

TUTIFRUTY

GRID

Dokumentácia pre programátorov

Stefan Horvath

7.1.2013

Dokumentácia popisuje komponent Grid, jeho vlastnosti a použitie.

História dokumentu

| Verzia | Autor | Dátum | Vykonané zmeny |
|--------|----------------|----------|----------------|
| 0.10 | Štefan Horváth | 7.1.2013 | Prvá verzia |

Všeobecné informácie

Grid je webový komponent napísaný v jazykoch PHP a JavaScript. Je všeobecne použiteľný na php frameworkoch bežiacich pod návrhovým vzorom MVC.

Grid slúži na zobrazenie formátovaných údajov vo forme tabuľky. Tiež umožňuje pridávať, editovať a mazať záznamy v tabuľke, pričom pre každú vykonanú operáciu volá udalosť, pomocou ktorej môže programátor upraviť databázu alebo iné dátové úložisko.

Grid obsahuje atribúty pre jednotlivé riadky a stĺpce, vďaka ktorým je možné nastaviť individuálne pre každý riadok/stĺpec, či má byť viditeľný, editovateľný, mazateľný. Tiež existujú atribúty pre celý grid, ktoré umožňujú nastaviť parametre platné pre všetky riadky a stĺpce.

Grid sa používa jednoducho tak, že sa zahrnie súbor s implementáciou (grid.php) do projektu a následne sa zavolá jeho konštruktor. Po zavolaní metódy na zobrazenie sa grid automaticky vypíše s defaultným dizajnom, ktorý je možné upraviť pomocou CSS.

Použitie

Inštalácia

Inštalácia je jednoduchá, treba skopírovať súbor grid.php do vášho projektu. Následne treba zahrnúť súbor na stránku na ktorej má byť použitý (pomocou *include*).

Naplnenie údajmi a vypísanie

Inštanciu gridu vytvoríme pomocou *new Grid()*. Keď máme inštanciu, treba grid naplniť údajmi. To sa realizuje pomocou metódy *bind*.

Prvý parameter je vo forme *array of array* alebo *array of object*. Podľa indexov nižšieho array (object) sa tvoria stĺpce gridu, teda indexy sú nie typu integer ale string.

Druhý parameter je IDčko, podľa ktorého sa identifikuje riadok záznamu.

Dáta vstupujúce do bind-u **musia** obsahovať stĺpec z IDčkom. Metóda bind vracia informáciu typu boolean, či sa naplnenie podarilo alebo nie. Je doporučené realizovať vypísanie gridu a prípadne ďalšie nastavenia iba v prípade, že metóda bind vráti *true*.

Po naplnení údajov je možné zobrazíť grid pomocou metódy *display*.

Príklad:

```
$grid = new Grid();
```

```
$users = array(  
    array('user_id' => 1, 'name' => 'jozo', 'age' => '22', 'food' => 'apple'),  
    array('user_id' => 2, 'name' => 'miso', 'age' => '25', 'food' => 'cheeseburger'),  
    array('user_id' => 3, 'name' => 'fero', 'age' => '30', 'food' => 'space meat'),  
    array('user_id' => 4, 'name' => 'duri', 'age' => '18', 'food' => 'apple')  
);
```

```
if ( $grid->bind($users, 'user_id') ) {  
    $grid->display();  
}
```

```
}
```

CRUD pomocou gridu

CRUD na gride je realizovaný dvoma spôsobmi – interne alebo externe, pričom mazanie je možné spraviť iba externe. Grid ku každému riadku tabuľky generuje tlačítka, cez ktoré je možné robiť CRUD daného riadka (v predchádzajúcom príklade negeneruje žiadne tlačítka, nakoľko v príklade nebol CRUD inicializovaný). Interný spôsob je ten, kde sa po kliknutí na tlačítko „upraviť“ zmení riadok na formulár, do ktorého je možné priamo písať zmeny a externý je ten, kde po kliknutí na „upraviť“ je používateľ presmerovaný na formulár na inej stránke.

Inicializácia CRUD externým spôsobom

Pred zavolaním metódy `display` je treba nastaviť URL na jednotlivé stránky (resp. jednotlivé CRUD operácie – `create`, `update`, `delete`), na ktoré má grid presmerovať po kliknutí na tlačidlo príslušnej operácie. Grid generuje k tlačítkam pre jednotlivé riadky tabuľky nastavené URL, ku ktorým ale v prípade `update` a `delete` dopĺňa IDčko, aby príslušný controller vedel o aký záznam sa jedná.

Príklad:

```
$grid->add_url = "users/add";  
$grid->edit_url = "users/edit";  
$grid->remove_url = "users/delete";  
$grid->add_mode = "external";  
$grid->edit_mode = "external";
```

Inicializácia CRUD interným spôsobom

Inicializácia rovnako ako v prípade externého spôsobu s tým rozdielom, že sa na miesto *external* napíše *internal* (v podsate je to možné úplne vynechať, nakoľko `internal` je predvolený spôsob).

Je ešte jeden rozdiel a to ten, že po kliknutí na tlačítko `create/edit` sa prerobí riadok tabuľky na formulár. Tento formulár sa odosiela po kliknutí tlačítka „potvrď“ cez POST na príslušný controller.

Príklad:

```
$grid->add_url = "users/add";  
$grid->edit_url = "users/edit";  
$grid->remove_url = "users/delete";  
$grid->add_mode = "internal"; // je možné úplne vynechať  
$grid->edit_mode = "internal"; // je možné úplne vynechať
```

Atribúty pre stĺpce

Dotaz na stĺpec

Stĺpce sú indexované podľa indexov vstupných dát, ktoré boli naplnené cez `bind`. Na stĺpec sa dotazujeme pomocou metódy `header`, ktorá prijíma ako parameter index stĺpca (teda string) a vracia objekt stĺpca.

Príklad:

```
$grid = new Grid();
```

```

$users = array(
    array('user_id' => 1, 'name' => 'jozo', 'age' => '22', 'food' => 'apple'),
    array('user_id' => 2, 'name' => 'miso', 'age' => '25', 'food' => 'cheeseburger'),
    array('user_id' => 3, 'name' => 'fero', 'age' => '30', 'food' => 'space meat'),
    array('user_id' => 4, 'name' => 'duri', 'age' => '18', 'food' => 'apple')
);

if ( $grid->bind($users, 'user_id') ) {
    $col = $grid->header('name'); // získa stĺpec obsahujúci mená
    $grid->display();
}

```

V prípade že pod daným indexom nie je žiaden stĺpec, vráti NULL.

Atribúty stĺpca

| názov | typ | predvolená hodnota | popis |
|------------------|-----------|--------------------|---|
| visible | boolean | true | či sa má stĺpec zobraziť |
| editable | boolean | true | či sú povolené úpravy na tomto stĺpci (nemá zmysel pri externom spôsobe CRUD) |
| text | string | <prázdny_string> | text, ktorý sa má zobraziť v hlavičke stĺpca, v prípade prázdneho stringu sa zobrazí index |
| component | Component | - | nastavenie komponentu, ktorý sa má zobraziť na stĺpci v danom riadku pri internej úprave alebo pridávaní (viď nižšie) |

Metódy stĺpca

set_anchor(controller, id)

Metóda nastavuje, aby sa dáta v tomto stĺpci zobrazili ako link na controller.

controller - Názov controlleru, na ktorý má link odkazovať.

id - Názov stĺpca, ktorý obsahuje id-čko (id-čko z tohto stĺpca sa pripisuje k URL linku na controller, aby bolo možné ďalej identifikovať kliknutý záznam).

set_datetime(inputFormat, outputFormat)

Metóda nastavuje formátovanie dátumu. V prípade, že text v bunke nezodpovedá dátumu, nezobrazí sa nič.

inputFormat - Formátovací reťazec, podľa ktorého sa parsuje string v bunke pod daným stĺpcom na vytvorenie internej štruktúry datetime.

outputFormat - Formátovací reťazec, podľa ktorého sa formátuje čas a dátum na výstupe.

Bližšie informácie o formátovacích značkách na <http://php.net/manual/en/function.date.php>

set_numformat(format)

Metóda nastavuje formátovanie čísla. V prípade nevalidného formátovacieho reťazca na pri display zobrazí varovanie.

Syntax: `set_numformat({<desatinnych_miest>:<desatinna_bodka>:<separator_tisiciek> })`

Default: `desatinnych_miest = 0 desatinna_bodka = "." separator_tisiciek = ""`

Number: 12345.6789

```
set_numformat('Suma {2:: } EUR') --> Suma 12 345.68 EUR
set_numformat('Suma {2::} EUR') --> Suma 12345.68 EUR
set_numformat('Suma {2:.} EUR') --> Suma 12345.68 EUR
set_numformat('Suma {2:: } EUR') --> Suma 12 345,68 EUR
set_numformat('Suma {2:} EUR') --> Suma 12345,68 EUR
set_numformat('Suma {2} EUR') --> Suma 12345,68 EUR
set_numformat('Suma {:: } EUR') --> Suma 12 346 EUR
set_numformat('Suma {} EUR') --> Suma 12346 EUR
set_numformat('Suma EUR') --> Suma EUR
set_numformat('{}') --> 12346
```

Nastavenie komponentu v stĺpci

Pri internom spôsobe editácie a pridávaní sa prerobí príslušný riadok na formulár. Prerobia sa všetky bunky spadajúce pod stĺpce, ktoré majú povolenú editáciu. Predvolene je nastavený ako editovací komponent textbox. Je možné nastaviť aj iný komponent na editáciu. Momentálne je na výber textbox a combobox.

Pre nastavenie komponentu je nutné najprv získať objekt stĺpca. Ten obsahuje objekt component ako svoj atribút. Cez tento atribút sa realizuje inicializácia požadovaného komponentu.

Textbox

Nastavenie tohto komponentu sa robí jednoduchým priradením textu „textbox“ na atribút *type* v atribúte *component*.

```
$grid->header('name')->component->type = "textbox";
```

Combobox

Nastavenie tohto komponentu sa robí priradením textu „combobox“ na atribút *type* v atribúte *component*. Keďže combobox ponúka na výber nejaké možnosti, je nutné komponent naplniť dátami podobne ako celý grid. To sa robí pomocou metódy *bind*, ktorá prijíma nasledujúce parametre:

- *array of array* alebo *array of object* – každý „riadok“ musí mať pod sebou minimálne 2 záznamy, jeden reprezentuje idčko, druhý samotný text ktorý sa zobrazuje v comboboxe užívateľovi na výber
- *id_index* – index, ktorý hovorí na ktorom indexe je ID záznamu v predchádzajúcom poli dát
- *value_index* – index, ktorý hovorí na ktorom indexe je hodnota záznamu v predchádzajúcom poli dát

Príklad:

```
$users = array(
    array('user_id' => 1, 'name' => 'jozo', 'age' => '22', 'food' => 'apple'),
    array('user_id' => 2, 'name' => 'miso', 'age' => '25', 'food' => 'cheeseburger'),
    array('user_id' => 3, 'name' => 'fero', 'age' => '30', 'food' => 'space meat'),
    array('user_id' => 4, 'name' => 'duri', 'age' => '18', 'food' => 'apple')
);
```

```
$foods = array(
```

```

        array('food_id' => 1, 'food_name' => 'apple', 'food_healthy' => 'best', 'food_price' => '0.259', 'food_date' => '2012-12-20 00-00'),
        array('food_id' => 2, 'food_name' => 'cheeseburger', 'food_healthy' => 'worst', 'food_price' => '1.2', 'food_date' => '2012-12-28 00-00'),
        array('food_id' => 3, 'food_name' => 'sausage', 'food_healthy' => 'bad', 'food_price' => '0.64389', 'food_date' => '2012-12-30 00-00'),
        array('food_id' => 4, 'food_name' => 'bread', 'food_healthy' => 'good', 'food_price' => '2.058', 'food_date' => '2012-12-16 00-00'),
        array('food_id' => 6, 'food_name' => 'space meat', 'food_healthy' => 'best', 'food_price' => '15204', 'food_date' => '2012-12-10 00-00')
    );

```

```
$grid = new Grid();
```

```

if ( $grid->bind($users, 'user_id') )
{
    $grid->header('user_id')->visible = false;
    $grid->header('name')->set_anchor("users/detail", "user_id");
    $grid->header('food')->component->type = "combobox";
    $grid->header('food')->component->bind($foods, 'food_id', 'food_name');

    $grid->add_url = "{$this->router->class}/add";
    $grid->edit_url = "{$this->router->class}/edit";
    $grid->remove_url = "{$this->router->class}/delete";

    $grid->display();
}

```

Atribúty pre riadky

Dotaz na riadok

Riadky sú indexované podľa hodnôt zo vstupných dát, pričom ako index sa berú hodnoty v stĺpci s indexom zadaným pri bind. Na riadok sa dotazujeme pomocou metódy `row`, ktorá prijíma ako parameter index riadka (teda integer) a vracia objekt riadka.

Príklad.

```
$row = $grid->row(2); // získa riadok identifikovaný ako 2
```

V prípade že pod daným indexom nie je žiaden riadok, vráti NULL.

Atribúty riadka

| názov | typ | predvolená hodnota | popis |
|------------------|---------|--------------------|--|
| visible | boolean | true | či sa má riadok zobraziť |
| editable | boolean | true | či sú povolené úpravy na tomto riadku (ak nie sú, nezobrazí sa ani tlačítka na úpravu) |
| removable | boolean | true | či je povolené zmazanie tohto riadka (ak nie, nezobrazí sa ani tlačítka na zmazanie) |

CSS pre grid

Grid generuje pre jednotlivé svoje elementy atribút *class*, ktorý slúži ako selektor na CSS štýl. Týmto spôsobom je možné definovať vlastný štýl pre jednotlivé elementy gridu. Robí sa to pridaním alebo úpravou príslušného selektora v CSS súbore. Zoznam a popis selektorov je v nasledujúcej tabuľke.

| názov class | popis |
|------------------------|--|
| grid_container | reprezentuje kontajner, v ktorom sa celý grid nachádza |
| grid_table | element <i>table</i> , v ktorom sú riadky a stĺpce |
| grid_header | reprezentuje celý riadok hlavičky (úplne prvý element <i>tr</i>) |
| grid_header_cell | reprezentuje všetky bunky (elementy <i>th</i>) hlavičky okrem prvej a poslednej |
| grid_header_cell_first | reprezentuje prvú bunku (element <i>th</i>) hlavičky |
| grid_header_cell_last | reprezentuje poslednú bunku (element <i>th</i>) hlavičky |
| grid_row | reprezentuje riadok v tabuľke |
| grid_cell | reprezentuje akúkoľvek bunku (element <i>td</i>) všetkých riadkov tabuľky okrem prvej a poslednej |
| grid_cell_first | reprezentuje prvú bunku (element <i>td</i>) všetkých riadkov tabuľky |
| grid_cell_last | reprezentuje poslednú bunku (element <i>td</i>) všetkých riadkov tabuľky |
| grid_cell_editing | reprezentuje bunku, v ktorej je editovací komponent (textbox alebo combobox) |
| grid_form_elem | reprezentuje editovací komponent (textbox alebo combobox) |
| grid_row_btn_cell | reprezentuje bunky, v ktorých sú umiestené funkčné tlačítka alebo anchory |