

Обучение: Таблица PHP/SQL часть 4

Июнь 21 2010
 Редактирование Таблицы

Отлично, у нас теперь есть таблица, которая отображает материал. Вы можете также видеть все хорошие особенности, которые может предложить Ext. Вы можете изменить размеры или переместить столбцы. Вы можете решить, что показать или скрыть. И лучше всего то, что Вы можете заставить небольшие красные треугольники появиться на углах ячейки, поскольку Вы редактируете их. Теперь это так...

Теперь, Вы спросите меня. Что происходит, когда я редактирую содержимое ячейки? Чтож, ответ весьма прост... Ничего. Хотя Ext делает много хороших вещей, оно позволяет нам обрабатывать все сохранение данных, что означает... То, что мы должны будем закодировать это сами. Это - цель данной части обучающей программы.

Сужение некоторых полей

Прежде, чем идти далее, есть одна вещь, которую мы должны исправить. Как Вы могли видеть в Таблицах SQL, политические партии берутся из предопределенного списка (всего 6 партий) и не должны быть изменены за пределами тех выборов. Лучший способ осуществить это состоит в том, чтобы изменить textbox combobox'ом в ColumnModel и предоставить различные вероятности. Есть два решения:

- Вы можете создать новый DataStore, который соединится с database.php и запросит список партий, чтобы отобразить. Просто подключите combobox к datastore, и готово. Это главным образом полезно, когда данные изменяются во время работы приложения, которое не относится нашей программы.
- Вы можете предопределить скрытый combobox в основном файле HTML и выполнять Ваш combobox согласно этому. Это прекрасно подходит, если Вы уверены, что данные статические. Это делается через команду, 'transform'.

- Вы можете создать simplestore. SimpleStore - это datastore, который определен непосредственно в пределах скрипта.

Теперь, было бы очень хорошей тренировкой опробовать метод DataStore. Как маленький совет, позвольте мне дать Вам код ColumnModel, который должен быть изменен :

```
{
  header: 'Party',
  dataIndex: 'Name',
  width: 120,
  editor: new Ext.form.ComboBox({
    store: PartiesListingDataStore, // or whatever you've called it
    displayField:'Name',           // we have two indexes, ID and Name
    typeAhead: true,
    mode: 'remote',
    triggerAction: 'all',
    selectOnFocus:true
  }),
  hidden: false
}
```

Вы сделали это? Хорошо, давайте пробежимся по третьему решению, используя SimpleStore.

Мы нам нужно содержимое, которое содержит все наши статические данные. Давайте встраивать это теперь:

```
header: 'Party',
dataIndex: 'PartyName',
width: 150,
editor: new Ext.form.ComboBox({
  typeAhead: true,
  triggerAction: 'all',
  store:new Ext.data.SimpleStore({
    fields:['partyValue', 'partyName'],
    data: [['1','No Party'],[2,'Federalist'],[3,'Democratic-Republican'],[4,'Democratic'],[5,'Whig'],[6,'Republican']]
  }),
  mode: 'local',
  displayField: 'partyName',
  valueField: 'partyValue',
  listClass: 'x-combo-list-small'
})
```

Как Вы можете видеть, мы можем определить какое поле отобразить, и какое поле взять как значение. Здорово... Это означает, что мы можем отправить значение партии (IDparty) непосредственно в наш скрипт PHP, и нет нужды в дополнительном запросе.

Редактирование

Вот мы и добрались до основной части... Теперь, когда Вы знаете, что мы - те, которые, предполагают, сделать все сохранение, Вы вероятно задаетесь вопросом, как это должно быть сделано. Есть разные подходы к этой проблеме...

- Вы могли сохранить строку каждый раз, когда изменяюсь содержимое. Это всегда хорошо, если у Вас есть межподчиненные ячейки. Это не лучшее решение, если Вы должны сохранить много данных, хотя...
- Вы могли пройтись по всей таблице и сохранять только ячейки, которые были изменены в данном пункте (текущее автосохранение, или кнопка сохранения по-умолчанию).
- Вы могли просто сохранить всю архитектуру в случайное время.
- Вы могли также не сохранять таблицу и пойти, есть чизбургер.

Выбирать Вам, но это зависит от объема того что вам нужно сохранить. Я не могу сказать, что один метод лучше чем другой...

Мы выберем первый вариант и сохраним 'президента' каждый раз когда одна из его ячеек изменится. Мы делаем это, первоначально создав функцию saveThePresident. Эта функция делает запрос к database.php от недавно созданной задачей 'UPDATEPRES' и пошлет все данные из строки и будет ждать ответа от нашего скрипта сервера PHP. Все очень просто: database.php или ответит 1 для успешного изменения или 0 для остальных. Если изменение было успешно, мы просто перезагружаем нашу память, чтобы видеть, что изменения автоматически появляются в нашей таблице.

```
function saveThePresident(oGrid_event){
  Ext.Ajax.request({
    waitMsg: 'Please wait...',
    url: 'database.php',
    params: {
      task: "UPDATEPRES",
      IDpresident: oGrid_event.record.data.IDpresident,
      FirstName: oGrid_event.record.data.FirstName,
      LastName: oGrid_event.record.data.LastName,
      PartyName: oGrid_event.record.data.PartyName,
      TookOffice: oGrid_event.record.data.TookOffice.format("Y-m-d"), // this time we'll format it thanks to ext
      LeftOffice: oGrid_event.record.data.LeftOffice.format("Y-m-d"),
      Income: oGrid_event.record.data.Income
    },
    success: function(response){
      var result=eval(response.responseText);
      switch(result){
        case 1:
          PresidentsDataStore.commitChanges(); // changes successful, get rid of the red triangles
          PresidentsDataStore.reload();         // reload our datastore.
          break;
        default:
          Ext.MessageBox.alert('Uh uh...', 'We couldn't save him...');
          break;
      }
    },
    failure: function(response){
      var result=response.responseText;
      Ext.MessageBox.alert('error','could not connect to the database. retry later');
    }
  });
}
```

Хорошо, теперь, когда функция сохранения готова, мы должны прикрепить ее туда, где пользователь редактирует ячейку. Это сделано в самом конце нашего javascript файла :

PresidentListingEditorGrid.on('afteredit', saveThePresident);

На заметку: место, где это вставлено, является нерелевантным. Я, возможно, поместил это в конструктор или сразу после того, как моя таблица была проиллюстрирована.

ОК, мы завершили все дела для клиентской стороны. Теперь мы должны пойти к серверу и редактировать наш database.php файл. Мы сначала должны будем распознать название задачи, поэтому давайте добавим другой выбор к нашим входам выключателя:

```
case "UPDATEPRES":
  updatePresident();
  break;
```

Это довольно очевидно. Давайте напомним функцию updatePresident () :

```
function updatePresident()
{
  $IDpresident = $_POST['IDpresident'];
  $FirstName = addslashes($_POST['FirstName']);
  $LastName = addslashes($_POST['LastName']);
  $PartyName = $_POST['PartyName'];
  $TookOffice = $_POST['TookOffice'];
  $LeftOffice = $_POST['LeftOffice'];
  $Income = $_POST['Income'];

  // First, find the $IDparty
  $query = "SELECT IDParty FROM parties WHERE Name='".$_{$PartyName}.'";";
  $result = mysql_query($query);
  if(mysql_num_rows($result)>0){
    $arr = mysql_fetch_array($result);
    $IDparty = $arr['IDParty'];
  } else {
    echo '0';    // failure
  }

  // Now update the president
  $query = "UPDATE presidents SET firstname = '$FirstName', lastname = '$LastName', tookoffice = '$TookOffice', leftoffice = '$LeftOffice', IDparty = '$IDparty', income='$Income' WHERE IDpresident=$IDpresident";
  $result = mysql_query($query);
  echo '1';    // success
}
```

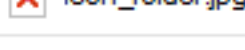
Ну что же пора проверять. Работает? Удивлены? Действительно ли это первый раз, когда все работает? Пораженны?

На заметку : Мы сделали два запроса (в принципе это можно сделать и одним запросом... Вы можете найти как?), один, чтобы получить IDparty для выбранной партии, и другой запрос непосредственно для модификации. Отметьте, что эти запросы не проверяют ошибки. Мы могли закончить с двумя президентами, имеющими то же самое название, или работать в то же самое время. Я позволю Вам откорректировать все это самостоятельно!

Что стоит запомнить из этого раздела...

- Таблица нуждается и в datastore и в columnmodel, чтобы отобразить себя правильно.
- Datastore должен быть перезагружен, если изменения произведены в базе данных, чтобы возможно было сделать дополнительный запрос.

Вы можете увидеть результаты Части 4 нашего обучения [здесь](#).

Вы можете загрузить исходники кода здесь :  [Загрузить zip файл](#)