

Generated Documentation



Contents

Package util Classes	1
Class grid	1
Constructor grid	3
Method ancho	3
Method autofiltro	4
Method botonEliminar	4
Method botonInsertar	4
Method botonModificar	5
Method campoFuncion	5
Method campoLookUp	5
Method campoOrden	6
Method camposNoVisibles	6
Method camposSoloLectura	7
Method debugger	7
Method encabezados	7
Method enlacenicio	8
Method filtro	8
Method hints	8
Method mostrar	9
Method paginador	9
Method rutalmagenes	9
Method titulo	10
Method tituloCampo	10
Method total	10
Method trigger	11
Method unidadMedida	11
Appendices	12
Appendix A - Class Trees	13
util	13
Appendix D - Todo List	14

Package util Classes

Class grid

[line 95]

GRID Class Copyright(C) 2005-2008 Isaluso

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Esta es una clase que permite realizar altas, bajas, modificaciones y consultas de una tabla de Interbase/Firebird. Necesita la extensión de php para el acceso a la BD Interbase/Firebird a [php_interbase].

History: 1.8 [22/01/2008]

- Mejorado el ordenamiento por columnas
- Librería autónoma de funciones util.php
- Internacionalización [es,en]
1.7 [15/01/2008]
- Se corrigió ciclo infinito en la función encabezados
- Se cambió el estilo del filtro => más semántico
- Se cambió el autofiltro " a 'Sin Valor'
- Modificación del paginador, aplicación de algunas técnicas de usabilidad
1.6.1 [05/06/2007]
- Se corrigió el POST de INSERCIÓN/MODIFICACIÓN para que agregue GET[] faltantes
1.6 [23/04/2007]
- Se agregó función para mostrar hints de campos
- Solucionado el autofiltro cuando es campo 0
1.5 [07/02/2007]
- Validación accesible: JavaScript para validar campos obligatorios

- Validación en envío de formulario y en blur de campos: float, integer, date, timestamp
- Se agregaron LABELS asociados a los INPUTS de formularios
- Se agregó método unidadMedida(), para mostrar cadena con nombre de la unidad de medida
 - 1.2.1 [23/01/2007]
- Corregido bug que impide consultas en FireBird 2.0
 - 1.2 [08/01/2007]
- Agregado función de totalización de campos en el pie de la tabla
- Corregido el redondeo de decimales utilizado en la función formatoNumero
 - 1.1.1 [01/01/2007]
- Queda seleccionado por defecto en los combobox el valor del autofiltro (cuando se hace una inserción)
 - 1.1 [20/12/2006]
- Se agrega opción de ocultar el botón de inserción y el de eliminación
- Solucionado el problema de clave primaria en la posición 0
 - 1.0 [29/11/2006]
- Se modificó todos los comentarios para hacerlo compatible con PHPDoc
- Se separó diseño de código
- Se creó un CSS para darle estilo
- Se limpió el código para hacerlo más semántico
- Se modificó el comportamiento de las columnas de ordenación haciéndolas similares a Yahoo
- Hacemos foco en el primer elemento del formulario
- Utilización de estilo de Cuadro de diálogo
 - 0.9.6 [09/11/2006]
- Se cambió el componente de ingreso en los timestamp
- Se agregó debug de queries
- Se corrigió bug en BOOLEANS y tipo de dato FLOAT, TINYMCE en Textareas repetidos, BLOBS nulos
- Se agregó función para fijar títulos de todos los campos
- Se agregó función def por usuario que creará el componente a utilizar en el Insert de un campo
- El identificador de \$this->campos es ahora un autoincremental, no el position del campo
- Se agregó función para configurar campos como de sólo lectura, no modificables
 - 0.9.5 [03/11/2006]
- Se solucionó la creación del enlace de la ordenación de columnas cuando se aplica filtro
- Se agregó opción de autofiltro por campo que es clave foránea
 - 0.9 [10/10/2006]
- Se agregó un límite (WHERE) para mostrar en los campos LookUp
- Solucionado los checkbox de los campos booleanos
 - 0.8.2 [10/07/2006]
- Se agregó el símbolo ... en paginador cuando se excede la cantidad de vínculos a mostrar
 - 0.8.1 [05/07/2006]
- Se utiliza otra clase de calendario (<http://www.dynarch.com/projects/calendar/>)
 - 0.8 [29/06/2006]
- Se agregó arrastre de los parámetros entre páginas
- El paginador ahora muestra hasta un límite de links, no todos

- **Package** util
- **Author** Isaluso
- **Version** 1.8
- **TODO** Lista de tareas a realizar:
 - Quitar páginas intermedias de mensajes de modificaciones satisfactorias
 - Permitir varias eliminaciones simultáneas, utilizando checkbox
 - Limpieza de código
 - Documentación y variables en ingles
 - Mejoras Ajax
 - JavaScript menos intrusivo
 - Permitir más de una instancia de grid por página

Constructor *void* function grid::grid(\$bd, \$user, \$pass, \$tabla) [line 277]

Function Parameters:

- *string* **\$bd** nombre y ruta de la base de datos
- *string* **\$user** nombre del usuario
- *string* **\$pass** contraseña
- *string* **\$tabla** nombre de la tabla en la base de datos

Constructor: se realizan las asignaciones a la variable \$campos

- **See** campos

void function grid::ancho(\$ancho) [line 882]

Function Parameters:

- *integer* **\$ancho** ancho en porcentaje que se aplicará a toda la grilla

Método para asignar el ancho porcentual de la tabla

- **Access public**

void function grid::autofiltro(\$campo) [line 1013]

Function Parameters:

- *string* **\$campo** nombre del campo de la tabla que será autofiltro

Método para fijar el campo que tendrán autofiltro para filtrar la grilla. Hay que llamar una vez a la funcion por cada campo.

- **Access public**

void function grid::botonEliminar([\$mostrar = true]) [line 1110]

Function Parameters:

- *boolean* **\$mostrar true**: se muestra

Método para determinar si se muestra el botón de eliminar registro

- **Access public**

void function grid::botonInsertar([\$mostrar = true]) [line 1101]

Function Parameters:

- *boolean* **\$mostrar true**: se muestra

Método para determinar si se muestra el botón de insertar registro

- **Access** public

void function grid::botonModificar([\$mostrar = true]) [*line 1119*]

Function Parameters:

- *boolean* **\$mostrar** true: se muestra

Método para determinar si se muestra el botón de modificar registro

- **Access** public

void function grid::campoFuncion(\$campo, \$funcion) [*line 1055*]

Function Parameters:

- *string* **\$campo** nombre del campo de la tabla
- *string* **\$funcion** nombre de la función

Método para asociar un función a un campo para llamara cada vez que se muestre el valor del campo

- **Access** public

void function grid::campoLookUp(\$campo, \$look, [\$where = ""]) [*line 970*]

Function Parameters:

- *string* **\$campo** nombre del campo de la tabla
- *string* **\$look** nombre del campo en la tabla foránea que se mostrará en el OPTION

- *string* **\$where** filtro que se podría aplicar a la tabla foránea para no mostrar todos los registros

Método para crear un LookUpComboBox (SELECT) en los campos que son Foreign Key

- **Access public**

void function grid::campoOrden(\$campo, [\$orden = 1]) [*line 986*]

Function Parameters:

- *string* **\$campo** nombre del campo de la tabla
- *string* **\$orden** orden 1: ASC, 2:DESC

Método para indicar el campo por el que se ordenará el grid si no se eligió en el encabezado

- **Access public**

void function grid::camposNoVisibles(\$campos) [*line 936*]

Function Parameters:

- *array* **\$campos** arreglo con los nombres de los campos que no serán visibles

Método para asignar los campos que no serán visibles en el grid

- **Access public**

void function grid::camposSoloLectura(\$campos) [line 952]

Function Parameters:

- *array* **\$campos** arreglo con los nombres de los campos que no se modificarán

Método para configurar si un campo es de sólo lectura

- **Access public**

void function grid::debugger([\$accion = true]) [line 1044]

Function Parameters:

- *boolean* **\$accion**

Método para comenzar el modo de debug para corrección de errores que mostrará los resultados de los querys

- **Access public**

void function grid::encabezados(\$titulos) [line 892]

Function Parameters:

- *array* **\$titulos** arreglo con los títulos de cada columna

Método para asignar los títulos del grid, pero solamente a los campos visibles

- **Access public**

`void function grid::enlaceInicio($link) [line 846]`

Function Parameters:

- *string* **\$link** enlace a la página de inicio

Método para crear un enlace a una página de inicio

- **Access public**

`void function grid::filtro($filtro, [$titulo = "]) [line 856]`

Function Parameters:

- *string* **\$filtro** cadena con el filtro a aplicar
- *string* **\$titulo** cadena que se mostrará como título del filtro en la grilla

Método para crear un filtro a modo de WHERE en la tabla y de ese forma no mostrar todos los registros

- **Access public**

`void function grid::hints($campo, $hint, $mensaje) [line 1086]`

Function Parameters:

- *string* **\$campo** nombre del campo de la tabla
- *string* **\$mensaje** cadena con el hint
- **\$hint**

Método para asignar un mensaje tipo hint a un campo

- **Access public**

void function grid::mostrar() [line 1698]

Método que según la ACCION llama al método correspondiente

- **Access public**

void function grid::paginador([\$nrofilas = 20]) [line 837]

Función Parameters:

- *integer* **\$nrofilas** filas por página

Método que se invoca para que el resultado se vea con una barra de paginación

- **Access public**

void function grid::rutalmagenes(\$ruta) [line 872]

Function Parameters:

- *string* **\$ruta** cadena de la ruta al directorio de imágenes dentro del servidor

Método para asignar la ruta absoluta de las imágenes

- **Access public**

void function grid::titulo(\$titulo) [line 926]

Function Parameters:

- *string* **\$titulo** titulo de la tabla

Método para asignar el CAPTION de la tabla

- **Access public**

void function grid::tituloCampo(\$titulos) [line 909]

Function Parameters:

- *array* **\$titulos** arreglo con los títulos de cada columna

Método para asignar los títulos de todos los campo de la tabla

- **Access public**

void function grid::total(\$campos) [line 1027]

Function Parameters:

- *array* **\$campos** arreglo con los nombres de los campos que no se totalizarán

Método para determinar si se mostrará un pie de tabla con el total de campo

- **Access public**

void function grid::trigger(\$campo) [line 999]

Function Parameters:

- *string* **\$campo** nombre del campo de la tabla

Método para determinar el campo que posee un trigger => no se mostrará el campo para insertar

- **Access public**

void function grid::unidadMedida(\$campo, \$mensaje) [line 1071]

Function Parameters:

- *string* **\$campo** nombre del campo de la tabla
- *string* **\$mensaje** cadena con la unidad de medida que se mostrará

Método para asignar una cadena con el nombre de la unidad de medida del campo

- **Access public**

Appendices

Appendix A - Class Trees

Package util

grid

- [grid](#)

Appendix D - Todo List

In Package util

In [grid](#)

- Lista de tareas a realizar:
 - Quitar páginas intermedias de mensajes de modificaciones satisfactorias
 - Permitir varias eliminaciones simultáneas, utilizando checkbox
 - Limpieza de código
 - Documentación y variables en ingles
 - Mejoras Ajax
 - JavaScript menos intrusivo
 - Permitir más de una instancia de grid por página

Index

C

constructor <code>grid::grid()</code>	3
<i>Constructor: se realizan las asignaciones a la variable \$campos</i>	

G

grid::mostrar()	9
<i>Método que según la ACCION llama al método correspondiente</i>	
grid::paginador()	9
<i>Método que se invoca para que el resultado se vea con una barra de paginación</i>	
grid::hints()	8
<i>Método para asignar un mensaje tipo hint a un campo</i>	
grid::filtro()	8
<i>Método para crear un filtro a modo de WHERE en la tabla y de ese forma no mostrar todos los registros</i>	
grid::enlaceInicio()	8
<i>Método para crear un enlace a una página de inicio</i>	
grid::rutaImágenes()	9
<i>Método para asignar la ruta absoluta de las imágenes</i>	
grid::titulo()	10
<i>Método para asignar el CAPTION de la tabla</i>	
grid::unidadMedida()	11
<i>Método para asignar una cadena con el nombre de la unidad de medida del campo</i>	
grid::trigger()	11
<i>Método para determinar el campo que posee un trigger => no se mostrará el campo para insertar</i>	
grid::total()	10
<i>Método para determinar si se mostrará un pie de tabla con el total de campo</i>	
grid::tituloCampo()	10
<i>Método para asignar los títulos de todos los campo de la tabla</i>	
grid::encabezados()	7
<i>Método para asignar los títulos del grid, pero solamente a los campos visibles</i>	
grid::debuger()	7
<i>Método para comenzar el modo de debug para corrección de errores que mostrará los resultados de los queries</i>	
grid::botonInsertar()	4
<i>Método para determinar si se muestra el botón de insertar registro</i>	
grid::botonEliminar()	4
<i>Método para determinar si se muestra el botón de eliminar registro</i>	
grid::autofiltro()	4
<i>Método para fijar el campo que tendrán autofiltro para filtrar la grilla. Hay que llamar una vez a la funcion por cada campo.</i>	
grid::ancho()	3
<i>Método para asignar el ancho porcentual de la tabla</i>	

grid::botonModificar()	5
<i>Método para determinar si se muestra el botón de modificar registro</i>	
grid::campoFuncion()	5
<i>Método para asociar un función a un campo para llamara cada vez que se muestre el valor del campo</i>	
grid::camposSoloLectura()	7
<i>Método para configurar si un campo es de sólo lectura</i>	
grid::camposNoVisibles()	6
<i>Método para asignar los campos que no serán visibles en el grid</i>	
grid::campoOrden()	6
<i>Método para indicar el campo por el que se ordenará el grid si no se eligió en el encabezado</i>	
grid::campoLookUp()	5
<i>Método para crear un LookUpComboBox (SELECT) en los campos que son Foreign Key</i>	
grid	1
<i>GRID Class Copyright(C) 2005-2008 Isaluso</i>	