Progettazione

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.

Guida

Model

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

```
        .....
        $this->tree = $this->menLib->getMenu($voceMenu, $only_menu = false,
$gruppo, $return_model = 'adjacency', $filtro = false);
        $arr = array('arrayTable' => $this->tree,
            'rowIndex' => 'idx');

        $griglia = new TableView($this->tableId, $arr);
        $griglia->setPageNum(1);
        $griglia->setPageRows('1000');
    ....
```

Libreria: menLib

Nella libreria menLib troveremo chiaramente molte funzioni, ma ce ne interessano 2 in particolare:

- Funzione getMenu
- Funzione caricaTreeLegami

Funzione getMenu

```
 public function getMenu($root = 'TI_MEN', $only_menu = false, $gruppo = '',
$return_model = 'adjacency', $filtro = true) {
  $inc = 0;
        $albero = array();
        $albero[$inc]['INDICE'] = $inc;
```

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

Successivamente dovremo impostare alcuni campi obbligatori:

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

```
        .....
        $albero[$inc]['INDICE'] = $inc;
        $albero[$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
```

Dopo di che possono essere impostatati anche gli altri campi all'interno dell'array :

```
        ......
        $albero[$inc]['loaded'] = 'true';         // Carica le foglie
insieme all'albero
        $albero[$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;
        $save_count = count($albero);
```

Alla fine dell'impostazione dei campi, possiamo notare **“$save_count = count($albero);”**.
La variabile **$save_count** viene quindi settata con il conteggio totale delle righe in **$albero**, che servirà più avanti come controllo.

A questo punto ci servirà richiamare un'altra funzione, per definire i vari legami tra i livelli:, la funzione **creaTreeLegami** contenuta sempre nella libreria menLib.

```
        $albero = $this->caricaTreeLegami($chiave, $albero, 1, $inc,
$only_menu, $filtro);
        if ($save_count == count($albero)) {
            $albero[$inc]['isLeaf'] = 'true';
        }
        return $albero;
    }
```

Possiamo notare che:

- al campo $level viene assegnato il valore 1
- al campo $parent viene assegnato il valore $inc, che in questo caso vale 0 (I livelli che verranno creati avranno quindi come parent 0)

Funzine caricaTreeLegami

```
    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;
                }
```

In questa prima parte di codice, possiamo notare un primo controllo: **if ($level == 10)**. Questo controllo serve ad evitare che la funzione venga ripetuta all'infinito,perchè come vedremo più avanti farà riferimento a sestessa, e serve a definire anche un numero massimo di sottolivelli che si possono avere. (In genere è impostata a **10** perchè è praticamente impossibile che si presenti un menu con più di 10 sottolivelli.)\

Questo All'interno del foreach

```
                $inc = count($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 + 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
style=\"font-weight:bold;color:darkred;\">" .
$albero[$inc]['pm_descrizione'] . "</span>";
                        }
                    }
                }
            }
        }
        return $albero;
    }
```

</code>

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

Permanent link:
https://wiki.nuvolaitalsoft.it/doku.php?id=sviluppo:proggridalbero&rev=1351175669

Last update: **2018/03/19 10:45**