

Обучение: Таблица PHP/SQL часть 5

Июн 24 2010Добавление Нового Вхождения

Хорошо, мы установили нашу таблицу, и мы теперь в состоянии сделать в ней некоторые изменения. Они записаны непосредственно в нашей таблице SQL. Эта часть обучающей программы будет рассказывать Вам, как добавить новый вход в Вашу таблицу.

Есть много способов, которыми мы могли бы осуществить это. Один из путей, который показывают в Справке, состоит в том, чтобы вставить пустой новый вход в нашу таблицу и редактировать ее. Это - также путь, к которому приходит M. LeCompte. То, что мы собираемся сделать, является немного другим. Мы построим новое окно, которое будет содержать форму для заполнения данными о президенте.

Создание формы

Хорошо, давайте посмотрим на форму datafields, в которой нам понадобится :

- Textfield, чтобы содержать имя.
- Textfield, чтобы содержать фамилию.
- Два datefields, чтобы содержать информацию о входе или покидании офиса.
- Numberfield, чтобы содержать совокупный доход.
- Combobox, чтобы содержать выбранную партию.

Все это должно будит быть включено в форму и содержаться в окне. Давайте определим это в начале скрипта :

```
var PresidentCreateForm;  
var PresidentCreateWindow;
```

Для того, чтобы вещи стали ясными и понятными для всех, мы создадим каждое поле прежде, чем добавить их к форме. Каждое поле будет глобальной переменной (это - определенно не лучший способ!! в будущем Вам создавать различные идентификаторы для каждого поля и иметь справочную информацию на них..... Все же в данном случае у нас будут глобальные переменные, потому что это легче для понимания...), Только помните, что Вы врядли хотели бы иметь 35078 глобальных переменных, влияющих на всю программу... Кроме того, способ, которым мы воспользуемся, будет препятствовать тому, чтобы мы случайно получили две Creation Forms сразу.

На заметку : Если Вы абсолютно не представляете как систематизировать проганный код в вашем приложении, Вам бы не помешало ознакомиться с [the Application Layout Tutorial](#), которое написал Jozef Sakalos.

Давайте определим поля в начале скрипта :

```
var FirstNameField;  
var LastNameField;  
var EnteringOfficeField;  
var LeavingOfficeField;  
var IncomeField;  
var PartyField;
```

Мы должны написать код, который описывает поведение этих полей: (в пределах функции onReady):

```
FirstNameField = new Ext.form.TextField({  
    id: 'FirstNameField',  
    fieldLabel: 'First Name',  
    maxLength: 20,  
    allowBlank: false,  
    anchor: '&nbsp;:95%',  
    maskRe: /([a-zA-Z0-9\ ]+)/S/  
});
```

```
LastNameField = new Ext.form.TextField({  
    id: 'LastNameField',  
    fieldLabel: 'Last Name',  
    maxLength: 20,  
    allowBlank: false,  
    anchor: '&nbsp;:95%',  
    maskRe: /([a-zA-Z0-9\ ]+)/S/  
});
```

```
EnteringOfficeField = new Ext.form.DateField({  
    id: 'EnteringOfficeField',  
    fieldLabel: 'Entering Office',  
    format: '&nbsp;:m/d/Y',  
    allowBlank: false,  
    anchor: '95%'  
});
```

```
LeavingOfficeField = new Ext.form.DateField({  
    id: 'LeavingOfficeField',  
    fieldLabel: 'Leaving Office',  
    allowBlank: false,  
    format: '&nbsp;:m/d/Y',  
    anchor: '95%'  
});
```

```
IncomeField = new Ext.form.NumberField({  
    id: 'IncomeField',  
    fieldLabel: 'Income',  
    allowNegative: false,  
    allowBlank: false,  
    anchor: '95%'  
});
```

Определить все это было достаточно просто. Теперь у нас есть поле партии... Помните, что поле должно переписываться данными, содержащимися в базе данных SQL. В части 3 мы говорили об использовании регулярного DataStore. В части 4 мы видели три различных способа заполнить наше combo-box...

В этой части обучения мы продолжим использовать SimpleStore. Вам стоит иметь в виду, что, выполняя combobox согласно предопределенному HTML, combobox мог бы создать некоторые проблемы совместимости браузера (думайте о бедных IE пользователях немного...):

```
PartyField = new Ext.form.ComboBox({  
    id: 'PartyField',  
    fieldLabel: 'Party',  
    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',  
    allowBlank: false,  
    valueField: 'partyValue',  
    anchor: '95%',  
    triggerAction: 'all'  
});
```

Создание Формы

Теперь, когда поля все готовы и созданы, мы готовы создать нашу форму :

```
PresidentCreateForm = new Ext.FormPanel({  
    labelAlign: 'top',  
    bodyStyle: 'padding:5px',  
    width: 600,  
    items: [{  
        layout: 'column',  
        border: false,  
        items: [{  
            columnWidth: 0.5,  
            layout: 'form',  
            border: false,  
            items: [FirstNameField, LastNameField, PartyField]  
        }, {  
            columnWidth: 0.5,  
            layout: 'form',  
            border: false,  
            items: [EnteringOfficeField, LeavingOfficeField, IncomeField]  
        }]  
    }],  
    buttons: [{  
        text: 'Save and Close',  
        handler: createThePresident  
    }, {  
        text: 'Cancel',  
        handler: function(){  
            // because of the global vars, we can only  
            // instantiate one window... so let's just hide it.  
            PresidentCreateWindow.hide();  
        }  
    }]  
});
```

```
PresidentCreateWindow= new Ext.Window({  
    id: 'PresidentCreateWindow',  
    title: 'Creating a New President',  
    closable: true,  
    width: 610,  
    height: 250,  
    plain: true,  
    layout: 'fit',  
    items: PresidentCreateForm  
});
```

Хорошо... Не так уж плохо? Одна важная вещь нужно все же запомнить - у каждого вложенного элемента есть поле 'layout'. Не забывайте про них, поскольку они являются часто проблемой обсуждаемой на форумах... Ну хорошо, у нас теперь есть большая форма, которая будет отображена. Прежде, чем мы сможем отобразить ее, мы должны обработать события кнопки и создать функции, чтобы сохранить данные. Хорошая вещь о способе, которым мы устанавливаем нашу форму - то, что мы можем напрямую обратиться к нашим данным в полях используя методы setValue () или getValue ().

Сначала давайте настроим некоторые маленькие функции, чтобы работать с нашей формой.

```
// reset the Form before opening it  
function resetPresidentForm(){  
    FirstNameField.setValue("");  
    LastNameField.setValue("");  
    EnteringOfficeField.setValue("");  
    LeavingOfficeField.setValue("");  
    IncomeField.setValue("");  
    PartyField.setValue("");  
}
```

```
// check if the form is valid  
function isPresidentFormValid(){  
    return (FirstNameField.isValid() & & &  
    LastNameField.isValid() & & &  
    EnteringOfficeField.isValid() & & &  
    LeavingOfficeField.isValid()  
    & & & IncomeField.isValid()  
    & & & PartyField.isValid());  
}
```

```
// display or bring forth the form  
function displayFormWindow(){  
    if(!PresidentCreateWindow.isVisible()){  
        resetPresidentForm();  
        PresidentCreateWindow.show();  
    } else {  
        PresidentCreateWindow.toFront();  
    }  
}
```

Создание Президента

Теперь мы всего лишь должны организовать метод сохранения. Вы заметите что это сильно схоже тем что мы делали когда добавляли ячейку в таблицу:

```
function createThePresident(){  
    if(isPresidentFormValid()){  
        Ext.Ajax.request({  
            waitMsg: 'Please wait...',  
            url: 'database.php',  
            params: {  
                task: "CREATEPRES",  
                firstname: FirstNameField.getValue(),  
                lastname: LastNameField.getValue(),  
                enteringoffice: EnteringOfficeField.getValue().format('Y-m-d'),  
                leavingoffice: LeavingOfficeField.getValue().format('Y-m-d'),  
                income: IncomeField.getValue(),  
                party: PartyField.getValue()  
            },  
            success: function(response){  
                var result=eval(response.responseText);  
                switch(result){  
                    case 1:  
                        Ext.MessageBox.alert('Creation OK',  
                            'The president was created successfully.');                        PresidentsDataStore.reload();  
                        PresidentCreateWindow.hide();  
                        break;  
                    default:  
                        Ext.MessageBox.alert('Warning',  
                            'Could not create the president.');                        break;  
                }  
            },  
            failure: function(response){  
                var result=response.responseText;  
                Ext.MessageBox.alert('error',  
                    'could not connect to the database.  
                    retry later');            }  
        });  
    } else {  
        Ext.MessageBox.alert('Warning',  
            'Your Form is not valid!');    }  
}
```

Хорошо... Теперь все, в чем мы нуждаемся, это вызвать displayFormWindow (). Мы можем или сделать это через contextmenu (правый клип), который мы покроем позже, или мы можем сделать это через инструментальную панель. Давайте идти далее и займемся построением этой инструментальной панели :

Мы просто должны добавить это к нашему конструктору EditorGrid :

```
tbar: [{  
    text: 'Add a president',  
    tooltip: 'Great Tooltip',  
    iconCls: 'add', // this is defined in our styles.css  
    handler: displayFormWindow  
}]
```

Последний штрих

Здорово, я никогда не видел Вас настолько нетерпеливыми, в предвкушение испытать эту вещь... Хорошо, но мы еще не полностью сделали. Нам нужно чтобы сервер мог обработать запрос CREATEPRES... Давайте впишем это в database.php файл :

Сначала выключатель:

```
case "CREATEPRES":  
    createPresident();  
    break;
```

Затем функция:

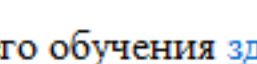
```
function createPresident(){  
  
    $firstname = addslashes($_POST['firstname']);  
    $lastname = addslashes($_POST['lastname']);  
    $enteringoffice = $_POST['enteringoffice'];  
    $leavingoffice = $_POST['leavingoffice'];  
    $income = $_POST['income'];  
    $party = $_POST['party'];  
  
    // Here we should probably do some database checking,  
    // to make sure that we do not have the same entry  
    // twice for ex... And we would return a different  
    // error code (ECHO '0' or whatever you want...)  
    // For now we'll pretend like the entry is valid.  
    $query = "INSERT INTO presidents ('IDpresident'  
    , 'IDparty', 'firstname', 'lastname', 'tookoffice'  
    , 'leftoffice', 'income' ) VALUES (NULL , '$party'  
    , '$firstname', '$lastname', '$enteringoffice'  
    , '$leavingoffice', '$income')";  
    $result = mysql_query($query);  
    echo '1';  
}
```

Вот так то лучше, теперь вы можете добавлять президентов в свою таблицу!

Что стоит запомнить из этого раздела...

- SimpleStore story

Вы можете увидеть результаты Части 5 нашего обучения [здесь](#).

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