

[Обучение Ext JS] :: Часть 6. Редактор таблиц

Фев 22 2010

В предыдущей главе мы узнали как отображать данные в структурированной таблице с которой пользователь может взаимодействовать. Но у них есть одно очень важное ограничение. Пользователь не мог редактировать данные. К счастью, в Ext существует EditorGridPanel, позволяющий использовать изменять поле формы. Вот про это и пойдет речь этой нашей главе. По принципу работы EditorGridPanel во многом похоже на Excel, позволяя пользователю выбирать ячейку и изменять ее содержимое.

В этой главе мы научимся:

- Предоставлять пользователю возможность редактирования таблицы соединенной с хранилищем данных
- отправлять измененные данные на сервер, позволяя пользователям обновлять серверные базы данных используя редактор таблиц Ext JS
- Управлять таблицей при помощи программного кода и реагировать на события
- Совершать продвинутое форматирование и более мощные редакторы таблицы

Но сначала, давайте посмотрим что можно сделать с редактируемой таблицей.
Что же я могу сделать с редактируемой таблицей?
EditorGridPanel очень похож на формы, с которыми мы работали ранее. Фактически, редактор таблицы использует те же самые поля формы, что и наша форма. Используя поля формы для выполнения редактирования ячеек таблицы, мы получим ту же функциональность, что дает поле формы. Включая ограниченный ввод и валидацию значений. Совместив все это GridPanel, у нас получается приложение способное выполнить практически все, что мы пожелаем.

Movie Database

Add Movie

Remove Movie

Title	Director	Released	Genre	Tagline
Office Space	Mike Judge	02/19/1999	Comedy	Work Sucks
Super Troopers	Jay Chandrasekhar	02/15/2002	Comedy	Altered State Police
American Beauty	Sam Mendes	10/01/1999	Drama	... Look Closer
The Big Lebowski	Joel Coen	03/06/1998	Comedy	The "Dude"
Fight Club	David Fincher	10/15/1999	Action	How much can you

Все поля в этой таблице могут быть изменены инлайн при использовании таких полей формы, как поля текстов, комбинированные окна и поле для захвата даты.

Работая с редактируемыми таблицами
Начнем с того, что превратим не редактируемую таблицу в редактируемую очень просто. Сложности начинаются тогда нам надо создать процесс обрабатывающий правки и отправляющий данные на сервер. Но, как только вы научитесь, сложности это больше не составит
Давайте изменим код, созданный в начале пятой главы, и добавим возможность изменять название, имя режиссера и строку подзаголовка. Вот так будет выглядеть измененный код:

Исходный код примера:

Исходный код примера:

```
var title_edit = new Ext.form.TextField();
var director_edit = new Ext.form.TextField({vtype: 'name'});
var tagline_edit = new Ext.form.TextField({
    maxLength: 45
});

var grid = new Ext.grid.EditorGridPanel({
    renderTo: document.body,
    frame: true,
    title: 'Movie Database',
    height: 200,
    width: 520,
    clicksToEdit: 1,
    store: store,
    columns: [
        {header: "Title", dataIndex: 'title',
            editor: title_edit},
        {header: "Director", dataIndex: 'director',
            editor: director_edit},
        {header: "Released", dataIndex: 'released',
            renderer: Ext.util.Format.dateRenderer('m/d/Y')},
        {header: "Genre", dataIndex: 'genre',
            renderer: genre_name},
        {header: "Tagline", dataIndex: 'tagline',
            editor: tagline_edit}
    ]
});
```

Необходимо было сделать четыре вещи для превращения нашей таблицы в редактируемую:

- Определение таблицы меняется с Ext.grid.GridPanel на Ext.grid.EditorGridPanel
- Мы добавили clicksToEdit в настройки таблицы. Это не обязательно, но выставляется двумя щелчками мышки.
- Создали поле формы для каждого столбца который мы захотим изменить.
- Отправили поля формы в нашу модель столбца через редактор настроек

Редактор может быть любым из существующих типов формы поля Ext JS, или созданный вами (пользовательский). Мы начнем с создания поля ввода текста которое будет использоваться для редактирования поля названия фильма

```
var title_edit = new Ext.form.TextField();
```

Затем добавим эту форму в модель столбца как редактор:

```
{header: "Title", dataIndex: 'title', editor: title_edit}
```

Следующим шагом будет изменение компонента GridPanel используя EditorGridPanel и добавить настройку clicksToEdit:

Исходный код примера:

Исходный код примера:

```
var grid = new Ext.grid.EditorGridPanel({
    renderTo: document.body,
    frame: true,
    title: 'Movie Database',
    height: 200,
    width: 520,
    clicksToEdit: 1,
    // removed extra code for clarity
});
```

Все это превращает нашу таблицу в редактируемую. Можно выбрать любое поле для которого мы прописали редакторы и менять значения.

Movie Database

Title	Director	Released	Genre	Tagline
Office Space The Musical	Mike Judge	02/19/1999	Comedy	Work Sucks
Super Troopers The Musical	Jay Chandrasekhar	02/15/2002	Comedy	Altered State Police
American Beauty The Musical	Sam Mendes	10/01/1999	Drama	... Look Closer
The Big Lebowski	Joel Coen	03/06/1998	Comedy	The "Dude"
Fight Club	David Fincher	10/15/1999	Action	How much can you kn

Как мы видим, здесь внесены некоторые изменения в названия некоторых фильмов, превративших их в мюзиклы. Редактор включается одним щелчком по ячейке. Нажатие клавиш Enter, Tab или щелчок мышью за пределами редактируемой ячейки сохранит сделанные изменения.
Маленькая красная галочка, появляющаяся в верхнем левом углу, показывает то, что ячейка "грязная". Что это такое мы рассмотрим через секунду. Сначала, давайте сделаем более сложные редактируемые ячейки

Изменение большего количества ячеек
Наши редактируемую таблицу мы начали с того, что сделали редактируемым один столбец. Для того чтобы настроить редактор мы создали ссылку на поле формы:

```
var title_edit = new Ext.form.TextField();
```

Затем мы использовали это поле как редактор столбца:

```
{header: "Title", dataIndex: 'title', editor: title_edit}
```

Это основные требования для каждого поля. Теперь немного расширим эти знания.

Изменение большего количества полей
Сейчас нам предстоит создать редактор для других полей. Различные типы информации могут потребовать некоторых уточнений, создательного того, что необходимо для этих полей.
Любое поле формы может быть использовано как редактор. Некоторые стандартные типы:

- TextField
- NumberField
- ComboBox
- DateField
- TimeField
- CheckBox

Эти редакторы могут быть расширены для достижения особых настроек, но сейчас, давайте начнем с полей которые есть в нашей таблице — Дата выхода в прокат и жанра.

Изменение значения данных
Для редактирования столбца даты выхода в прокат прекрасно подойдет DateField. Сначала, нам надо настроить поле редактора и указать какой формат использовать:

```
release_edit = new Ext.form.DateField({
    format: 'm/d/Y'
});
```

Затем, мы применяем редактор к столбцу вместе с тем рендерером, что использовался ранее:

```
{header: "Released", dataIndex: 'released', renderer:
    Ext.util.Format.dateRenderer('m/d/Y'), editor: release_edit}
```

Movie Database

Title	Director	Released	Genre	Tagline
Office Space	Mike Judge	02/19/1999	Comedy	Work Sucks
Super Troopers	Jay Chandrasekhar	02/15/2002	Comedy	Altered State Police
American Beauty	Sam Mendes	<div>February 2002</div>	...	Look Closer
The Big Lebowski	Joel Coen		The "Dude"	
Fight Club	David Fincher		How much can you	

Этот столбец получает очевидные преимущество от сосуществования рендерера и редактора. Как только по щелчку запускается редактор, рендерер отправляет рендеринг поля редактору. И наоборот. Таким образом, по окончании редактирования, поле будет в нужном нам формате.

Редактирование при помощи ComboBox
Давайте настроим редактор столбца жанра. На роль редактора, который будет выводить нам выпадающее меню с доступными жанрами, прекрасно подходит комбинированное окно.

Исходный код примера:

Исходный код примера:

```
var genre_edit = new Ext.form.ComboBox({
    typeAhead: true,
    triggerAction: 'all',
    mode: 'local',
    store: genres,
    displayField: 'genre',
    valueField: 'id'
});

{header: "Genre", dataIndex: 'genre', renderer: genre_name,
    editor: genre_edit}
```

Мы закончили с редактируемыми полями у которых были фиксированный набор настроек.

Movie Database

Title	Director	Released	Genre	Tagline
Office Space	Mike Judge	02/19/1999	Comedy	Work Sucks
Super Troopers	Jay Chandrasekhar	02/15/2002	Comedy	Altered State Police
American Beauty	Sam Mendes	10/01/1999	New Genre	... Look Closer
The Big Lebowski	Joel Coen	03/06/1998	Comedy	The "Dude"
Fight Club	David Fincher	10/15/1999	Drama	How much can you

Реакция на изменение ячейки
Конечно, теперь нам надо понять как сохранить все эти изменения. Я уверен, что пользователь не будет доволен, если мы выбросим все изменения. Процесс сохранения мы начнем со списка событий и реакций на них нашего собственного обработчика. Прежде чем мы начнем кодить, нам надо понять как работает редактор таблицы.

Что такое "грязная" ячейка?
Поле которое было изменено, считается "грязным", до тех пор, пока хранилище не скажет другое. Это "грязное" значение было сохранено во временное хранилище, содержащее версию наших данных наравне со всеми сделанными изменениями. Настоящее хранилище не изменится.
Мы можем сохранить изменения в основном хранилище, вызвав функцию commit, или же, можно отменить все изменения вызвав обработчик reject. Эти обработчики можно вызывать для всей таблицы, отдельной ячейки или для результата происходящего события.

Давайте представим что e это объект события редактирования. Мы можем отменить запись вызвав обработчик reject

```
e.record.reject();
```

Или сохранить изменения:

```
e.record.commit();
```

Реакция на изменения
Для сохранения наших данных в хранилище мы подождем пока не закончится редактирование случая на предмет пост-редакционного события.
Добавление необходимого нам слушателя на панель таблицы:

Исходный код примера:

Исходный код примера:

```
var grid = new Ext.grid.EditorGridPanel({
    // more config options clipped //,
    title: 'Movie Database',
    store: store,
    columns: // column model clipped //,
    listeners: {
        afteredit: function(e){
            if (e.field === 'director' && e.value === 'Mel Gibson'){
                Ext.Msg.alert('Error', 'Mel Gibson movies not allowed');
                e.record.reject();
            } else {
                e.record.commit();
            }
        }
    }
});
```

Как и в случае с другими слушателями в Ext, слушателю редактора таблицы дана функция выполняться когда происходит событие. Функция на пост-редакцию вызывается одним аргументом: объектом с набором полезных свойств. Мы можем использовать эти свойства для принятия решений по только что сделанным изменениям.

Property	Property
grid	Таблица в которой происходят изменения
record	Вся запись которая подверглась изменению. Значения других столбцов могут быть возвращены при использовании свойств их объектов 'data'
field	Название редактируемого столбца
value	Строка содержащая новое значение ячейки
originalValue	Строка содержащая старое (не измененное) значение ячейки
row	Индекс редактируемого ряда
column	Индекс редактируемого столбца

Например, если бы мы хотим чтобы фильмы Мэла Гибсона никогда не попали в нашу базу, можно сделать простую проверку на этот случай:

```
if (e.field === 'director' && e.value === 'Mel Gibson'){
    Ext.Msg.alert('Error', 'Mel Gibson movies not allowed');
    e.record.reject();
} else {
    e.record.commit();
}
```

Сначала, мы проверяем было ли изменено поле режиссера, затем, мы убеждаемся что новое значение не равно Мэлу Гибсону. Если оба события проходят проверку, мы возвращаем запись в хранилище. Это значит, что как только мы вызовем обработчик наше основное хранилище пополнится новой записью.

Мы так же можем отклонять изменения.

Конечно все, что до этого мы сделали для обновления данных, хранится в памяти браузера. Я уверен что вы просто умоляете от желания обновить сервер. К этому мы перейдем довольно скоро.

- 1
- 2
- [следующая >](#)
- [последняя >](#)