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

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:

```
        .....
        $this->tree = $this->menLib->getMenu($voceMenu, $only_menu = false,
$gruppo, $return_model = 'adjacency', $filtro = false);//Va a chiamare la
funzione getMenu nella libreria menLib
        $arr = array('arrayTable' => $this->tree,  // Con il risultato ne
costruirà un array
            'rowIndex' => 'idx');

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

---

menLib

Funzione GetMenu

Codice nella “menLib” :

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

        $albero[$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);
                $albero[$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
```

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