

.. / 8.3 ExtJS: GridPanel — табличка с редактированием полей

<ExtJS-4>

Как правило, в богатых приложениях таблицы используются не столько для отображения, сколько для управления данными - удаления, добавления, редактирования записей

В качестве основы здесь как и всюду я взял пример с официального сайта, только немного сократил. Сейчас я попробую внести возможность редактирования данных на базе примера из предыдущей части, где выводился телефонный справочник. Код практически весь сохраняется, модификация не очень большая. Что меняется?

1. Вводится модель

Первое и самое главное - вводится **Модель**. Модель (подробности будут отдельно) - это компонент, введенный только в четвертой версии ExtJS. По существу это объект определяющий записи - набор полей с описанием, **валидаторы**, **прокси** для чтения-записи данных. А так же **ассоциации** позволяющие связывать разнородные модели (например модель товара с моделью поставщика).

```
1 Ext.define('Phones', {
2     extend: 'Ext.data.Model',
3     fields: ['id', 'city', 'code'],           // список
        полей
4     validations: [{ // валидаторы
5         type: 'length',
6         field: 'city', // поле город должно быть не пустым
7         min: 1
8     }]
9 });
```

2. Изменения в хранилище/прокси

Соответственно меняется и хранилище. Если в предыдущих примерах поля указывались при описании хранилища, здесь вместо описания полей идет указание модели. Появились и другие изменения:

[client side XSLT](#) (3)[CSS](#) (2)[ExtJS 4. С самого начала](#) (26)[Ext JS «Большое приложение»](#) (8)[Ext JS баги и траблы](#) (2)[FCKEditor](#) (5)[HTML5](#) (1)[jQuery](#) (6)[livejournal](#) (3)[SEO](#) (6)[WYSIWYG](#) (7)[xslt:группировки](#) (3)[xslt:метод мюнха](#) (2)[xslt:примеры](#) (8)[xslt:рекурсия](#) (4)[Краткий курс XSLT](#) (1)[Codeigniter Шаг за Шагом](#) (8)

```

01 var store = Ext.create('Ext.data.Store', {
02     autoLoad: true,
03     autoSync: true, // указание на автосохранения в
хранилище
04                     // при каждом изменении одно записи
05     model: 'Phones', // указание модели созданной выше
06     proxy: {        // описание прокси
07         type: 'rest', // способ взаимодействия с
сервером
08         url: '630.php', // адрес серверного обработчика
09         reader: {
10             type: 'json',
11             root: 'data'
12         },
13         writer: {
14             type: 'json',
15             root: 'data'
16         },
17     }
18 });

```

А теперь подробнее разберу что осталось за комментариями.

- **rest** - такой хитрый способ взаимодействия с сервером по http-протоколу. Он касается только серверной части.

Останавливаться на этом не буду, подробности реализации на php можно найти по ссылкам приведенным в конце статьи. Я бы предпочел иметь обычный GET/POST, но как настроить в ExtJS4 прокси на отправку данных по-другому в этом примере я не нашел. Сделать по аналогии с предыдущим примером у меня не получилось.

- **reader и writer** - читалка и писалка. И то и другое - объекты для обмена данными с сервера.

type: 'json' - тип данных при обмене с сервером.

root: 'data' - объект в json в котором передаются записи (кроме собственно данных передается признак успешного завершения success, и всякая прочая балда опционально)

Чтобы было понятно, приведу пример json-данных которыми происходит обмен с сервером:

```

1  { 'success': 'true', // - признак нормального завершения
2    'message': 'бла-бла',
3    ..... // что-то еще
4    'data': [{ 'id': 1, city: 'Бабруйск', code: '12-413' }...] //
данные.
5  }

```

3. Вводится плагин для редактирования строк

Вообще-то есть два способа редактирования - один редактирования отдельного поля записи (ячейки таблицы) другой - всей записи целиком. Я использую последний способ. Для того чтобы подключить этот плагин необходимо его проинициализировать:

```

1 var Editing = Ext.create('Ext.grid.plugin.RowEditing');

```

Инициализация таблицы:

Дальше следует инициализация самой таблицы с дополнительными компонентами:

```
01  var grid = Ext.create('Ext.grid.Panel', {
02      plugins: [Editing], // указали плагин для
редактирования
03      store: store,
04      ... // всякие общие вещи - размер, id, title
05      columns: [{ // описание колонок
06          text: 'id',
07          dataIndex: 'id' // привязка к индексу модели
08      }, {
09          text: 'City',
10          dataIndex: 'city',
11          field: { //!!! теперь это не текст.
12              xtype: 'textfield' //!!! теперь это input
13          }
14      }, {
15          header: 'Code',
16          dataIndex: 'code',
17          field: { //!!! теперь это не текст.
18              xtype: 'textfield' //!!! теперь это input
19          }
20      }
21      ],
22      bbar: new Ext.PagingToolbar({ // bbar - нижний
тулбар с листалкой
23          store: store, // указано
хранилище
24          displayInfo: true, // вывести инфо
обо общем числе записей
25          displayMsg: 'Показано {0} - {1} из {2}' //
формат инфо
26      })),
27      tbar: [ // верхний тулбар с кнопками удаления и
вставки
28          {
29              text: 'Ins',
30              handler: function(){
31                  // при вставке создается пустая запись в
в хранилище.
32                  store.insert(0, new Phones());
33                  // переход к редактированию новой строки
Editing.startEdit(0, 0);
34              }
35          },
36          {
37              itemId: 'delete', // id кнопки, потом
пригодится
38              text: 'Del',
39              handler: function(){
40                  // определяем какая строка выделена
41                  var selection =
grid.getView().getSelectionModel().getSelection()[0];
42                  // удаление строки из хранилища
43                  if (selection) {store.remove(selection);}
44              }
45          }
46      ]
47  });
```

В результате получается следующее:

Phones			
Ins Del			
id	City	Code	
59	Калтасы	347-79	
58	Калманка	38551	
57	Казаченское	391-16	
56	Ишимбай	347-94	
55	Исянгулово	347-15	
54	Ирбейское	391-74	
Page 1 of 3 Показано 1 - 25 из 55			

Далее

1. [GridPanel - таблички. Часть 3 Редактирование полей \(продолжение\)](#)
2. [Form. Часть 1 - простые компооненты](#)
3. [Form. Часть 2 - составные компоненты](#)
4. [Form. Часть 3 - Submit](#)
5. [Деревья в ExtJS. Часть 1. Инициализация](#)

2011-12-04

<ExtJS-4>

См. также

1. [RESTful PHP — 5 простых советов](#)
2. [ExtJS4 The Data Package \(перевод официальной документации\)](#)
3. [Подход REST в PHP](#)
4. [sencha.com: Learn The Grid Component](#)
5. [Working with RESTful Services in CodeIgniter](#)

А потрындеть?

04.03.12 Сухроб аллә нәрсә әйтте:

Почему редактирование не работает ??

[Ответить](#)

[Оставить сообщение](#)

Имя*:

Хоумпейдж:

Сообщение*:

15 ↗