## **Progettazione**

- Struttura
- Guida
- 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.

# 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.

```
$\text{sthis->tree} = \text{$this->menLib->getMenu($voceMenu, $only_menu} = false,
$\text{gruppo, $return_model} = 'adjacency', $filtro = false);
$\text{sarr} = \text{arrayTable'} => \text{$this->tree,}
$\text{'rowIndex'} => 'idx');
$\text{griglia} = \text{new TableView($this->tableId, $arr);}
$\text{griglia->setPageNum(1);}
$\text{griglia->setPageRows('1000');}
$\text{.....}$
```

## menLib

#### **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'

2025/12/18 14:38 3/4 Progettazione

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

```
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 +
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:proggridalbero&rev=1351173792

Last update: 2018/03/19 10:45

