MIOLO2

Guia de Consulta Rápida

Miolo

_REQUEST($vars, $from = 'ALL')
Obtém o valor de um variável, consultando $_REQUEST, $_SESSION e $_GLOBAL. $var é uma string com o nome da variável, ou um array de strings.

assert($cond, $msg = '', $goto = '')
Avalia a condição $cond; se for true, interrompe a execução apresentando o prompt com $msg.

checkAccess($trans, $access, $deny = false)
Avalia se o usuário logado tem a permissão de acesso $access na transação $trans. $deny indica se o processamento será interrompido ou não, caso não haja direito de acesso.

checkLogin()
Indica se o usuário está logado ou não.

confirmation($msg, $gotoOK = '', $gotoCancel = '', $eventOk = '', $eventCancel = '', $halt = true)
Exibe um prompt de confirmação, com a mensagem $msg. $gotoOK e $gotoCancel são urls a serem executadas de acordo com a escolha do usuário. $eventOk e $eventCancel são nomes de métodos a serem executados. $halt indica se o processamento deve ser interrompido ou não.

error($msg = '', $goto = '', $caption = '', $event = '', $halt = true)
Exibe um prompt de erro, com a mensagem $msg. $goto é a url a ser executada. $caption é o título da janela de erro. $event é o evento a ser executado. $halt indica se o processamento deve ser interrompido.

getAbsolutePath($rel = NULL)
Obtém o path absoluto para o arquivo $rel, a partir do diretório de instalação do MIOLO.

getAbsoluteURL($rel, $module = NULL)
Obtém a URL completa para o arquivo $rel (no modulo $module).

getActionURL($module = '', $action = 'NONE', $item = '', $args = NULL, $dispatch = NULL, $scramble = false)
Obtém uma URL completa para execução do handler $action no módulo $module. $item é um parâmetro a ser passado para o handler. $args é um array de parâmetros opcionais. $dispatch indica o dispatcher a ser usado (por default index.php) e $scramble indica se a URL será criptografada.

getAuth()
Obtém uma referência para o objeto MAuth.

getBusiness($module, $name = 'main', $data = NULL)
Inclui a definição da classe $name, do módulo $module. $data é um parâmetro a ser passado para o construtor da classe.

getBusinessMAD($name = 'main', $data = NULL)
Inclui a definição da classe $name, do módulo MAD (o módulo de administração do Miolo, conforme definido em miolo.conf). $data é um parâmetro a ser passado para o construtor da classe..

getConf($key)
Obtém o valor da chave $key conforme definido no arquivo de configuração do Miolo (miolo.conf) ou do módulo (module.conf).

getContext($url = '', $style = 0, $scramble = false)
Obtém uma referência para o objeto MContext.

getCurrentAction()
Obtém o parâmetro “action” da URL atualmente sendo executada.

getCurrentModule()
Obtém o parâmetro “module” da URL atualmente sendo executada.

getCurrentURL()
Obtém a URL atualmente sendo executada.

getDatabase($conf = NULL, $user = NULL, $pass = NULL)
Obtém uma referência para um objeto Mdatabase, que representa uma conexão com o banco de dados, cujas definições estão no arquivo de configuração na chave $conf.

getLogin()
Obtém uma referência para o objeto MLogin.

getModulePath($module, $file)
Obtém o path absoluto para o arquivo $file do módulo $module.

getPage()
Obtém uma referência para o objeto MPage.

getPerms()
Obtém uma referência para o objeto MPerms.

getSession()
Obtém uma referência para o objeto MSession.

getSysTime()
Obtém o timestamp atual no formato 'd/m/Y H:i:s'.

getUI()
Obtém uma referência para o objeto MUI.

import($namespace, $class = '')
Cria uma referência para uma classe que poderá ser incluída posteriormente. Indica que a classe será acessada.

information($msg, $goto = '', $event = '', $halt = true)
Exibe um prompt de informação, com a mensagem $msg. $goto é a url a ser executada. $event é o evento a ser executado. $halt indica se o processamento deve ser interrompido.

invokeHandler($module, $action)
Executa o handler indicado por $action, no módulo $module.

logError($error, $conf = 'miolo')
Grava a mensagem $error no arquivo de log de erros.

logMessage($msg)
Grava a mensagem $msg no arquivo de log de mensagens.

logSQL($sql, $force = false, $conf = '?')
Grava o comando $sql no arquivo de log de SQL.

prompt($prompt, $halt = true)
Exibe um prompt indicado pelo objeto Mprompt. $halt indica se o processamento deve ser interrompido.

question($msg, $gotoYes = '', $gotoNo = '', $eventYes = '', $eventNo = '', $halt = true)
Exibe um prompt de interrogação, com a mensagem $msg. $gotoYes e $gotoNo são urls a serem executadas de acordo com a escolha do usuário.$eventYes e $eventNo são eventos. $halt indica se o processamento deve ser interrompido..

setConf($key, $value)
Modifica o valor de uma chave do arquivo de configuração (miolo.conf ou module.conf).

setLog($logname)
Define o nome do arquivo de log.

trace($msg, $file = false, $line = 0)
Envia uma mensagem $msg para o mecanismo de trace, definido em miolo.conf.

traceDump()
Executa um dump das informações de trace.

uses($name, $module = NULL)
Inclui o arquivo indicado por $name. $module pode ser usado se o arquivo estiver dentro de um módulo.

usesBusiness($module, $name = 'main')
Inclui o arquivo de classe de negócio indicado por $name do módulo $module.

Business

MBusiness

Os objetos Mbusiness, que representam os objetos da camada de domínio da aplicação (objetos de negócio), são instanciados através da chamada MIOLO->getBusiness($module, $name, $data).

beginTransaction()
Inicia o estado de transação.

checkError()
Retorna true se houver registro de erros associado ao objeto.

endTransaction()
Finaliza uma transação, com commit ou rollback, dependendo se houver erros associados ao objeto ou não.

execute($sql, $parameters = NULL)
Executa o comando DML representado pelo objeto $sql. $parameters são os parâmetros necessários à execução.

executeBatch($cmds)
Executa um conjunto de comandos (script). $cmds é um array de objetos MSQL com os comandos a serem executados. Executa os comandos dentro de uma transação.

getBusiness($module, $name = 'main', $data = NULL)
Retorna uma instância de um objeto de domínio.

getById($data=NULL)
Implementação default do método getById que deve existir em todos os objetos Mbusiness.

getData()
Implementação default do método getData, que retorna um objeto cujos atributos são os atributos do objeto de domínio.

getDatabase($database = NULL)
Define uma conexão com o banco de dados indicado por $database (a definição deve existir nos arquivos de configuração).

getDb()
Retorna o objeto Mdatabase em uso pelo objeto.

getErrors()
Retorna um array com os erros associados ao objeto.

getTransaction()
Retorna o objeto Mtransaction associado ao objeto, se houver algum.

log($operacao, $descricao)
Registra uma mensagem no arquivo de log do MIOLO. $operação e $ descrição são strings dependentes da aplicação.

query($sql, $parameters = NULL, $maxrows = 0)
Retorna o objeto Mquery resultante da executação do comando $sql.

setData($data = NULL)
Implementação default do método setData, que permite transferir (copiar) os valores dos atributos do objeto $data para os atributos do objeto Mbusiness.

setTransaction(MTransaction $transaction)
Define o objeto Mtransaction como $transaction. É usado quando o objeto deve participar de uma transação que já está em andamento.

Database

MDatabase

Representa uma conexão com o banco de dados, é instanciado através da chamada MIOLO->getDatabase($database), onde $database é a string de definição dos parâmetros de conexão estabelecidos nos arquivos de configuração. Os comandos SQL são representados por objetos MSQL.

count($sql)
Retorna o número de registros a partir do comando $sql.

execute($sql)
Executa o comando DML $sql.

executeBatch($sql_array)
Executa a sequencia de comandos DML definida pelo array $sql_array. Cada elemento do array é um objeto MSQL.

getErrors()
Retorna a lista de erros associados à conexão, se houver algum.

getNewId($sequence = 'admin', $tableGenerator = 'miolo_sequence')
Retorna o próximo valor da seqüência $sequence. $tablegenerator é usado em SGBD que não possuem comandos para seqüências, sendo estas definidas a partir de tabelas específicas criadas com esta finalidade.

getQuery($sql, $maxrows = 0)
Executa o comando $sql e retorna o objeto Mquery associado à consulta.

MQuery

Os objetos Mquery representam os resultados de uma consulta feito ao banco de dados. Estes dados estão representados em um array bidimensional, chamado resultset, onde as linhas representam os registros e as colunas representam os campos.

result
Array bidimensional que representa o resultset.

chunkResult($key = 0, $value = 1, $showKeyValue = false)
Condensa as várias colunas de um dataset em um único array associativo, onde o índice do array é definido pelos valores da coluna $key, e o valor de cada elemento do array é definido pela coluna $value. $showKeyValue indica se o valor do índice deve ser concatenado na valor final.

chunkResultMany($key, $values, $type = 'S', $separator = '')
Condensa as várias colunas de um dataset em um único array associativo, onde o índice do array é definido pelos valores da coluna $key, e o valor de cada elemento do array é definido pela concatenação dos valores das colunas $values. $type indica como será a concatenação: 'S' - os valores são concatenados com strings; 'A' – os valores são colocados em um array. $separator é usado no caso de concatenação de strings, para indicar o caracter separador.

eof()
Retorna true se é feita uma tentativa de acesso a um registro posterior ao último.

fields($fieldName)
Retorna o valor do campo $fieldname no registro corrente.

getColumnCount()
Retorna a quantidade de colunas do dataset.

getColumnName($colNumber)
Retorna o nome da coluna na posição $colNumber.

getColumnNames()
Retorna um array com os nomes de todas as colunas.

getColumnNumber($colName)
Retorna o número da coluna que tem o nome $colName.

getCSV($filename = '')
Gera um arquivo $filename com a representaçao CSV do resultset.

getFieldValues()
Retorna um array associativo com os valores dos campos no registro corrente. Os índices são os nomes dos campos.

getRowCount()
Retorna o número de linhas no dataset.

getRowObject()
Obtém um objeto cujos atributos têm como nome os nomes das colunas do dataset e com valores correspondentes aos campos no registro corrente.

getRowValues()
Retorna um array com os valores dos campos no registro corrente. Os índices são as posições de cada coluna (iniciando em 0).

getValue($colName)
Retorna o valor do campo $colName no registro corrente.

treeResult($group, $node)
Cria uma estrutura de árvore baseada em arrays com os valores do dataset. $group é uma string com a lista de campos que serão agrupados (ex. '1,2') e $node é uma lista de campos que serão concatenados em cada nó da árvore (ex. '3,4').

MSQL

Os objetos MSQL representam um comando SQL a ser executado.

__construct($columns = '', $tables = '', $where = '', $orderBy = '', $groupBy = '', $having = '', $forUpdate = false)
Instancia um objeto MSQL. $columns é a lista de colunas, $tables é a lista de tabelas, $where é a condição de pesquisa da cláula WHERE, $orderBy é o conteúdo da cláusula ORDER BY, $having é a condição da cláusula HAVING e $forUpdate indica se o comando SELECT terá a cláusula FOR UPDATE (usada em locks de registros).

setParameters($arg, …)
Define os valores para os parâmetros. O argumento para o método pode ser um único valor ou uma lista de valores.

setRange($arg, …)
Associa um objeto MqueryRange ao comando. O argumento para o método pode ser um objeto MqueryRange já criado, ou dois parâmetros (número da página a ser acessada e a quantidade de linhas por página) que serão usados para criar um objeto MqueryRange.

Forms

MCompoundForm

_form = array()
Array com os formulários a serem exibidos na área 'form' do formulário.

_info = array()
Array com os controles a serem exibidos na área 'info' do formulário.

_panel = array()
Array com os controles MactionPanel a serem exibidos na área 'panel' do formulário.

MForm

__construct($title='',$action='',$close='',$icon='')
Constructor. $title é o título da janela do formulário, $action é a url a ser chamada no post, $close é a url para o ícone de fechar do formulário, $icon é a url da imagem para o ícone da janela.

addButton($btn)
Insere um controle Mbutton no formulário.

addError($err)
Insere uma mensagem na lista de erros.

addField($field,$hint=false)
Insere o controle $field na lista de campos do formulário.

addFields($fields)
Insere os campos do array $fields na lista de campos do formulário.

addInfo($info)
Insere a string $info na lista de informações.

addJsCode($jscode)
Insere o código javascript na página do formulário.

addValidator($validator)
Insere um controle Mvalidator no formulário.

clearButtons()
Remove todos os botões do formulário.

clearFields()
Remove todos os campos do formulário.

createFields()
Método a ser sobreposto pelos formulários, com o objetivo de delimitar a criação de campos.

getButton($name)
Obtém o objeto Mbutton com nome $name.

getData()
Obtém um objeto com propriedades cujos nomes são os nomes dos campos do formulário e com os valores que foram submetidos em cada campo.

getField($name)
Obtem o controle cujo com nome $name.

getFieldAttr($name,$attr, $index=NULL)
Obtém o valor do atributo $attr do campo $name. $index é usado se o controle for indexado.

getFieldList()
Obtém um array de controles, com os campos do formulário.

getFieldValue($name,$value=false)
Obtém o valor do campo $name.

getFormValue($name,$value=NULL)
Obtém o valor postado no controle $name.

getPage()
Obtém uma refer~encia para o objeto MPage.

hasErrors()
Retorna true se o formulário tem erros (inseridos por addError).

hasInfos()
Retorna true se o formulário tem informações (inseridas por addInfo).

isSubmitted()
Indica se o formulário foi submetido (postado) ou se está sendo chamado pela primeira vez.

onSubmit($jscode)
Define um código javascript para o evento onSubmit da página.

setAction($action)
Define uma url para a ação do formulário.

setButtonAttr($name,$attr,$value)
Define o valor $value para o atributo $attr do botão $name.

setButtonLabel($index, $label)

Define o label para o botão que está no índice $index da lista de botões.

setButtons($btn)
Define os botões do formulário através do array $btn.

setClose($action)
Define a url a ser executada se o formulário for fechado.

setData($data)
Transefere os valores dos atributos do objeto $data para os campos do formulário que tiverem o mesmo nome.

setFieldAttr($name,$attr,$value)
Define o valor $value para o atributo $attr do campo $name.

setFields(&$fields)
Define os campos do formulário através do array $fields.

setFieldValidator($name,$value)
Define o validator $value para o campo $name.

setFieldValue($name,$value)
Atribui o valor $value ao campo $name.

setFormValue($name,$value)
Atribui o valor $value ao campo $name.

setIcon($icon)
Define a url para a imagem do ícone.

setLabelWidth($width)
Define a largura (em pixels ou percentual) para os labels dos campos.

setShowHints($state)
Define se os hints serão exibidos ou não.

setTitle($title)
Define o título do formulário. Para remover a barra de título usa-se $title=NULL.

setValidators($validators)
Define os validators do formulário com base no array $validators.

validate($required, $assert=true)
Valida os dados do formulário com base nos nomes de campos no array $required. Retorna true se todas os campos foram preenchidos e false se algum campo está vazio.

validateAll($assert=true)
Verifica se todos os campos do formulário estão preenchidos.

Grids

MGrid

O controle Mgrid permite a exibição, navegação (paginação), filtragem, seleção e ordenação de dados de um array bidimensional.

selecteds = array()
Array com os índices de todas as linhas que tenham o controle de seleção marcado, na página atual.

allSelecteds = array
Array com os índices de todas as linhas que tenham o controle de seleção marcado, em todas as páginas (allSelecteds[page] = array de índices).

__construct($data, $columns, $href, $pageLength = 15, $index = 0, $name = '', $useSelecteds = true)
Instancia um objeto Mgrid. $data é um array bidimensional com os dados a serem exibidos, $columns é um array com os objetos MgridColumn definindo as colunas, $href é a url onde o grid está sendo exibido, $pageLength define a quantidade de linha por página (0 exibe todas as linhas), $name é o identificador do controle (necessário quando existem dois grids no mesmo formulário), $useSelecteds define se a seleção de linhas será mantida em sessão.

addActionDelete($href)
Adiciona uma ação de remoção para a coluna de ações. $href é a url a ser chamada.

addActionIcon($alt, $icon, $href)
Adiciona uma ação na coluna de ações. $alt é o hint para o ícone, $icon é o nome do arquivo de imagem, $href é a url a ser chamada.

addActionSelect()
Adiciona uma coluna com checkbox para seleção de linhas,

addActionText($alt, $text, $href)
Adiciona uma ação na coluna de ações. $alt é um hint para o texto, $text é o texto a ser exibido no link, $href é a url a ser chamada.

addActionUpdate($href)
Adiciona uma ação de edição para a coluna de ações. $href é a url a ser chamada.

addError($err)
Adiciona a mensagem $err no array de erros do grid.

addFilterControl($index, $control, $type = 'text')
Adiciona um controle (MtextField ou Mselection) na área de filtros do grid. $index é a posição do controle (0-based), $control é o objeto, $type é o tipo de controle ('text' ou 'selection').

addFilterSelection($index, $label, $options, $value = '')
Adiciona um controle Mselection na área de filtros do grid. $index é a posição do controle (0-based), $label é o caption, $options é o array de opções, $value é o valor default para o filtro.

addFilterText($index, $label, $value = '')
Adiciona um controle MTextField na área de filtros do grid. $index é a posição do controle (0-based), $label é o caption, $value é o valor default para o filtro.

generateEmptyMsg()
Gera um controle Mdiv para a indicar que não existem dados a serem exibidos.

generateFilter()
Gera um controle Mdiv para a área de filtros do grid.

generateFooter()
Gera um controle Mdiv para a área de rodapé do grid.

generateHeader()
Gera um controle Mdiv para a área de cabeçalho do grid.

generateNavigationFooter()
Gera um controle Mdiv para a área de navegação no rodapé do grid.

generateNavigationHeader()
Gera um controle Mdiv para a área de navegação no cabeçalho do grid.

getData()
Retorna o array de dados.

getDataValue($row, $col)
Retorna o valor do array na linha $row, coluna $col.

getFilter()
Retorna true/false indicando se a área de filtros está sendo exibida ou não.

getFilterControl($index)
Retorna o controle de filtro na posição $index.

getFiltered()
Retorna true/false indicando se os dados exibidos estão filtrados ou não.

getFilterValue($index)
Retorna o valor do controle de filtro na posição $index.

getPageLength()
Retorna o número de linhas máximo a ser exibido.

getPageNumber()
Retorna o número da página que está sendo exibida.

hasErrors()
Indica se há erros associados ao grid.

setClose($action)
Define a url a ser chamada quando o grid for fechado.

setColumnAttr($col, $attr, $value)
Permite definir ou alterar um atributo de uma coluna. $col é a posição da coluna no array de colunas, $attr é o nome do atributo e $value o valor a ser atribuído.

setColumns($columns)
Define as colunas do grid. $columns é um array de objetos MgridColumn.

setControls($controls)
Define os controles a serem exibidos na área de controles. $controls é um array de controles.

setData($data)
Define o array de dados.

setFilter($status)
Define se a área de filtro deve ser exibida ou não.

setFooter($footer)
Define o conteúdo da área de rodapé.

setIndex($index)
Define a posição da coluna que servirá de índice para o grid. Usado com o padrão '$id' na url das ações.

setLinkType($linktype)
Define o comportamento dos links do formulário ('hyperlink' – método GET; 'linkbutton' – método POST).

setPageLength($pageLength)
Define o número máximo de linhas por página. Para exibir todas as linhas, $pageLength deve ser definido em zero.

setRowMethod($class, $method)
Define um método a ser chamado antes de renderizar cada linha do grid, permitindo fazer ajustes na exibição. $class define a classe e $method é o nome do método.

setTitle($title)
Define o caption do grid.

setWidth($width)
Define a largura do grid.

MDataGrid

O controle MDataGrid permite a exibição, navegação (paginação), filtragem, seleção e ordenação dos dados resultantes da execução de uma query (resultset).

__construct($query, $columns, $href, $pageLength = 15, $index = 0, $name = '', $useSelecteds = true)
Instancia um objeto MDataGrid. $query é um objeto Mquery, representando uma consulta que foi executada no banco de dados, $columns é um array com os objetos MDataGridColumn definindo as colunas, $href é a url onde o grid está sendo exibido, $pageLength define a quantidade de linha por página (0 exibe todas as linhas), $index define a coluna que será usada como índice do grid, $name é o identificador do controle (necessário quando existem dois grids no mesmo formulário), $useSelecteds define se a seleção de linhas será mantida em sessão.

MObjectGrid

O controle MObjectGrid permite a exibição, navegação (paginação), filtragem, seleção e ordenação dos dados provenientes de atributos de objetos que estão agrupados em um array.

__construct($array, $columns, $href, $pagelength = 15, $index = 0)

Instancia um objeto MObjectGrid. $array é um array com os objetos que serão exibidos, $columns é um array com os objetos MObjectGridColumn definindo as colunas, $href é a url onde o grid está sendo exibido, $pageLength define a quantidade de linhas por página (0 exibe todas as linhas), $index define a coluna que será usada como índice do grid.

Colunas

MDataGridColumn($field, $title = '', $align = 'left', $nowrap = false, $width = 0, $visible = true, $options = null, $order = false, $filter = false)

Instancia uma coluna do datagrid. $field é o nome do campo relativo à query, $title é o título da coluna, $align define o alinhamento ('left','center' ou 'right'), $nowrap define se o conteúdo da célula pode ser quebrado, $width define a largura da coluna, $visible define se a coluna será exibida ou não, $options define um array em que o valor da coluna é usado como índice, $order define se o grid pode ser ordenado por esta coluna, $filter define se o grid pode ser filtrado por esta coluna.

MDataGridControlt($control, $field, $title = '', $alinhamento = null, $nowrap = false, $width = 0, $visible = true)

Instancia uma coluna do datagrid, cujos valores serão exibidos como controles. $control é o controle que serve de base para a exibição; em cada linha, o valor do controle recebe o valor da coluna naquela linha. $field é o nome do campo relativo à query, $title é o título da coluna, $align define o alinhamento ('left','center' ou 'right'), $nowrap define se o conteúdo da célula pode ser quebrado, $width define a largura da coluna, $visible define se a coluna será exibida ou não.

MGridHyperlink($field, $title = '', $href, $width = 0, $visible = true, $options = null, $order = false, $filter = false)

Instancia uma coluna do datagrid, cujos valores serão exibidos como links. $field é o nome do campo relativo à query, $title é o título da coluna, $href é a url base para o link das céludas (o padrão #?# é substituído pelo valor da coluna), $width define a largura da coluna, $visible define se a coluna será exibida ou não, $options define um array em que o valor da coluna é usado como índice, $order define se o grid pode ser ordenado por esta coluna, $filter define se o grid pode ser filtrado por esta coluna.

Persistência

Criteria

retrieveAsQuery($parameters=null)

Retorna um objeto MQuery como resultado da consulta.

retrieveAsCursor($parameters=null)

Retorna um objeto Cursor (um array de objetos) como resultado da consulta.

Métodos para definição do critério

Para melhor compreensão do processo de definição de um critério, os seguintes conceitos devem ser compreendidos:

- classe base: a classe usada como base para a consulta
- attributo: é um atributo/propriedade de uma classe persistente. Pode ser definido simplesmente através do nome do atributo, ou através do caminho (path) do atributo através das associações da classe
- operando: é um valor usado em um critério. Pode ser um atributo, uma constante, uma função ou um subcriteria (um objeto RetrieveCriteria)
- operador: define qual a operação usada no critério. As seguintes operações estão definidas: =, <>, >, <, >=, <=, LIKE, IN
- associação: é o nome de uma associação definida para a classe base do critério. O nome da associação é definido no atributo <target> do mapeamento da associação.
- alias: nome usado como sinônimo de uma tabela do esquema relacional.

addColumnAttribute($attribute, $alias='')

Acrescenta um campo a ser recuperado na consulta. Se não for definido nenhum campo desta forma, todos os atributos da classe base serão retornados. O parâmetro $alias permite definir um alias para a coluna, no comando SQL a ser gerado.

addCriteria($op1, $operator, $op2)

Acrescenta uma operação a ser executada no critério. $op1 e $op2 são operandos, e $operator define a operação. Se outras operações já tiverem sido definidas, é usado o operador AND.

addGroupAttribute($attribute)

Define o atributo a ser usado na cláusula GROUP BY.

addHavingCriteria($op1, $operator, $op2)

Define a operação a ser usada na cláusula HAVING. Se outras operações já tiverem sido definidas, é usado o operador AND.

addJoinCriteria($criteria)

Define uma operação de JOIN entre duas classes. A operação de join é realizada definindo-se dois critérios e associando um com o outro através deste método.

addOrCriteria($op1, $operator, $op2)

Acrescenta uma operação a ser executada no critério. $op1 e $op2 são operandos, e $operator define a operação. Se outras operações já tiverem sido definidas, é usado o operador OR.

addOrderAttribute($attribute, $ascend=true)

Define o atributo a ser usado na cláusula ORDER BY, e a direção da ordenação (ascendente ou descendente).

addOrHavingCriteria($op1, $operator, $op2)

Define a operação a ser usada na cláusula HAVING. Se outras operações já tiverem sido definidas, é usado o operador OR.

getCriteria($op1, $operator, $op2)

Retorna um objeto BaseCriteria, que pode ser anexado a um objeto CriteriaCondition, para criação de filtros compostos.

setAlias($alias, $classMap=NULL)

Define um alias para a classe definida por $classMap. Se $classMap=NULL, o alias é definido para a classe base.

setAssociationAlias($associationName,$alias)

Define um alias para a associação $associationName. Este alias pode ser usado para referenciar atributos de outras classes.

setAssociationType($associationName,$joinType)

Define o tipo de join a ser feito em relação à associação. $joinType pode ser 'inner','left','right'. O default, quando este método não é usado, é fazer um INNER JOIN com a associação.

setAutoAssociationAlias($alias0, $alias1)

Define os aliases a serem usados quando é necessário fazer um auto-relacionamento da classe.

setDistinct($distinct=false)

Define a flag DISTINCT da query.

setReferenceAlias($alias)

Define o alias a ser usado para que atributos em subqueries possam referenciar uma classe usada na query externa.

PersistentObject

Uma vez que a classe MBusiness herda da classe PersistentObject, todos os objetos de negócio são automaticamente persistentes, ou seja, podem usar os seguintes métodos de persistência.

isPersistent()

Retorna true/false dependendo se o objeto é persistente ou não. Um objeto é considerado persistente os valores de seus atributos foram obtidos do banco de dados através da camada de persistência. Os métodos de recuperação (retrieve) de objetos definem este valor como true.

setPersistent($value)

Força o status de um objeto como sendo persistente ou não. Um objeto persistente é salvo no banco de dados com um SQL UPDATE e um objeto não-persistente com um SQL INSERT.

isProxy()

Retorna true/false dependendo se o objeto é do tipo "proxy" ou não. Em um objeto do tipo proxy, apenas os atributos com proxy=true foram recuperados do banco de dados, ou seja, um objeto do tipo proxy não possui todos os seus atributos preenchidos.

setProxy($value)

Força o status de um objeto como proxy ou não.

retrieve()

Preenche os atributos do objeto, acessando o banco de dados e recuperando o(s) registro(s) necessário(s) com base no atributo chave do objeto. O atributo chave do objeto deve ser preenchido antes da execução do método. As associações podem ser automaticamente recuperadas, dependendo da configuração feita no mapeamento.

retrieveFromQuery($query)

Preenche os atributos do objeto, com os dados do primeiro registro do resultset da query $query. As associações são automaticamente recuperadas, dependendo da configuração feita no mapeamento.

retrieveFromCriteria($criteria)

Preenche os atributos do objeto, com os dados do primeiro registro do resultset da query executada através de $criteria. As associações são automaticamente recuperadas, dependendo da configuração feita no mapeamento.

retrieveAssociation($target, $orderAttributes = null)

Recupera os dados referentes a uma associação $target e preenche o atributo correspondente com o objeto (ou com um array de objetos). Usado quando a recuperação da associação não é automática.

retrieveAssociationAsCursor($target, $orderAttributes = null)

Recupera os dados referentes a associação $target e preenche o atributo correspondente com um objeto Cursor.

retrieveAsProxy()

Preenche os atributos do objeto, acessando o banco de dados e recuperando o(s) registro(s) necessário(s) com base no atributo chave do objeto. O atributo chave do objeto deve ser preenchido antes da execução do método. Somente são preenchidos os atributos que tenham a opção proxy=true na sua definição.

getCriteria()

Retorna um objeto do tipo RetrieveCriteria, usado para executar consultas customizadas no banco de dados.

getDeleteCriteria()

Retorna um objeto do tipo DeleteCriteria, usado para executar um comando SQL DELETE no banco de dados.

save()

Armazena (persiste) um objeto no banco de dados. O armazenamento pode envolver uma ou mais inclusões ou atualizações, em uma ou mais tabelas do banco de dados. As associações podem ser automaticamente armazenadas, dependendo da configuração feita no mapeamento.

saveAssociation($target)

Persiste os dados referentes a associação $target, atualizando as tabelas e objetos envolvidos. Usado quando a persistência da associação não é automática.

delete()

Remove um objeto no banco de dados. A remoção pode envolver uma ou mais exclusões ou atualizações, em uma ou mais tabelas do banco de dados. As associações podem ser automaticamente removidas, dependendo da configuração feita no mapeamento.

deleteAssociation($target, $object)

Remove a associação $target do objeto $object com outro objeto. Isto implica na remoção da chave estrangeira nas relações 1:1 ou 1:N, ou do registro de ligação nas relações N:N.

Reports

MEzPDFReport

Classe base para reports usado ezPDF.

__construct($type = '2', $orientation = 'portrait')

Instancia um objeto MEzPDFReport.
$type : '1' (objeto pdf = MCpdf) ou '2' (objeto pdf = MCezPDF)
$orientation : 'portrait' ou 'landscape'

getOutput()

Obtém o texto pdf de saída.

getPdf()

Retorna o objeto pdf sendo usado no report.

newPage()

Começa uma nova página, posicionando o ponteiro no topo da nova página.

setFont($font)

$font : indica o arquivo de fonte a ser usada no report (ex. 'Helvetica.afm', 'tahoma.afm', 'Courier.afm')

setOutput($value = '')

$value : define o texto pdf de saída (ou o obtém do objeto pdf, se $value = '').

execute()

Gera o arquivo pdf, salva em disco e redireciona para exibição no browser.

MPDFReport

Classe para a criação de um report usando o layout de grid (tabela), usado principalmente para listagem de dados. Estende a classe MGrid.

__construct($data, $columns, $pageLength=1, $index=0)

Instancia um objeto MPDFReport.
$data : matriz (linha x coluna) com os dados a serem impressos.
$columns : array de objetos MPDFReportColumn, com as definições para cada coluna.
$pagelength : tamanho da página, em linhas.
$index : número da coluna que serve como índice dos dados.

addColumn($column)

Adiciona uma coluna ao report.
$column : objeto MPDFReportColumn.

clearPageBreak()

Inibe a quebra de página.

generateEmptyMsg()

Gera mensagem para quando não houver dados a serem impressos. Pode ser sobreposta por um report específico.

generateFooter()

Gera o rodapé do report. Pode ser sobreposta por um report específico. Por default chama GeneratePageFooter e GenerateReportFooter.

generateHeader()

Gera o cabeçalho do report. Pode ser sobreposta por um report específico. Por default chama GenerateReportReport e GeneratePageReport.

generatePageFooter()

Gera o rodapé de cada página do report. Pode ser sobreposta por um report específico.

generatePageHeader()

Gera o cabeçalho de cada página do report. Pode ser sobreposta por um report específico.

generatePageTitle()

Gera o título de cada página do report. Pode ser sobreposta por um report específico.

generateReportHeader()

Gera o cabeçalho do report. Pode ser sobreposta por um report específico.

getPDF()

Retorna o objeto MEzPDF que serve de base para o report.

initializeOptions()

Inicializa as opções do report (ver SetOption).

pageBreak()

Gera uma quebra de página.

setColumns($columns)

Define as colunas do report.
$columns : array de objetos MPDFReportColumn.

setOption($option, $value)

Define o valor $value para a opção $option. As principais opções são:

- 'showLines'=> 0,1,2, default is 0 (1->show the borders, 0->no borders, 2-> show borders AND lines
- between rows.)
- 'showHeadings' => 0 or 1, default is 1
- 'shaded'=> 0,1,2, default is 1 (1->alternate lines are shaded, 0->no shading, 2->both sets are shaded)
- 'shadeCol' => (r,g,b) array, defining the colour of the shading, default is (0.8,0.8,0.8)
- 'shadeCol2' => (r,g,b) array, defining the colour of the shading of the second set, default is (0.7,0.7,0.7), used when 'shaded' is set to 2.
- 'fontSize' => 10
- 'textCol' => (r,g,b) array, text colour, default is (0,0,0)
- 'titleFontSize' => 14
- 'rowGap' => 2 , the space between the text and the row lines on each row
- 'colGap' => 5 , the space between the text and the column lines in each column
- 'lineCol' => (r,g,b) array, defining the colour of the lines, default, black (0,0,0)
- 'xPos' => 'left','right','center','centre',or coordinate, reference coordinate in the x-direction, default is 'center'
- 'xOrientation' => 'left','right','center','centre', position of the table w.r.t 'xPos'. This entry is to be used in conjunction with 'xPos' to give control over the lateral position of the table. default is 'center'
- 'width' => <number>, the exact width of the table, the cell widths will be adjusted to give the table this width. default is 0.
- 'maxWidth' => <number>, the maximum width of the table, the cell widths will only be adjusted if the table width is going to be greater than this. Default is 596.

setPDF($pdf)

Define o objeto MEzPDF que será usado como base do report.

setTrigger($trigger, $class, $method, $param)

Define um trigger (um método executado automaticamente na ocorrência de um evento).
$trigger : 'BeforeNewPage' ou 'AfterNewPage'
$class : nome da classe
$method : nome do método
$param : array com parâmetros para o método $method

setWidth($width)

Define a largura (em percentagem) do report em relação à largura da página.

UI

MUI

alert($msg, $info, $href = '')

Exibe um prompt de erro, com a mensagem $msg. $info é o título da janela e $href é a url a ser executada na confirmação do usuário.

createForm($title = '')

Retorna uma instância de Mform, com o título $title.

getForm($module, $name, $data = NULL, $dir = NULL)

Retorna uma instância do formulário $name, do módulo $module. $data é passado como parâmetro para o construtor do formulário. $dir indica um subdiretório dentro do diretório de formulários.

getGrid($module, $name, $data = NULL, $dir = NULL)

Retorna uma instância do grid $name, do módulo $module. $data é passado como parâmetro para o construtor do grid. $dir indica um subdiretório dentro do diretório de grids.

getImage($module, $name)

Retorna uma URL para acesso a imagem $name, do módulo $module.

getImageSrc($name, $module = '')

Retorna um path absoluto para acesso a imagem $name, do módulo $module.

getImageTheme($theme, $name)

Retorna uma URL para acesso a imagem $name, do tema $theme.

getReport($module, $name, $data = NULL, $dir = NULL)

Retorna uma instância do reletório $name, do módulo $module. $data é passado como parâmetro para o construtor do relatório. $dir indica um subdiretório dentro do diretório de relatórios.

Validators

MCEPValidator

__construct($field,$label='',$type = 'optional',$msgerr='')

MCNPJValidator

__construct($field,$label='',$type = 'optional',$msgerr='')

MCompareValidator
__construct($field,$label='', $operator, $value, $datatype='s', $type = 'optional',$msgerr='')

MCPFValidator
__construct($field,$label='',$type = 'optional',$msgerr='')

MDATEDMYValidator
__construct($field,$label='',$type = 'optional',$msgerr='')

MDATEYMDValidator
__construct($field,$label='',$type = 'optional',$msgerr='')

MEmailValidator
__construct($field,$label='',$type = 'optional',$msgerr='')

MIntegerValidator
__construct($field,$label='', $type = 'optional',$msgerr='')

MMASKValidator
__construct($field,$label='',$mask,$type = 'ignore',$msgerr='')

MPasswordValidator
__construct($field,$label='',$type = 'required',$msgerr='')

MPHONEValidator
__construct($field,$label='',$type = 'optional',$msgerr='')

MRangeValidator
__construct($field,$label='', $min, $max, $datatype='s', $type = 'optional', $msgerr='')

MRegExpValidator
__construct($field,$label='', $regexp='', $type = 'optional', $msgerr='')

MRequiredValidator
__construct($field,$label='',$max=0, $msgerr='')

MTIMEValidator
__construct($field,$label='',$type = 'optional',$msgerr='')

Widgets (Controles)

MActionHyperLink
__construct($name, $label, $module = '', $action = '', $item = null, $args = null)

Instancia um objeto MActionHyperlink, cujo objetivo é renderizar um hyperlink para um handler. $name é o identificador do objeto, $label é o caption para o objeto. $module, $action,$item e $args são usados para construir a url no padrão do Miolo.

MActionPanel
__construct($name = '', $caption = '', $controls = NULL, $close = '', $icon = '', $iconType='large')

Instancia um objeto MActionPanel, cujo objetivo é renderizar um painel com links para ações. $name é o identificador do controle,, $caption é o título do painel, $controls é um array com os objetos do painel (geralmente ícones com links), $close é a url executada quando o painel é fechado, $icon é a url da imagem do ícone, $iconType indica o tamanho dos ícones (‘large’ ou ‘small’).

addAction($label, $image, $module = 'main', $action = '', $item = NULL, $args = NULL)

Adiciona um controle MimageLinkLabel ao panel. $label é o texto do link, $image é a url da imagem, $module, $action, $item e $args são usados para construir a url de ação.

addBreak()

Adiciona uma quebra na seqüência dos ícones.

addUserAction($transaction, $access, $label, $image, $module = 'main', $action = '', $item = '', $args = NULL)

Adiciona um controle MimageLinkLabel ao panel, se o usuário tiver permissão de acesso à ação. $transaction e $access são checados com MIOLO::checkAccess(). $label é o texto do link, $image é a url da imagem, $module, $action, $item e $args são usados para construir a url de ação.

insertAction($pos, $label, $image, $module = 'main', $action = '', $item = NULL, $args = NULL)

Adiciona um controle MimageLinkLabel ao panel, na posição indicada por $pos. $label é o texto do link, $image é a url da imagem, $module, $action, $item e $args são usados para construir a url de ação.

insertUserAction($pos, $transaction, $access, $label, $image, $module = 'main', $action = '', $item = '', $args = NULL)

Adiciona um controle MimageLinkLabel ao panel, na posição indicada por $pos, se o usuário tiver permissão de acesso à ação. $transaction e $access são checados com MIOLO::checkAccess(). $label é o texto do link, $image é a url da imagem, $module, $action, $item e $args são usados para construir a url de ação.

setControlSize($width, $height)

Define o tamanho do controle a ser exibido. $width é a largura e $height é a altura (ambos em pixels – ex: ‘35px’).

setIconType($type = 'large')

Define o formato do ícone (‘large’ ou ‘small’).

MBaseGroup
__construct($name = '', $caption = '', $controls = '', $disposition = 'none', $border = 'css')

Instancia um objeto MbaseGroup. $name é o identificador do objeto, $controls é o array com os controles que estão agrupados, $disposition define como os controles serão exibidos(‘none’/’horizontal’ ou ‘vertical’) e $border indica como será renderizada a borda da caixa (‘none’ para nenhuma e ‘css’ para usar a definição das folhas de estilo).

setScrollHeight($height)

Define a altura da caixa para exibição dos controles, em pixels (ex. ‘40px’). Se necessário é exibida uma scrollbar.

setBorder($border)

Define a forma de renderização da borda (‘none’ ou ‘css’).

MButton
__construct($name = '', $label = '', $action = NULL, $image = NULL)

Instancia um objeto Mbutton. $name é o identificador do objeto, $label é o texto do botão, $action indica a ação a ser executada no clique do botão e $image indica a imagem a ser exibida no botão. $action pode ter os seguintes valores:

- 'SUBMIT' : submete o formulário com um POST
- 'RESET' : limpa os campos do formulário
- 'PRINT' : imprime a página atual
- 'PDF' : gera um arquivo pdf da página atual
- 'RETURN' : volta para a página anterior
- 'NONE' : nenhuma ação
- url : uma url completa a ser chamada
- javascript : um código javascript a ser executado no evento onClick

setImage($location)

Define a url da imagem a ser exibida.

setOnClick($onclick)

Define a ação para o evento onClick.

setAction($action)

Define a ação para o botão.

MButtonFind
__construct($space = NULL)

MButtonWindow
__construct($name = NULL, $label = NULL, $href = NULL, $target = '')

Instancia um objeto Mbutton para chamar uma URL. $name é o identificador do objeto, $label é o texto do botão, $href indica a URL a ser chamada no clique do botão e $target é o identificador da janela.

MCalendarField
__construct($name='', $value='', $label='', $size=20, $hint='', $type='calendar-win2k-1')

Instancia um objeto MCalendarField, para entrada de datas. $name é o identificador do objeto, $value é o valor inicial a ser atribuído, $label é o caption do campo, $size é o tamanho da caixa de entrada, $hint é um texto de ajuda e $type define as características visuais do calendário, através de arquivos CSS localizados em <miolo>/html/scripts/datepicker/css.

MCheckBox
__construct($name = '', $value = '', $label = '', $checked = false, $text = NULL, $hint = '')

Instancia um objeto MCheckBox. $name é o identificador do objeto, $value é o valor a ser enviado no POST, $label é o caption do controle, $checked é um booleano indicando se o controle está selecionado, $text é o texto a ser exibido no controle, $hint é o texto de ajuda.

setChecked($checked)

Define se o controle será marcado ou não (true/false).

MCheckBoxGroup
__construct($name = '', $label = '', $options = '', $hint = '', $disposition = 'horizontal', $border = 'none')

Instancia um objeto McheckBoxGroup (um grupo de controles checkbox). $name é o identificador do objeto, $options é o array com os controles que estão agrupados, $hint é um texto de ajuda, $disposition define como os controles serão exibidos(‘none’/’horizontal’ ou ‘vertical’) e $border indica como será renderizada a borda da caixa (‘none’ para nenhuma e ‘css’ para usar a definição das folhas de estilo). $options pode ser: um array de

controles McheckBox, um array de controles Moption, um array de pares label/valor ou simplesmente um array de valores.

MComboBox

__construct($name='',$value='',$label='',$options='',$showValues=false,$hint='',$size=6)

Instancia um objeto MComboBox, cujo objetivo é renderizar uma caixa de texto associada a uma caixa de seleção. Quando a seleção é feita, a caixa de texto é atualizada para o valor selecionado. $name é o identificador do objeto, $value é o valor do objeto, $label é o caption, $option é um array associativo com as opções a serem exibidas (o índice é a opção e o conteúdo é o valor), $showValues indica se os valores serão exibidos ao lado das opções, $hint é um pequeno texto de ajuda e $size indica quantas linhas serão exibidas simultaneamente no controle.

MContainer

__construct($name = NULL, $controls = NULL, $disposition = 'none')

Instancia um objeto Mcontainer, cujo objetivo é agrupar visualmente um conjunto de controles. $name é o identificador do objeto, $controls é o array de controles a serem agrupados e $disposition indica a disposição dos controles ('none'/'horizontal' ou 'vertical').

setClass($cssClass)

Define o seletor CSS a ser usado na renderização do container.

setDisposition($disposition)

Define a disposição dos controles ('none'/'horizontal' ou 'vertical').

setSpaceHeight($value)

Define o espaçamento em pixels entre os campos no caso de $disposition='vertical'.

MContent

__construct($module = false, $name = false)

Instancia um controle Mdiv cujo conteúdo é obtido do arquivo $name, no módulo $module.

MControl

Esta é a classe base para todos os controles (vísiveis ou não) usado na interface com o usuário. De forma geral, os controles são envolvidos por uma Box (uma tag div do HTML), que permite controlar suas características e posicionamento.

constantes

const FORM_MODE_WHOLE_ROW = 0;
O controle ocupa a linha inteira do form.
const FORM_MODE_SHOW_SIDE = 1;
Exibe o caption lado-a-lado.
const FORM_MODE_SHOW_ABOVE = 2;
Exibe o caption acima do controle.
const FORM_MODE_SHOW_NBSP = 3;
Exibe o caption com espaços ao lado do controle.

__clone()

Retorna um clone do controle.

__construct($name = NULL)

Construtor. $name é o nome do controle.

_AddStyle($name, $value)

Inclui um atributo na cláusula style do CSS.

addAttribute($name, $value = '')

Inclui um atributo HTML para o controle.

addBoxStyle($name, $value)

Inclui um atributo na cláusula style do CSS do box do controle.

addStyle($name, $value)

Atribui $value à propriedade $name.

addStyleFile($styleFile)

Indica uma folha de estilos a ser inserida na renderização.

attachEventHandler($name, $handler, $param = NULL)

Associa o método $handler ao evento $name.

attributes()

Retorna uma string com os atributos HTML.

eventHandler()

Executa o método associado a um evento do controle.

generate()

Retorna o código HTML relativo ao controle.

generateBox($content)

Renderiza o controle dentro de um Box e retorna o código HTML.

getAttributes()

Retorna uma string com os atributos HTML mais o atributo style.

getId()

Retorna a identificação (id) do controle.

getName()

Retorna o nome do controle.

getStyle()

Retorna o conteúdo do atributo style do controle.

setAttribute($name, $value)

Atribui o valor $value ao atribute HTML $name.

setAttributes($attr)

Define os atributos HTML do controle através da string $attr.

setBoxAttributes($attr)

Define os atributos HTML da box do controle através da string $attr.

setBoxClass($cssClass, $add = true)

Define o seletor CSS para a Box do controle; $add permite adicionar mais de um seletor.

setBoxId($id)

Define a identificação do Box do controle.

setCaption($caption)

Define o caption (label) do controle.

setClass($cssClass, $add = true)

Define o seletor CSS do controle.

setColor($value)

Define a cor do controle.

setEnabled($state)

Define se o controle está habilitado ou não.

setFont($value)

Define a fonte a ser usada.

setFormMode($mode)

Define o modo de renderização (ver constantes).

setHeight($value)

Define a altura do controle.

setId($id)

Define a identificação (id) do controle.

setJsHint($hint)

Define um hint javascript.

setName($name)

Define o nome do controle.

setPosition($left, $top, $position = 'absolute')

Define a posição do controle com base em atributos CSS.

setReadOnly($status)

Define se o controle será readonly ou não.

setStyle($style)

Define uma string para o atributo style.

setVisibility($value)

Define se o controle será visível ou não.

setWidth($value)

Define a largura do controle.

MCurrencyField

__construct($name='', $value='', $label='', $size=10, $hint='')

Instancia um objeto MCurrencyField, para entrada de valores monetários. $name é o identificador do objeto, $value é o valor inicial a ser atribuído, $label é o caption do campo, $size é o tamanho da caixa de entrada, $hint é um texto de ajuda.

MDiv

__construct($name = NULL, $content = ' ', $class = NULL, $attributes = NULL)

Instancia um objeto renderizado com a tag . $name é o identificador do objeto, $content é uma string ou array com o conteúdo do objeto, $class é o seletor CSS a ser aplicado e $attributes é uma string com os atributos HTML.

MFieldLabel

__construct($id, $text = NULL)

Instancia um objeto MfieldLabel, cujo objetivo é renderizar um label $text para o controle definido por $id, usando a tag <label>.

MFileContent

__construct($filename = null, $isSource = false)

Instancia um controle Mdiv cujo conteúdo é obtido do arquivo $filename. $isSource indica é o arquivo conteúdo um código-fonte PHP ou não.

MFileField

__construct($name='', $value='', $label='', $size=40, $hint='')

Instancia um objeto MFileField, para entrada de path de arquivos que serão enviados ao servidor. $name é o identificador do objeto, $value é o valor inicial a ser atribuído, $label é o caption do campo, $size é o tamanho da caixa de entrada, $hint é um texto de ajuda.

MHContainer

__construct($name = NULL, $controls = NULL)

Instancia um objeto MHcontainer, cujo objetivo é agrupar visualmente um conjunto de controles, na disposição horizontal. $name é o identificador do objeto e $controls é o array de controles a serem agrupados.

MHiddenField

__construct($name='', $value='')

Instancia um objeto MHiddenField, para definição de campos ocultos. $name é o identificador do objeto e $value é o valor inicial a ser atribuído.

MHr

__construct()

Instancia um objeto renderizado com a tag <hr>.

MImage

__construct($name = NULL, $label = NULL, $location = NULL, $attrs = NULL)

Instancia um objeto Mimage, cujo objetivo é exibir uma imagem. $name é o identificador do objeto, $location é a url para a imagem e $attrs é a string com os atributos HTML.

MImageButton

__construct($name = '', $label = '', $action = '', $location = '', $attrs = NULL)

Instancia um objeto MImageButton, cujo objetivo é renderizar uma imagem que, quando clicada, realiza um POST do formulário. $name é o identificador do objeto, $label é o caption para o objeto (não exibido), $action é a url a ser chamada, $location é a url da imagem e $attrs é uma string com os atributos HTML a serem aplicados na imagem.

MImageFormLabel

__construct($name = NULL, $label = NULL, $location = NULL, $attrs = NULL)

Instancia um objeto MimageFormLabel, cujo objetivo é exibir uma imagem. $name é o identificador do objeto, $location é a url para a imagem e $attrs é a string com os atributos HTML. $label é um texto exibido abaixo da imagem, de forma centralizada.

Input

MImageLink

__construct($name = '', $label = '', $action = '', $location = '', $attrs = NULL)

Instancia um objeto MImageLink, cujo objetivo é renderizar uma imagem com hyperlink. $name é o identificador do objeto, $label é o caption para o objeto (não exibido), $action é a url a ser chamada, $location é a url da imagem e $attrs é uma string com os atributos HTML a serem aplicados na imagem.

MImageLinkLabel

__construct($name = '', $label = '', $action = '', $location = '', $attrs = NULL)

Instancia um objeto MImageLinkLabel, cujo objetivo é renderizar uma imagem com hyperlink. $name é o identificador do objeto, $label é o caption para o objeto (exibido abaixo da imagem), $action é a url a ser chamada, $location é a url da imagem e $attrs é uma string com os atributos HTML a serem aplicados na imagem.

MInputButton

__construct($name = '', $label = '', $action = NULL, $image = NULL)

Instancia um objeto MinputButton. $name é o identificador do objeto, $label é o texto do botão, $action indica a ação a ser executada no clique do botão e $image indica a imagem a ser exibida no botão. O objetivo deste controle é permitir a customização do botão através de seletores CSS.

Choice

MInputGrid

__construct($name, $label = '', $rows = 0)

Instancia um objeto MinputGrid, que apresenta diversos campos de entrada organizados em linhas e colunas. $name é o identificador do objeto, $label é o caption e $rows o número de linhas a serem exibidas. As colunas devem ser adicionas através do método addColumn.

addColumn($label, $name, $colWidth = 0, $value = '')

Adiciona uma coluna. $label indica o título da coluna, $name o identificador do objeto MInputGridColum, $colWidth a largura da coluna e $value o valor inicial para os campos. Os campos são acessados com o formato array (<nome>[r][c], onde r é o número da linha – 0-based – e c é o número da coluna – 0-based).

getRowCount()

Obtém o número de linhas definidas para o controle.

setRowCount($rows)

Define o número de linhas para o controle.

MLabel

__construct($text = NULL, $color = '' , $bold=false)

Instancia um objeto MLabel, cujo objetivo é mostrar um texto na página. $text é o texto a ser exibido, e $color define a cor do texto.

MLink

__construct($name = NULL, $label = NULL, $href = NULL, $text = NULL, $target = '_self')

Instancia um objeto Mlink, cujo objetivo é renderizar um hyperlink. $name é o identificador do objeto, $label é o caption para o objeto, $href é a url a ser chamada, $text é o texto exibido no link (se for NULL é exibido o $label) e $target define em qual frame a página chamada será aberta.

setAction($module = '', $action = '', $item = null, $args = null)

Constrói a url como chamada a um handler, no padrão do Miolo.

setHREF($href)

Define a url a ser chamada.

setOnClick($onClick)

Define um código javascript a ser executado no evento onClick do link.

setText($text)

Define o texto a ser exibido no link.

MLinkButton

__construct($name = '', $label = '', $action = '')

Instancia um objeto MlinkButton, cujo objetivo é renderizar um hyperlink que, quando clicado, realiza um POST do formulário. $name é o identificador do objeto, $label é o caption para o objeto, $action é a url a ser chamada.

MLinkButtonGroup

__construct($name = '', $label = '', $options = '', $disposition = 'horizontal', $border = 'css')

Instancia um objeto MLinkButtonGroup (um grupo de controles linkbutton). $name é o identificador do objeto, $options é o array com os controles MLinkButton que estão agrupados, $disposition define como os controles serão exibidos('none'/'horizontal' ou 'vertical') e $border indica como será renderizada a borda da caixa ('none' para nenhuma e 'css' para usar a definição das folhas de estilo).

MMultiLineField

__construct($name='', $value='', $label='', $size=20, $rows=1, $cols=20, $hint='', $validator=null)

Instancia um objeto MMultiLineField, para entrada de dados com múltiplas linhas. $name é o identificador do objeto, $value é o valor inicial a ser atribuído, $label é o caption do campo, $size é o tamanho da caixa de entrada, $rows é o número de linhas a serem exibidas, $cols é o número de colunas em cada linha, $hint é um texto de ajuda e $validator é um objeto Mvalidator associado ao objeto.

MMultiSelection

__construct($name='', $values=Array('1','2','3'), $label=' ', $options=Array('Option1','Option2','Option3'), $showValues=false, $hint='', $size='3')

Instancia um objeto MMultiselection, cujo objetivo é renderizar uma caixa de seleção que permite múltiplas seleções. $name é o identificador do objeto, $value é um array com os valores selecionados, $label é o caption, $option é um array associativo com as opções a serem exibidas (o índice é a opção e o conteúdo é o valor), $showValues indica se os valores serão exibidos ao lado das opções, $hint é um pequeno texto de ajuda, $size indica quantas linhas serão exibidas simultaneamente no controle.

MMultiSelectionField

__construct($name = '', $value = null, $label = '', $fields = '', $width = 200, $buttons = false, $info = '', $hint = '')

Instancia um objeto MmultiSelectionField, cujo objetivo é permitir a entrada de dados através de múltiplas combinações de campos Mselection. $name é o identificador do objeto, $label é o caption, $fields é um array de objetos Mselection, $width é a largura em pixels do campo de entrada de dados, $buttons inddica de os botões de navegação serão exibidos, $info é um caption para o campo de entrada e $hint um pequeno texto de ajuda.

getCodeValue()

Obtém um array bidimensional com os valores do campo de entrada.

setCodeValue($value)

Define os valores do campo de entrada. $value é um array bidimensional com os valores a serem atribuídos ao campo.

MMultiTextField3

__construct($name = '', $value = null, $label = '', $fields = '', $width = 200, $buttons = false, $layout = 'vertical', $hint = '')

Instancia um objeto MmultiTextField3, cujo objetivo é permitir a entrada de dados através de múltiplas combinações de campos de diversos tipos. $name é o identificador do objeto, $label é o caption, $width é a largura em pixels do campo de entrada de dados, $buttons inddica de os botões de navegação serão exibidos, $layout indica a disposição dos campos ('horizontal', 'vertical' ou 'vertical2') e $hint um pequeno texto de ajuda. $fields é um array de controles, através dos quais serão obtidos os valores para formar o valor do objeto.

getCodeValue()

Obtém um array bidimensional com os valores do campo de entrada.

setCodeValue($value)

Define os valores do campo de entrada. $value é um array bidimensional com os valores a serem atribuídos ao campo.

MOpenWindow

__construct($name = NULL, $label = NULL, $href = NULL, $target = '')

Instancia um objeto Mlink para chamar uma URL. $name é o identificador do objeto, $label é o texto do botão, $href indica a URL a ser chamada no clique do botão e $target é o identificador da janela.

MOrderedList

__construct($name = '', $options = array())Instancia um objeto para renderização de uma lista ordenada em HTML. $options é uma array com o conteúdo de cada elemento da lista.

MPage

Esta classe representa a página HTML que será renderizada.

addJsCode($jscode)
Insere código javascript na página.

addScript($url)
Indica que será usado o script apontado por $url.

addStyle($url)
Indica que será usada a folha de estilos apontada por $url (relativa ao diretório do MIOLO).

addStyleCode($code)
Insere um código javascript diretamente na página.

addStyleURL($url)
Indica que será usada a folha de estilos apontada por $url (absoluta).

getJsCode()
Obtém o código javascript inserido na página.

isPostBack()
Retorna se a página foi submetida ou se é a primeira vez que é chamada.

onLoad($jscode)
Define o código a ser usado no evento onLoad da página.

onSubmit($jscode)
Define o código a ser usado no evento onSubmit da página.

redirect($url)
Redireciona a execução da página para a $url.

request($vars, $component_name = '', $from='ALL')
Obtém valores que foram submetidos à página.

setAction($action)
Define a ação para a tag FORM.

setStyles($value)
Usado para definir as folhas de estilos associadas à página.

setTitle($title)
Define o título para a janela do browser onde será exibida a página.

MPageComment

__construct($text = NULL)
Insere um comentário HTML na página.

MPanel

__construct($name = '', $caption = '', $controls = NULL, $close = '', $icon = '')
Instancia um objeto Mpanel, cujo objetivo é renderizar um painel com controles. $name é o identificador do controle,, $caption é o título do painel, $controls é um array com os objetos do painel (geralmente ícones com links), $close é a url executada quando o painel é fechado e $icon é a url da imagem do ícone.

addControl($control, $width = '', $float = 'left')
Adiciona um controle no painel. $control é o controle, $width define a largura do controle e $float define a posição como o controle será renderizado ('left' – o controle é colocado logo após o controle anterior; 'clear' – o controle é colocado abaixo do controle anterior)

insertControl($pos, $control, $width = '', $float = 'left')
Insere um controle no painel em uma posição específica. $pos é um número indicando a posição, $control é o controle, $width define a largura do controle e $float define a posição como o controle será renderizado ('left' – o controle é colocado logo após o controle anterior; 'clear' – o controle é colocado abaixo do controle anterior)

setTitle($title)
Define o título (caption) do painel.

MPasswordField

__construct($name='', $value='', $label='', $size=20, $hint='', $validator=null)
Instancia um objeto MPasswordField, para entrada de senhas. $name é o identificador do objeto, $value é o valor inicial a ser atribuído, $label é o caption do campo, $size é o tamanho da caixa de entrada, $hint é um texto de ajuda e $validator é um objeto Mvalidator associado ao objeto.

MRadioButton

Instancia um objeto MCheckBox. $name é o identificador do objeto, $value é o valor a ser enviado no POST, $label é o caption do controle, $checked é um booleano indicando se o controle está selecionado, $text é o texto a ser exibido no controle, $hint é o texto de ajuda.

setChecked($checked)
Define se o controle será marcado ou não (true/false).

MRadioButtonGroup

__construct($name = '', $label = '', $options = '', $default = false, $hint = '', $disposition = 'vertical', $border = 'none')
Instancia um objeto MRadioBoxGroup (um grupo de controles radio). $name é o identificador do objeto, $options é o array com os controles que estão agrupados, $default é o valor do controle que será marcado inicialmente, $hint é um texto de ajuda, $disposition define como os controles serão exibidos('none'/'horizontal' ou 'vertical') e $border indica como será renderizada a borda da caixa ('none' para nenhuma e 'css' para usar a definição das folhas de estilo).
$options pode ser: um array de controles MRadioButton, um array de controles Moption, um array de pares label/valor ou simplesmente um array de valores.
Image

MRawText

__construct($text = NULL)
Instancia um objeto MRawTextl, cujo objetivo é mostrar um texto na página, sem nenhuma formatação feita via CSS. $text é o texto a ser exibido.

MSelection

__construct($name='',$value='',$label='', $options=Array('Não','Sim'),$showValues=false,$hint='',$size='')
Instancia um objeto Mselection, cujo objetivo é renderizar uma caixa de seleção. $name é o identificador do objeto, $value é o valor do objeto, $label é o caption, $option é um array associativo com as opções a serem exibidas (o índice é a opção e o conteúdo é o valor), $showValues indica se os valores serão exibidos ao lado das opções e $hint é um pequeno texto de ajuda.

getOption($value)
Obtém a opção correspondente a determinado valor.

setAutoSubmit($state=false)
Define se o formulário será submetido após a seleção.

setCols($value)
Define a largura, em colunas, do controle.

setOption($option,$value)
Define uma opção para seleção. $option é o texto a ser exibido, $value é o valor a ser postado.

MSpacer

__construct($space = NULL)
Instancia objeto renderizado com a tag . $space indica o espaçamento em pixels (ex. '10px').

MSpan

__construct($name = NULL, $content = ' ', $class = NULL, $attributes = NULL)
Instancia um objeto renderizado com a tag . $name é o identificador do objeto, $content é uma string ou array com o conteúdo do objeto, $class é o seletor CSS a ser aplicado e $attributes é uma string com os atributos HTML.

MSeparator

__construct($text = NULL, $margin = '', $color = '')
Insere uma linha como separador na página. $text é um texto que pode vir acima da linha, $margin indica a margem em pixels e $color define a cor da linha.

MText

__construct($name = '', $text = NULL, $color = '')
Instancia um objeto MText, cujo objetivo é mostrar um texto na página. $name é o identificador do objeto, $text é o texto a ser exibido e $color define a cor do texto.

MTextHeader

__construct($name = '', $level = '1', $text = NULL, $color = '')
Instancia um objeto MtextHeader, cujo objetivo é renderizar um header HTML (tag <h1>, <h2>,...). $name é o identificador do objeto, $level indica o nível do header, $text é o texto a ser exibido, $color define a cor do texto.

MTextLabel

__construct($name = '', $text = null, $label = '', $color = '')
Instancia um objeto MTextLabel, cujo objetivo é mostrar um texto na página. $name é o identificador do objeto, $text é o texto a ser exibido, $label é um caption para o texto e $color define a cor do texto.
Link

MTextField

__construct($name='',$value='',$label='', $size=10, $hint='', $validator=NULL)
Instancia um objeto MtextField, para entrada de dados. $name é o identificador do objeto, $value é o valor inicial a ser atribuído, $label é o caption do campo, $size é o tamanho da caixa de entrada, $hint é um texto de ajuda e $validator é um objeto Mvalidator associado ao objeto.

setValidator($value)

Associa um objeto Mvalidator.

addMask($mask, $optional = true, $msg = '')

Associa uma máscara ao campo. $mask é uma máscara de edição com os seguintes caracteres especiais:
- # : aceita caracteres na faixa [0-9]
- a : aceita caracteres na faixa [a-z][0-9]
- A: aceita caracteres na faixa [A-Z][0-9]
- l : aceita caracteres na faixa [a-z]
- L : aceita caracteres na faixa [A-Z]

$optional indica se a máscara deve ser totalmente preenchida ou não; $msg é a mensagem a ser apresentada ao usuário no caso de entrada inválida.

MTreeMenu

__construct($name = '', $template = 0, $action = '', $target = '_blank')

Renderiza uma árvore de controles, usando javascript. $name é o nome do controle; $template indica as imagens a serem usadas, segundo os templates disponíveis em /html/scripts/tigra/icons<template>; $action é a url a ser chamada no evento clique em um elemento da árvore – nesta url o caracter '#' é substituído pelo número do elemento da árvore. $target indica se a URL será chamada na mesma janela ou em outra janela.

MUnOrderedList

__construct($name = '', $options = array())

Instancia um objeto para renderização de uma lista não-ordenada em HTML. $options é uma array com o conteúdo de cada elemento da lista.

MVContainer

__construct($name = NULL, $controls = NULL)

Instancia um objeto MVcontainer, cujo objetivo é agrupar visualmente um conjunto de controles, na disposição vertical. $name é o identificador do objeto e $controls é o array de controles a serem agrupados.