2025/11/28 21:27 1/7 Grid ad Albero (TreeGrid)

Grid ad Albero (TreeGrid)

Per impostare una grid ad albero come questa:



Occorre definire dei parametri alla tabella. Prima di tutto tra i metadati della tabella dovremo definire:

- treeGrid:true, : Che abilita il comportamento "ad albero" può assumere la tabella.
- treeGridModel:'adjacency', : Determina il metodo utilizzato per la TreeGrid
- **ExpandColumn :'NOMECOLONNA',** :Dove vogliamo che l'effetto ad albero abbia effetto. (Solitamente di fianco alla prima colonna della tabella)

All'interno della grid dovremo definirne una "INDICE" che dovrà avere le seguenti caratteristiche:

- Essere ,ovviamente, un td→generico.
- Avere un nome univoco, come ad esempio "INDICE"
- Tra gli attributi:
 - Class:
 - **{hidden:true,key:true}** : In questo caso viene nascosta(hidden:true) e impostata come chiave (key:true).

Progettazione

- Struttura
- Codice nel Model
- Codici in menLib
 - Funzione GetMenu
 - Funzine caricaTreeLegami

Struttura

Come abbiamo già visto il treeGridModel utilizzato è quello Adjacency.

E' importante definire, nella parte di programmazione, che comportamendo dovranno assumere i vari

campi.

Le possibili configurazioni sono:

- Level : Che assume valori integer, e indica il livello di profondità
- **Parent :** Indica ,nell'array, l'indice del padre. Quale livello è padre. (Assume anche esso valori integer)
- isLeaf:
 - **true:** Indica se è o meno una foglia. Nota. La foglia è l'ultimo elemento dell'albero, quindi non è chiaramente espandibile.
 - false: Se impostato indica che l'elemento non è una foglia.
- · loaded:
 - **true:** Se impostato appena viene caricata la grid, vengono anche visualizzate le foglie.
 - **false:** Se impostato appena viene caricata la grid, non verranno visualizzate le varie foglie del parent, quindi sarà necessario un click per visualizzare le varie foglie.
- expanded:
 - **true:** Se impostato si avrà l'effetto grid ad albero.
 - false: Se impostato false, la tabella non visualizzerà nulla, perchè, come definito da programma utilizzando l'Adjacency Model, il programma si aspetta un effetto ad "albero".

Nota.

Se si vuole ottenere un effetto ad albero, dove vengono visualizzate automaticamente i nodi e le loro foglie, è necessario impostare sia **Expanded** che **Loaded** in "True". Se almeno uno dei 2 risulterà falso l'effetto sarà lo stesso: verrà visualizzato solo il nodo principale.

Model

Codice nel Model:

```
$\text{striction} \text{$\text{tree}} = $\text{this->menLib->getMenu}($\text{voceMenu}, $\text{sonly_menu} = false, $\text{gruppo}, $\text{return_model} = 'adjacency', $\text{filtro} = false); // Va a chiamare la funzione getMenu nella libreria menLib $\text{surr} = \text{arrayTable'} => $\text{this->tree}, // Con il risultato ne costruirà un array 'rowIndex' => 'idx');

$\text{griglia} = \text{new TableView}($\text{this->tableId}, $\text{surr});
$\text{griglia->setPageNum}(1);
$\text{griglia->setPageRows}('1000');
$\text{....}$
```

menLib

https://wiki.nuvolaitalsoft.it/ Printed on 2025/11/28 21:27

2025/11/28 21:27 3/7 Grid ad Albero (TreeGrid)

Funzione GetMenu

Codice nella "menLib":

```
public function getMenu($root = 'TI_MEN', $only_menu = false, $gruppo = ''
$return model = 'adjacency', $filtro = true) {
  sinc = 0;
        $albero = array(); // Setta $albero come array
        $albero[$inc]['INDICE'] = $inc;
        $albero[$inc]['PER_FLAGVIS'] = menLib::decodeFlag();
        $albero[$inc]['PER FLAGDEL'] = menLib::decodeFlag();
if (!$filtro) { // Se abbiamo impostato filtro:false o non ci serve
possiamo anche omettere questa condizione.
            if (!$Ita puntimenu rec) {
                $pm id = -1; // Verrà usato nel salvataggio dei permessi
                $pm descrizione = $me descrizione;
                $albero[$inc]['PER FLAGVIS'] = menLib::decodeFlag(1);
                $albero[$inc]['iconaVis'] = $this->iconaCons;
            } else {
                $pm id = $Ita puntimenu rec['pm id'];
                $pm_descrizione = $Ita_puntimenu_rec['pm_descrizione'];
                $sql = "SELECT * FROM MEN PERMESSI WHERE PER GRU = '" .
$gruppo . "' AND PER MEN = '" . $Ita menu root rec['me menu'] . "' AND
PER VME = '" . $Ita puntimenu rec['pm voce'] . "'";
                $Men permessi rec = ItaDB::DBSQLSelect($this->ITALWEB_DB,
$sql, false);
                if ($Men permessi rec) {
                    $albero[$inc]['PER FLAGVIS'] =
menLib::decodeFlag($Men_permessi_rec['PER FLAGVIS']);
                    $albero[$inc]['PER FLAGACC'] =
menLib::decodeFlag($Men permessi rec['PER FLAGACC']);
                $privilegio =
$this->menLib->privilegiPuntoMenu($Ita_menu_rec['me_menu'],
$Ita_puntimenu_rec, array($gruppo), 'PER_FLAGVIS', $this->defaultVis);
                $albero[$inc]['iconaVis'] = $this->getIcona($privilegio);
                $privilegio =
$this->menLib->privilegiPuntoMenu($Ita menu rec['me menu'],
$Ita_puntimenu_rec, array($gruppo), 'PER_FLAGDEL', $this->defaultDel);
                $albero[$inc]['iconaDel'] = $this->getIcona($privilegio);
```

```
$albero[$inc]['pm voce'] = $root;
        $albero[$inc]['me id'] = $chiave;
        $albero[$inc]['pm id'] = $pm id;
        $albero[$inc]['pm descrizione'] = $pm descrizione;
        $albero[$inc]['pm sequenza'] = 0;
        $albero[$inc]['level'] = 0; //Imposta il livello 0
        $albero[$inc]['parent'] = NULL;
                                                  // Imposta il parent
nullo perchè è il primo nodo
        $albero[$inc]['isLeaf'] = 'false';
                                                  // Non è una foglia,
essendo il primo nodo che avremo
        $albero[$inc]['expanded'] = 'true';
                                                  //Rende l'albero
espandibile
        $albero[$inc]['loaded'] = 'true';
                                                  // Carica le foglie
insieme all'albero
        $save count = count($albero);
                                                  // incrementa il
$save count
        $albero = $this->caricaTreeLegami($chiave, $albero, 1, $inc,
$only menu, $filtro); //Richiama la funzione caricaTreeLegami
        if ($save count == count($albero)) {      // Nel caso in cui il
$save count sia uguale al $count($albero) significa che non ci sono sotto
livelli o foglie e quindi viene impostato come foglia.
           $albero[$inc]['isLeaf'] = 'true'; //Impostato come foglia.
        return $albero;
   }
```

Funzine caricaTreeLegami

Codice nella "menLib":

```
public function caricaTreeLegami($chiave, $albero, $level, $indice,
$only menu = false, $filtro = true) {
        if ($level == 10) {
                                                   // Impostato a 10 perchè è
praticamente impossibile che si arrivi ad avere più di 10 livelli.
            return $albero;
        }
        $sql = "SELECT * FROM ita puntimenu WHERE me id = '" . $chiave . "'
ORDER BY pm sequenza";
        $Ita puntimenu_tab = ItaDB::DBSQLSelect($this->ITALSOFT_DB, $sql,
true);
        if ($Ita puntimenu tab) {
            foreach ($Ita puntimenu tab as $i => $Ita puntimenu rec) {
                if ($only menu && $Ita puntimenu rec['pm categoria'] !=
'ME') {
                    continue;
                }
                $inc = count($albero);
```

https://wiki.nuvolaitalsoft.it/ Printed on 2025/11/28 21:27

2025/11/28 21:27 5/7 Grid ad Albero (TreeGrid)

```
$albero[$inc] = $Ita puntimenu rec;
                $albero[$inc]['INDICE'] = $inc; // Imposta alla chiave
univoca indice come $inc
                $albero[$inc]['level'] = $level;
                $albero[$inc]['parent'] = $indice;
                $albero[$inc]['expanded'] = 'false';
                $albero[$inc]['loaded'] = 'false';
                $albero[$inc]['isLeaf'] = 'true';
                // Aquisizione privilegi
                $sql = "SELECT * FROM ita menu WHERE me id = '" . $chiave .
" ' " :
                $Ita menu rec = ItaDB::DBSQLSelect($this->ITALSOFT DB, $sql,
false);
                if ($filtro) {
                    // Se è un menu, verifico se esiste effettivamente
                    if ($Ita puntimenu rec['pm categoria'] == 'ME') {
                        $sql = "SELECT * FROM ita menu WHERE me menu = '" .
$Ita puntimenu rec['pm voce'] . "'";
                        $Ita_menu_giu_rec =
ItaDB::DBSQLSelect($this->ITALSOFT DB, $sql, false);
                        if (!$Ita_menu_giu_rec) {
                            unset($albero[$inc]);
                            continue;
                        }
                    }
                    $utente = $this->utenteAttuale;
                    $gruppi = $this->getGruppi($utente);
                    $defaultVis = App::getConf("Menu.visibilityDefault");
                    $privilegio =
$this->privilegiPuntoMenu($Ita menu rec['me menu'], $Ita puntimenu rec,
$gruppi, 'PER FLAGVIS', $defaultVis);
                    if (!$privilegio) {
                        unset($albero[$inc]);
                        continue:
                    }
                } else { // Sto amministrando tramite menAuthConfig!!!!!
                    $gruppo = $ POST[$this->gestorePermessi];
                    $sql = "SELECT * FROM MEN PERMESSI WHERE PER GRU = '"
$gruppo . "' AND PER_MEN = '" . $Ita_menu_rec['me_menu'] . "' AND PER_VME =
'" . $Ita puntimenu rec['pm voce'] . "'";
                    $Men_permessi_rec =
ItaDB::DBSQLSelect($this->ITALWEB DB, $sql, false);
                    if ($Men permessi rec) {
                        $albero[$inc]['PER_FLAGVIS'] =
menLib::decodeFlag($Men_permessi_rec['PER_FLAGVIS']);
                    } else {
                        $albero[$inc]['PER FLAGVIS'] = menLib::decodeFlag();
```

```
$albero[$inc]['PER FLAGACC'] = menLib::decodeFlag();
                    }
                    // Icone per griglie 'permessi menu'
                    $privilegio =
$this->privilegiPuntoMenu($Ita menu rec['me menu'], $Ita puntimenu rec,
array($gruppo), 'PER_FLAGVIS', $this->defaultVis);
                    $albero[$inc]['iconaVis'] =
$this->getIcona($privilegio);
                    $privilegio =
$this->privilegiPuntoMenu($Ita menu rec['me menu'], $Ita puntimenu rec,
array($gruppo), 'PER_FLAGDEL', $this->defaultDel);
                    $albero[$inc]['iconaDel'] =
$this->getIcona($privilegio);
                // Fine acquisizione privilegi
                if ($Ita puntimenu rec['pm categoria'] == 'ME') {
                    $albero[$inc]['isLeaf'] = 'false';
                    $sql = "SELECT * FROM ita_menu WHERE me_menu = '" .
$Ita puntimenu rec['pm voce'] . "'";
                    $Ita menu giu rec =
ItaDB::DBSQLSelect($this->ITALSOFT DB, $sql, false);
                    $me id = $Ita menu giu rec['me id'];
                    $save_count = count($albero);
                    $albero = $this->caricaTreeLegami($me id, $albero,
$level + 1, $inc, $only_menu, $filtro);
                    if ($save count == count($albero)) {
                                                               // Fa
riferimento a sestessa incrementando però il level di 1. (Fino a che non
arriverà a 10, come da controllo impostata all'inizio della menLib,nella
funzione caricaTreeLegami)
                        $albero[$inc]['isLeaf'] = 'true';
                    } else {
                        if (!$filtro) {
                            $albero[$inc]['pm_descrizione'] = "<span</pre>
style=\"font-weight:bold;color:darkred;\">" .
$albero[$inc]['pm descrizione'] . "</span>";
                        }
                    }
                }
            }
        return $albero;
   }
```

https://wiki.nuvolaitalsoft.it/ Printed on 2025/11/28 21:27

2025/11/28 21:27 7/7 Grid ad Albero (TreeGrid)

From:

https://wiki.nuvolaitalsoft.it/ - wiki

Permanent link:

https://wiki.nuvolaitalsoft.it/doku.php?id=sviluppo:grid_albero&rev=1351171057

Last update: 2018/03/19 10:45

