

# CRUD in extjs 3.0

A quick look at the new ext.data.  
Writer in 3.0

# Agenda

Today I will be sharing notes from my early experiments with the ext.data.Writer from extjs 3.0.

The code I did while looking at it this is available at :

<http://github.com/jacobandresen/yase/tree/master>

(find a version done around 2009-06-17 )

# Manipulating relational data

What we usually want to do with data

- Create INSERT
- Read / Retrieve SELECT
- Update UPDATE
- Delete / Destroy DELETE

So far we have only been able to READ data out of the box.  
the other operations needed work.

Now extjs 3.0 introduces the ext.data.writer. It can do the rest.

# Introducing the writer

We declare the writer like this

```
var settingsWriter = new Ext.data.JsonWriter({  
    returnJson: true,  
    writeAllFields: true  
});
```

where `writeAllFields` identifies that we want to write all the fields from the record to the database. If you have a fancy ORM then maybe you can set this to false.

# using a Lightweight MVC/REST architecture

We can now define a "settings resource" in the following manner on the proxy:

```
var settingsProxy = new Ext.data.HttpProxy({
  api: {
    read    : 'app.php/settings/view',
    create  : 'app.php/settings/create',
    update  : 'app.php/settings/update',
    destroy : 'app.php/settings/destroy'
  }
});
```

Note that view here is the same as "Read/Retrieve"!

# Hooking up the datastore

The Writer and the proxy can be hooked to the store like this

```
var settingsStore = new Ext.data.Store({  
    id: 'setting',  
    proxy: settingsProxy,  
    reader: settingsReader,  
    writer: settingsWriter,  
    ...  
});
```

# relaying information from components

When submitting data using the Ext.writer from components it can be useful to have centralized validations running in a central place. Let's say that we have a SettingsGrid - we can relay the save and update events like this :

..

```
initComponent : function() {  
    this.viewConfig = {  
        forceFit: true  
    };  
    this.relayEvents(this.store, ['destroy', 'save', 'update']);  
    this.tbar = this.buildTopToolBar();  
    SettingsGrid.superclass.initComponent.call(this);  
},
```

# looking at the restful examples

The extjs 3.0 distro supplies the following examples (and more!):

- `examples/restful`
- `examples/writer`

These examples outline a MVC framework that can map actions and path information to simple REST resources.

Steven Hiller has provided an example with Ruby on Rails :

[http://www.extjswithrails.com/2009/06/restful-store-example-with-rails\\_04.html](http://www.extjswithrails.com/2009/06/restful-store-example-with-rails_04.html)

# Looking at the php MVC framework

The framework consists of the following classes:

- Model (contains a simulated ORM in the examples)
- Request (parses the HTTP request to parameters)
- Controller (dispatches the action known for the path)
- Response (renders a suitable view (JSON) )

The framework is operated from a single point of entry that uses dynamic class loading to dispatch actions through the controller identified in the url from HTTP request.

# Following updated data from the grid to the server

When the user alters a value in the grid, then a "save" event occurs (if autosave is true).

Upon the "save" event the grid determines which cells has been altered. When we have an altered cell , then the corresponding record is sent to the server with the 'root' from the reader around it. E.g if we read with root "data", then we send back with root "data".

We can have several records being sent at once. when updating to the server (e.g multiple edits).

# Questions ?

[jacob.andresen@gmail.com](mailto:jacob.andresen@gmail.com)