

# wArrayBrowse

Con wArrayBrowse podemos crear una tabla browse de solo lectura, esta basado el las tablas HTML con las propiedades que le añade Materialize y un poco de "tuneado" por nuestra parte.

Admite un array (altems) con los datos a presentar y otro con las comunas (aCols) pero en este caso lo que definimos es el aspecto de cada columna y la cabecera de ellas, para esto crearemos controles [wItemColBrowse](#) donde podremos definir las. También podemos crearle pies a las columnas que compartirán aspecto con las cabeceras con el array aFooter.

Con [cOnClick](#) podemos definir un Función de nuestra App que sera llamada al picar con el ratón en cualquiera de las celdas, desde nuestra App recuperamos la linea en la que se ha picado leyendo el Parámetro que corresponde al cid del wArayBrowse, en caso de usar la llamada por Ajax también nos devolverá [nRowId](#) con la linea seleccionada.

Como todos los controles que permiten eventos, se puede utilizar [aParams](#) para añadir más parámetros y sus valores al generar el evento.

## Ejemplo de uso

### Este es el código para crear el wArrayBrowse del Ejemplo

```
WITH OBJECT WArrayBrowse():New(oPanel)
  :cId      := "Bancos"           // Id que le damos al ArrayBrowse
  :nStyle   := xc_Striped        // Estilo visual, con pautado pijama
  :cClrHeadPane := "red lighten-4" // Le damos color a las cabeceras
  :cOnClick  := "VisParam"       // En caso de picar en un elemento del
ArrayBrowse
  // Añadimos Columnas
  WITH OBJECT :AddCol("Código") AS wItemColBrowse
    :cClrPane := "blue darken-1" // Puedo darle color al fondo de la
columna
    :cClrText := "white"         // Y al Texto
  END WITH
  WITH OBJECT :AddCol("BIC") AS WItemColBrowse
    :lBold := .T.                // Podemos asignar la letra en Negrita
  END WITH
  :AddCol("Entidad",,xc_Left)
  // Le alimentamos datos directamente de la consulta
  :aItems := AsciiHTML(aConsulta) // Estos datos se han extraído de una
consulta SQL a MariaDB
  // Y creamos el ArrayBrowse
  :Create()
END WITH
```

## Y para recuperar el indice de la linea que se ha picado...

```
LOCAL nLin := Val( oCGI:GetCgiValue("bancos",0) )
```

## Propiedades

Nombre	Init	Descripción	Versión
altems		Array de datos a presentar { {Col1,Col2,ColX},{Col1,Col2,ColX} }	0.1
aCols		Array de <a href="#">wItemColBrowse</a>	0.1
nStyle	<a href="#">xc_NoStyle</a>	Tipo de Tabla <a href="#">xc_Striped</a> (pijama) ó <a href="#">xc_Highlight</a> (resaltada al seleccionar)	0.1
lResponsive		Habilita la tabla como responsive para los móviles	0.1
cClrHeadPane		Color del la Cabecera y del Pie	0.1
cClrHeadText		Color del Texto de la Cabecera y del Pie	0.1
nFontSizeHead	6	Tamaño de la Fuente de la Cabecera y el Pie según <a href="#">Materialize</a>	0.1
nFontSizeItem	6	Tamaño de la Fuente del Detalle según <a href="#">Materialize</a>	0.1
cCursor	"pointer"	<a href="#">Cursor</a> al colocar ratón, solo si hay <a href="#">cOnClick</a>	0.1
<a href="#">cOnClick</a>		Funcion del CGI que se ejecuta al hacer click	0.1
<a href="#">aParams</a>		Parámetros extras a enviar al CGI { {"Param1","Valor1"}, {"Param2","Valor2"} }	0.1
<a href="#">cAjaxBevel</a>		Bevel donde se incrusta el resultado de la llamada al <a href="#">Ajax</a>	0.1
<a href="#">cAjaxWaitOn</a>		Incluye un indicador de actividad mientras se ejecuta la función <a href="#">Ajax</a>	0.1
cTextNoData	"No hay Datos"	Texto que aparece cuando el altems esta vacío	0.1
aFooter		Crea una fila al final con el color de la cabecera {"Pie1","Pie2",...}	0.1

## Metodos

Nombre	Descripción	Versión
<a href="#">AddCol</a>	Permite añadir elementos a aCols	0.1
<a href="#">AddRow</a>	Permite añadir elementos a altems	0.1

## Hereda de...

## wControl

## Propiedades

Nombre	Init	Descripción	Versión
<a href="#">cClassId</a>		Identificador del control para generar el Id	0.1
<a href="#">cId</a>	Nil	Identificador único para cada Control	0.1
<a href="#">oParent</a>		Control contenedor	0.1
<a href="#">oHtml</a>		Objeto de la clase <a href="#">wHtml</a> con el código para su renderizado	0.1
<a href="#">oStyle</a>		Objeto de la clase <a href="#">wStyle</a> con las variantes de estilo	0.1
<a href="#">IEnabled</a>	.t.	El control aparece Habilitado o Inhabilitado	0.1
<a href="#">IHide</a>	.f.	El control aparece Oculto o Visible	0.1
<a href="#">aWidth</a>	{12,,}	Ancho del control según el tamaño de la pantalla	0.1
<a href="#">aOffset</a>	{,,}	Margen izquierdo según el tamaño de la pantalla	0.1
<a href="#">cClrText</a>	black	Color de Texto	0.1
<a href="#">cClrPane</a>	white	Color del Fondo	0.1

10/11/2024 12:05

## Métodos

Nombre	Descripción	Versión
Create	Este es el método que crea cada uno de los controles, esta desarrollado en cada Control	0.1
<a href="#">FullHtml</a>	Nos devuelve una cadena con todo el código HTML y JS necesario para crear el control	0.1
<a href="#">LogDebug</a>	Nos Añade un mensaje a la consola del navegador cuando estamos en modo Debugger	0.1
<a href="#">Assert</a>	Nos Añade un comentario en el HTML cuando estamos en modo Debugger	0.1
<a href="#">AddHTML</a>	Nos Añade código HTML dentro del control	0.1
<a href="#">AddParam</a>	Añade un parámetro a la propiedad <a href="#">aParams</a> , hay que pasarle un array {"nombre","valor"}	0.2

10/11/2024 12:05

From:  
<https://www.nefele.dev/wiki/> - **Nefele Project**

Permanent link:  
<https://www.nefele.dev/wiki/controles/warraybrowse?rev=1578433775>

Last update: **21/07/2020 10:50**

