboards

Un uso muy frecuente de las gráficas en OpenERP es precisamente en los Dashboards. Las gráficas Highcharts también pueden ser utilizadas en estas vistas. En el módulo web_graph_highcharts se han adicionado extensiones tanto en la especificación general de un *Board*, como a las *actions* que se configuran. A continuación se muestran ambas variantes:

- **Board:**
 - **changeable:** Oculta o muestra el botón Cambiar disposición.
 - **resetable:** Oculta o muestra el botón Reiniciar.
- **Action:**
 - **closable:** Oculta o muestra el botón de cerrar una acción.
 - **expandable:** Oculta o muestra el botón de minimizar una acción.
 - **draggable:** Deshabilita o habilita la posibilidad de arrastrar una acción.

A continuación un ejemplo ambas extensiones:

```
<board style="1-1-1" changeable="false" resetable="false">
  <column>
    <action name="%{action_hc)d" string="Column" draggable="false" closable="false"
      expandable="false"/>
  </column>
</board>
```



Figura 3. Dashboard del demo_graph_highcharts

8. Conclusiones

Con la creación de los módulos descritos en este documento, se pretende de alguna forma incidir sobre un punto en el que realmente OpenERP 6.1 no tiene muy potente, como son las gráficas. Aspecto muy importante porque en ocasiones su uso se hace necesario y útil para mostrar de forma visual datos de relevancia para el cliente. Highcharts presenta un poderoso mecanismo para el tratamiento y presentación de los datos de forma eficiente, haciendo uso de gran variedad de gráficas, todas muy vistosas y que permiten una buena interacción con el usuario. Es cierto que para la correcta especificación de los datos y la configuración de las gráficas se necesita un buen dominio del framework Highcharts. No obstante existe abundante [documentación](#) y varios [ejemplos](#) disponibles incluso por el propio framework Highcharts. Además, en módulo demo_graph_highcharts se muestra cómo obtener estas gráficas en una aplicación OpenERP real. Puede ocurrir perfectamente que algunos de estos módulos presenten algún error que no hemos detectado o que se le pueda adicionar cambios de utilidad. En cualquier caso estamos dispuestos a escuchar sugerencias o propuestas de cambios.