

Обучение: Таблица PHP/SQL часть 10

Июл 25 2010Компонент QuickSearch

Вы скоро будете в состоянии создать свои собственные удивительные таблицы. Давайте изучим компонент QuickSearch.

Мы объяснили ранее , как сделать регулярный поиск по нашей таблице. Это все же потребовало, чтобы пользователь нажал на Вкладку поиска и заполнил необходимы строки... Порой это бывает ОЧЕНЬ сложной задачей для большинства пользователей. Компонент Quicksearch - в прикольный textbox, в который пользователь вводит свой текст, жмет Enter, и затем... Чтож это зависит от того, что мы делаем...

Идем далее и делаем поиск для всей таблицы.

Прежде, чем мы начнем, мы должны добавить .js файл к нашему главному заголовку HTML:

```
<script type="text/javascript" src="searchfield.js"></script>
```

Вы можете найти файл в папке примеров/формы в архиве extjs, или Вы можете загрузить это здесь:

<http://extjs.com/deploy/dev/examples/form/SearchField.js>

Во-первых, мы добавляем это к нашей сетке в нашей панели инструментов. Мы только должны снабдить его основными параметрами которые есть в наших datastores :

```
, '-', new Ext.app.SearchField({
    store: PresidentsDataStore,
    params: {start: 0, limit: 15},
    width: 120
})
```

Хорошо, если Вам будет достаточно любопытно, то Вы посмотрите в своем любимом текстовом редакторе, что содержится в searchfield.js. И там, Вы будете видеть, что есть параметр, названный 'query', который отправляется и что затем массив перезагружается. Теперь... DataStore Президенты или будут типа 'LISTING' или типа 'Searching' (на основе его baseparams). Хорошо, кнопка Quicksearch просто добавит последовательность, введенную в textbox как дополнительный параметр.

Я только собираюсь осуществить это в случае, если мы просто перечисляем наших президентов. Я позволю Вам выяснять, как сделать это на поиске (quicksearch на существующем поиске).

Давайте откроем наш database.php. Теперь, когда мы включаем getList () функцию, мы должны проверить, есть ли у нас параметр 'query'. И если есть, то мы будем искать некоторые данные, которые похожи на это в нашей таблице. Следовательно, наша новая getList () функция будит выглядеть следующим образом :

```
$query = "SELECT * FROM presidents pr, parties pa WHERE pr.IDparty = pa.IDparty";
// Here we check if we have a query parameter :
if (isset($_POST['query'])) {
    $query .= " AND (pr.firstname LIKE '%" . addslashes($_POST['query']) . "%' OR pr.lastname LIKE '%" . addslashes($_POST['query']) . "%)";
}
$result = mysql_query($query);
$numrows = mysql_num_rows($result);
$start = (integer) (isset($_POST['start']) ? $_POST['start'] : $_GET['start']);
$end = (integer) (isset($_POST['limit']) ? $_POST['limit'] : $_GET['limit']);
$limit = $query . " LIMIT " . $start . ", " . $end;
$result = mysql_query($limit);
[...]
```

Теперь Вам следует взять параметр 'query', снять все пустые места, чтобы получить все слова в последовательности и сделать расширенный поиск.. Вы бы могли также сделать поиск по годам, и т.д.. Выполнение на поиске, который был уже сделан, полностью подобно. Только проверьте переменную 'query' прежде, чем сделать Ваш запрос... Вот и все!

Что стоит запомнить из этого раздела...

- Быстрый поиск шикарная вещь и легок в реализации. Вы можете увидеть его почти везде.

Вы можете увидеть результаты Части 10 нашего обучения [здесь](#).

Вы можете загрузить исходники кода здесь :



icon_folder.jpg

[Загрузить zip файл](#)