

vGAC

vGAC (vGoogle Api Chart) es un wrapper para V7 de la librería ofrecida por Google para generar charts online.

Permite mostrar el gráfico generado en un control Html propio de Velneo V7, grabar la imagen a fichero o guardar la imagen generada dentro de una tabla para después visualizarla o imprimirla.

Qué contiene

Contiene un proyecto de datos con todas las funciones necesarias para su uso. Además incluye una tabla usada para guardar los charts generados.

Además se incluye un frontend que muestra como utilizar vGAC y los distintos tipos de gráficos que se pueden crear.

Posibles usos

Heredando nuestra aplicación final de vGAC podemos crear fácilmente gráficos de barras, tartas, líneas, etc y mostrarlas en nuestra aplicación o imprimirlas en los informes.

Detalle

El API de Google para crear charts se basa en la creación de una url con todos la configuración y datos necesarios para componer el chart. La url es algo críptica por lo que hemos hecho es definir unas funciones mínimas para esa url. Además, para poder usar todas las funcionalidades que aporta el API de Google, existe una función que permite añadir la configuración extra que deseemos a la url antes de solicitar el chart. De manera que podamos usar la documentación ofrecida por Google <http://code.google.com/intl/es-ES/apis/chart/>, el formato de los parámetros que usaremos en las funciones creadas se usa tal y cómo se indica en la guía de Google.

Requisitos

Conexión a Internet

Cómo usarla

Veamos cómo usar esta Open App.

Ejemplo sencillo

Con este ejemplo tendremos en una variable local la url necesaria para por ejemplo mostrar el chart en un control html.

El código mínimo para generar un chart es el siguiente:

```

X Set ( ID_CHART, fun:CREATE@vGAC.dat("p", "200x100") )
X Set ( ID_CHART, fun:SET_DATASETS@vGAC.dat(ID_CHART, "60,40") )
X Set ( URL, fun:GET_URL@vGAC.dat(ID_CHART) )

```

Este código creará el siguiente chart:



CREATE(type, size)

type: el tipo de chart a usar según los que se indican en http://code.google.com/intl/es-ES/apis/chart/#chart_type.

size: ancho y alto en formato "ancho x alto".

Por ejemplo si hacemos `CREATE("p", "200x100")`, estamos creando un gráfico de tipo Pie de 200 pixels de ancho por 100 de alto.

Esta función crea una entrada en la tabla charts, donde se almacenan los datos de este chart y si lo deseamos se guarda la imagen resultante.

Retorna el ID único de la tabla charts.

SET_DATASETS(ID_CHART, DATA)

id_chart: id de este chart en la tabla charts.

data: Datos para el chart que estamos creando.

Los datos se pasan de una forma de u otra en función al tipo de chart. Para ver más detalles sobre esto podemos usar el frontend que se suministra como ejemplo o consultar la guía

<http://code.google.com/intl/es-ES/apis/chart/>

Retorna el ID del chart de tal manera que podríamos encadenar funciones, por ejemplo, GET_URL(SET_DATASETS(..., ...))

GET_URL(ID_CHART)

Retorna la url que debemos invocar para obtener el chart que hemos creado. Si pasamos esta variable local a un control html de Velneo V7, nos mostraría el chart generado.

Ejemplo más completo

Veamos ahora como crear un ejemplo un poco más complejo que incluye varias series, titulo, colores, leyenda, etc.

```

 Set ( ID_CHART, fun:CREATE@vGAC.dat(TYPE, "" )
 Set ( ID_CHART, fun:SET_SIZE@vGAC.dat(ID_CHART, WIDTH, HEIGHT) )
 Set ( ID_CHART, fun:SET_DATASETS@vGAC.dat(ID_CHART, DATA) )
 Set ( ID_CHART, fun:SET_TITLE@vGAC.dat(ID_CHART, TITLE, TITLE_COLOR, TITLE_SIZE) )
 Set ( ID_CHART, fun:SET_LEGEND@vGAC.dat(ID_CHART, LEGEND, "" )
 Set ( ID_CHART, fun:SET_COLORS@vGAC.dat(ID_CHART, COLORS) )
 Set ( ID_CHART, fun:SET_LABELS@vGAC.dat(ID_CHART, LABELS) )
 Set ( ID_CHART, fun:SET_ADITONAL_INFO@vGAC.dat(ID_CHART, ADITONAL_INFO) )
 Set ( URL, fun:GET_URL@vGAC.dat(ID_CHART) )
 Set ( URL_PREVIEW, URL )
    
```

Utilizando el frontend (se incluye explicación de uso más abajo), podemos ver cómo hay que pasar los datos en función al tipo de gráfica.

Veamos las funciones que no hemos comentado anteriormente.

La variable ID_CHART siempre indica el ID único en la tabla de charts para el chart que estamos creando. Para obtener más información sobre los valores a pasar consultar la guía de Google

<http://code.google.com/intl/es-ES/apis/chart/>

SET_SIZE(ID_CHART, WIDTH, HEIGHT): Definie el ancho y el alto en pixels para este chart. El máximo son 300.000 pixels, o sea, width x height <= 300.000.

Mirar http://code.google.com/intl/es-ES/apis/chart/#chart_size

SET_TITLE(ID_CHART, TITLE, TITLE_COLOR, TITLE_SIZE): Permite indicar el título, color del título y tamaño de la fuente del título de este chart. Mirar <http://code.google.com/intl/es-ES/apis/chart/#chtt>

SET_LEGEND(ID_CHART, LEGEND, LEGEND_POSITION): Permite especificar la leyenda a usar y la posición de la misma. Mirar <http://code.google.com/intl/es-ES/apis/chart/#chdl>

SET_COLORS(ID_CHART, COLORS): Permite indicar los colores del chart. Mirar http://code.google.com/intl/es-ES/apis/chart/#line_bar_pie_colors

SET_LABELS(ID_CHART, LABELS): Permite asignar etiquetas al chart. Mirar http://code.google.com/intl/es-ES/apis/chart/#pie_labels

Si necesitamos añadir configuración adicional para el gráfico, por ejemplo, rellenos o gradientes, contamos con una función donde podemos añadir ese tipo de información:

SET_ADDITIONAL_INFO(ID_CHART, ADDITIONAL_INFO): Agregamos información adicional al chart, por ejemplo "*chf=c,lg,0,76A4FB,1,ffffff,0|bg,s,EFEFEF*", especifica un gradiente azul de izquierda a derecha.

Funciones adicionales

Una vez que tenemos la url generada con la llamada a la función GET_URL, podemos hacer algo de lo siguiente:

Grabar el chart a disco

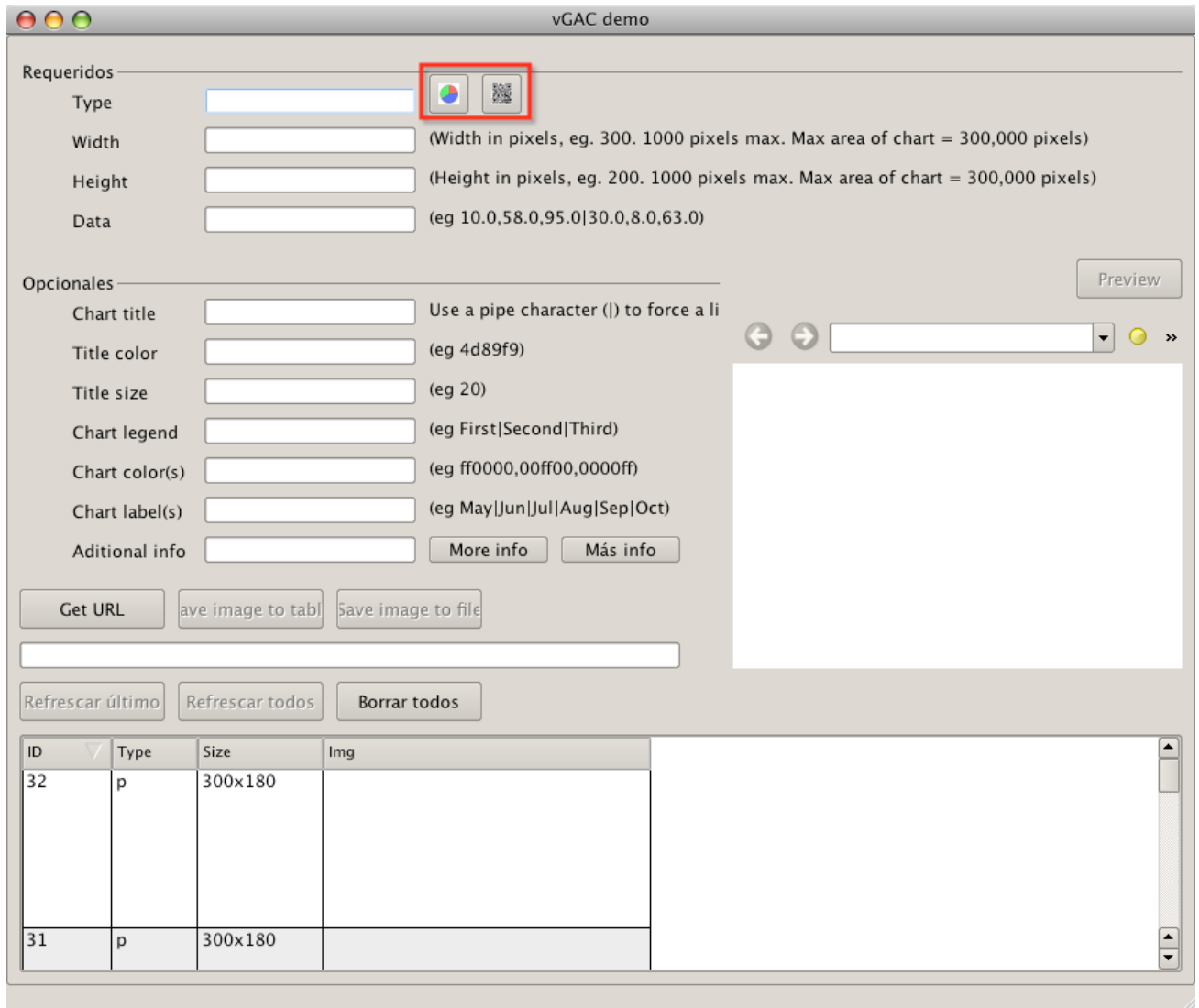
Usando la función SAVE_IMAGE_TO_FILE (ID_CHART, FILE), grabamos el chart asociado al fichero especificado por el parámetro FILE.

Guardar el chart en la tabla

Con la función SAVE_IMAGE(ID_CHART), guardamos el chart generado en la tabla de Charts, asociado al Id ID_CHART.

Frontend de ejemplo

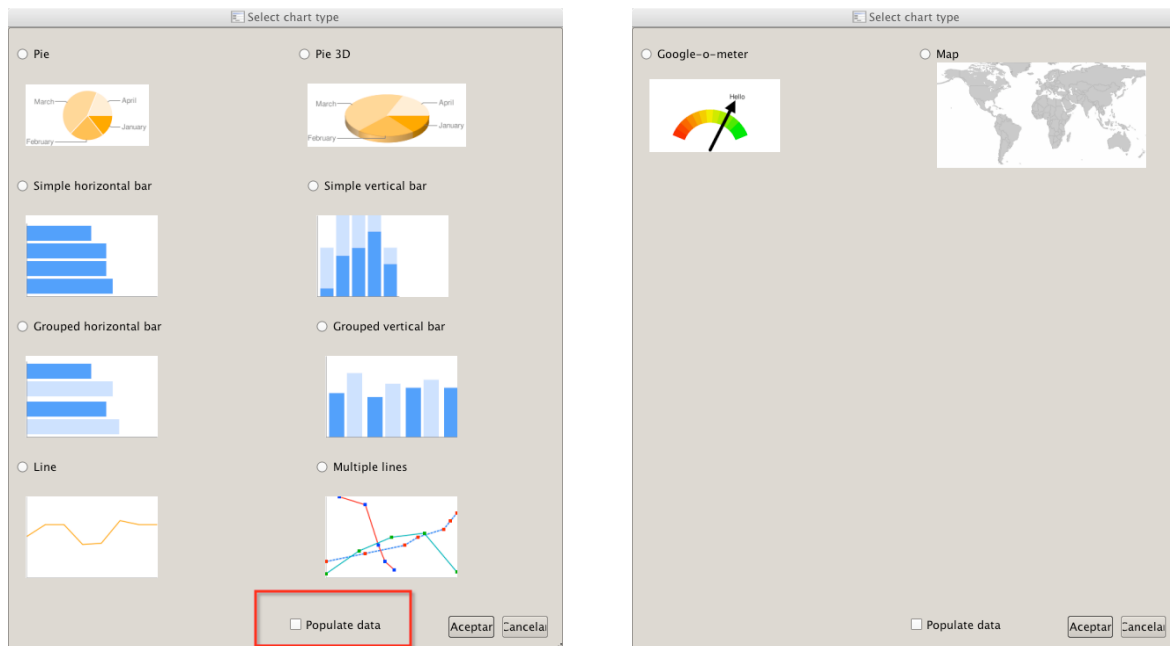
Hemos creado un frontend que permite crear varios tipos de charts y así ver las posibilidades de esta Open App. Cuando iniciamos la aplicación nos encontramos con la siguiente ventana:



The screenshot shows a web application window titled "vGAC demo". It features a form with two main sections: "Requeridos" (Required) and "Opcionales" (Optional). The "Requeridos" section includes fields for "Type" (with a color and QR code icon highlighted in a red box), "Width", "Height", and "Data". The "Opcionales" section includes fields for "Chart title", "Title color", "Title size", "Chart legend", "Chart color(s)", "Chart label(s)", and "Additional info". There are also buttons for "Get URL", "ave image to tabl", "Save image to file", "Refrescar último", "Refrescar todos", and "Borrar todos". A table at the bottom displays a list of items with columns for ID, Type, Size, and Img.

ID	Type	Size	Img
32	p	300x180	
31	p	300x180	

Usando los dos botones que aparecen en el cuadrado rojo, podemos seleccionar el tipo de gráfico que queremos generar. Disponemos de estos tipos:



Si marcamos el checkbox "Populate data", la aplicación nos creará unos datos de ejemplo para generar el gráfico. Posteriormente, en la pantalla inicial, podremos adaptarlos a nuestro gusto.

Si seleccionamos por ejemplo el Pie en 3D y marcamos "Populate data", nos aparecen los siguientes datos.

Para generar la url y que aparezca un preview hacemos click en el botón "Get URL":

Requeridos

Type:

Width: (Width in pixels, eg. 300. 1000 pixels max. Max area of chart = 300,000 pixels)

Height: (Height in pixels, eg. 200. 1000 pixels max. Max area of chart = 300,000 pixels)

Data: (eg 10.0,58.0,95.0|30.0,8.0,63.0)

Opcionales

Chart title: Use a pipe character (|) to force a li

Title color: (eg 4d89f9)

Title size: (eg 20)

Chart legend: (eg First|Second|Third)

Chart color(s): (eg ff0000,00ff00,0000ff)

Chart label(s): (eg May|Jun|Jul|Aug|Sep|Oct)

Additional info:

ID	Type	Size	Img
34	p3	300x180	
33	p3	300x180	

Si hacemos click en "Save image to table", nos guardará este chart en la tabla de charts.

ID	Type	Size	Img
37	p3	300x180	
36	p3	400x220	