

Multiselezione con più pagine

Una Griglia Multiselect serve per rendere selezionabili più di una Riga alla volta.

E' possibile però che capiti di avere una Griglia con molte righe e che ciò renda i tempi di creazione/aggiornamento della tabella molto lunghi.

Il problema può essere risolto utilizzando una Multiselect con più Pagine, ma dato che non è ancora implementato un sistema di *cattura delle righe precedentemente selezionate* dovremo utilizzare la funzione **MultiSelezionePost**

MultiSelezionePost

Con questa funzione potremo:

- tenere memorizzate tutte le righe precedentemente selezionate.
- visualizzare in ogni pagina le righe già selezionate
- deselezionare le righe

Per poter utilizzare questa funzione avremo bisogno prima di tutto di una **variabile in Session** che conterrà tutte le righe precedentemente selezionate e il nome della griglia: Le dichiariamo:

```
..
public $gridLicenze = "gfmConLicenze_gridLicenze";
public $SelezioneTab = array();
..
```

Nel construct:

```
function __construct() {
    parent::__construct();
    try {
        $this->GAFIERE_DB = ItaDB::DBOpen('GAFIERE');
        $this->SelezioneTab = App::$utente->getKey($this->nameForm .
'_SelezioneTab');
        ...
    }
}
```

Nel destruct:

```
function __destruct() {
    parent::__destruct();
    if ($this->close != true) {
        App::$utente->setKey($this->nameForm . '_SelezioneTab',
$this->SelezioneTab);
        ...
    }
}
```

Nel close:

```
public function close() {
    App::$utente->removeKey($this->nameForm . '_SelezioneTab');
    Out::closeDialog($this->nameForm);
}
```

Dopo di che chiamiamo la funzione nel **onClickTablePager** :

```
...
$sql = $this->CreaSql();
$this->SelezioneTab=$this->gfmLib->MultiSelezionePost($this->SelezioneTab,$t
his->gridLicenze,$sql);
...
```

Notiamo:

- L'assegnazione di **\$this→SelezioneTab**
- Il richiamo della funzione **MultiSelezionePost** con l'utilizzo di 3 campi:
 - \$this→SelezioneTab: Utilizziamo la variabile in session in quanto contiene i campi già selezionati (ovviamente se si tratta del primo richiamo sarà vuota)
 - \$this→gridLicenze: rappresenta il nome della griglia, in questo caso: **gfmConLicenze_gridLicenze**
 - \$sql: la stessa sql che utilizziamo per creare la griglia, in modo che la funzione che stiamo chiamando si scorra gli stessi campi e possa controllare i ROWID selezionati.

La funzione:

```
public function MultiSelezionePost($SelezioneTab,$Griglia,$sql){
// Invia in POST le row selezionate nella pagina precedente
foreach ($SelezioneTab as $key => $Result_rec) {
    if($_POST['jqg_'].$Griglia.'_'.$key] == '0'){
        $_POST['jqg_'].$Griglia.'_'.$key] = '0';
        $SelezioneTab[$key ] = '0';
    }else{
        if(!Out::valore('jqg_'].$Griglia.'_'.$key,'1')){
            $_POST['jqg_'].$Griglia.'_'.$key]='1';
        }
    }
}
//Si riscorre tutte le righe e quelle che trova controlla.
$Griglia_tab=ItaDB::DBSQLSelect($this->getDATABASE(),$sql, true);
foreach ($Griglia_tab as $Griglia_rec) {
    if( $_POST['jqg_'].$Griglia.'_'.$Griglia_rec['ROWID']] == '1'){
        $NuovoConto[$Griglia_rec['ROWID']]='1';
        Out::valore('jqg_'].$Griglia.'_'.$Griglia_rec['ROWID'],'1');
    }else {
        if($SelezioneTab[$Griglia_rec['ROWID']] == '1'){
            $NuovoConto[$Griglia_rec['ROWID']]='0';
        }
    }
}
```

```
    }  
    $SelezioneTab = $NuovoConto;  
    return $SelezioneTab;  
}
```

Possiamo notare: **\$SelezioneTab,\$Griglia,\$sql**

- **\$SelezioneTab**: Rappresenta l'array che contiene tutte le righe selezionate. -

From:

<https://wiki.nuvolaitalsoft.it/> - **wiki**

Permanent link:

<https://wiki.nuvolaitalsoft.it/doku.php?id=sviluppo:multiselectmultipage&rev=1361378975>

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

