2025/11/28 21:27 1/5 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

Esempio:

Nel Programma andremo a chiamare la funzione "getMenu" dalla libreria "menLib". Il risultato chiaramente poi finirà nell'array per costruire la Grid.

menLib

2025/11/28 21:27 3/5 Grid ad Albero (TreeGrid)

Funzione GetMenu

Nella funzione getMenu possiamo notare che:

\$inc viene settato a 0

Viene dichiarato l'array \$albero e gli viene attribuito il valore \$inc nel campo 'INDICE' di \$inc. In questo caso avremo nell'array:

```
• [0]
• INDICE = 0
```

Codice nella "menLib":

Successivamente dovremo impostare alcuni campi obbligatori:

'level', 'parent', 'isLeaf', 'expanded', 'loaded'

E successivamente possono essere impostatati anche gli altri campi all'interno della tabella(in questo caso nell'array):

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);
                $albero[$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';
                $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 +
```

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

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

```
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;
   }
```

From:

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

Permanent link:

https://wiki.nuvolaitalsoft.it/doku.php?id=sviluppo:grid_albero&rev=1351173489

Last update: 2018/03/19 10:45

