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 (aItems) 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 definirlas. 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 wArryBrowse, 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

```
  WITH OBJECT  WArrayBrowse():New(oPanel)
    :cId         := "Bancos"         // Id que le damos al ArrayBrowse
    :nStyle      := xc_Striped       // Estilo visual, con pautado pijama
    :cOnClick    := "VisParam"       // En caso de picar en un elemento del
ArrayBrowse
    // Añadimos Columnas
    WITH OBJECT :AddCol("Código") AS wItemColBrowse
      :cClrPane := "red"            // 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 extraen de una
consulta SQL a MariaDB
    // Y creamos el ArrayBrowse
    :Create()
  END WITH
```

```
  // Para recuperar el indice de la linea que se ha picado
  LOCAL nLin := Val( oCGI:GetCgiValue("bancos",0) )
```

Propiedades

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

Metodos

Nombre	Descripción	Versión
AddCol	Permite añadir elementos a aCols	0.1
AddRow	Permite añadir elementos a aItems	0.1

wControl

Propiedades

Nombre	Init	Descripción	Versión
cClassId		Identificador del control para generar el Id	0.1
cId	Nil	Identificador único para cada Control	0.1
oParent		Control contenedor	0.1
oHtml		Objeto de la clase wHtml con el código para su renderizado	0.1
oStyle		Objeto de la clase wStyle con las variantes de estilo	0.1
lEnabled	.t.	El control aparece Habilitado o Inhabilitado	0.1
lHide	.f.	El control aparece Oculto o Visible	0.1
aWidth	{12,,,}	Ancho del control según el tamaño de la pantalla	0.1

Nombre	Init	Descripción	Versión
aOffset	{,,,}	Margen izquierdo según el tamaño de la pantalla	0.1
cClrText	black	Color de Texto	0.1
cClrPane	white	Color del Fondo	0.1

14/11/2019 14:37 · fredy

Métodos

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

14/11/2019 14:37 · fredy

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

Permanent link:
https://nefele.dev/wiki/controles/warraybrowse?rev=1578259595

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