Learning ExtJS

Ext JS is a pure JavaScript application framework for building interactive web applications[1] using techniques such as Ajax, DHTMLand DOM scripting.

Current stable version is Extjs 5.1.1

Download

Link: http://www.sencha.com/legal/GPL/
Enter email and select extjs
Download extjs using download link sent to email.

Installation

1. Extract zip to /var/www/myapp/ directory
2. Rename /var/www/myapp/ext-* directory to /var/www/myapp/extjs (just for simplicity)

Application Structure

```
/var/www/myapp/
        index.html          => main web page
        app.js              => Initialize application
        app/
                model/              => Stores the model (defines structure of data)
                store/              => Stores the result store (stores actual data)
                view/               => Stores the views (presentation)
                controller/         => Stores the controllers (user interactions)
```

Writing first application

1. vim /var/www/myapp/index.html

```
<!DOCTYPE>
<html>
<title>Mailout Spam Monitor</title>
<head>
        <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no">
```

```
    <link rel='stylesheet' href='extjs/build/packages/ext-theme-neptune/build/resources/ext-theme-neptune-all-debug.css'> // Load theme
    <script type='text/javascript' src='extjs/build/ext-all-debug.js'></script>  // Load extjs file
    <script type='text/javascript' src='extjs/build/packages/ext-theme-neptune/build/ext-theme-neptune-debug.js'></script>  // Load extjs file for managing theme
    <script type='text/javascript' src='app.js'></script>  // Load our main  app class

</head>

<body>

</body>
</html>
```

2. vim /var/www/myapp/app.js

```
Ext.application( {
    name: 'SpamMonitor',                    // Name of application

    controllers: ['SpamMonitor.controller.Main'],       // define controller

    launch: function() {
        console.log('launched');                        // launch function is called on first load
                                                        // of page
        Ext.create('SpamMonitor.view.Main');            // This actually creates the main web
                                                        // page
    }

});
```

3. mkdir /var/www/myapp/controller

4. vim /var/www/myapp/controller/Main.js

```
// Defines the controller, here we are specifying the views, models and stores for our app.
Ext.define('SpamMonitor.controller.Main', {

    extend: 'Ext.app.Controller',

    views: ['SpamMonitor.view.Main'],
     models: ['SpamMonitor.model.Main'],
     stores: ['SpamMonitor.store.Main'],


});
```

5. vim /var/www/myapp/view/Main.js

```
// This view will simply create a panel and show a grid (table) containing domain and count
//columns.
Ext.define('SpamMonitor.view.Main', {
    extend: 'Ext.grid.Panel',           // extend the base class of extjs to create a Panel

    store: 'SpamMonitor.store.Main',    // This specifies the store for this view
    renderTo: Ext.getBody(),            // This binds the view to the entire web page <body>

    columns: [                                  // This specifies the structure of store
        {text: 'Domain', dataIndex: 'domain'},
        {text: 'Count', dataIndex: 'count'}
    ]
});
```

6. vim /var/www/myapp/app/model/Main.js

```
// Create a model defining the structure of store, it is made compulsory in extjs5 though we have
//already defined in view.

Ext.define('SpamMonitor.model.Main', {

    extend: 'Ext.data.Model',
    fields: [
        {name: 'domain', type: 'string'},
        {name: 'count', type: 'int'}
    ]
});
```

7. vim /var/www/myapp/store/Main.js

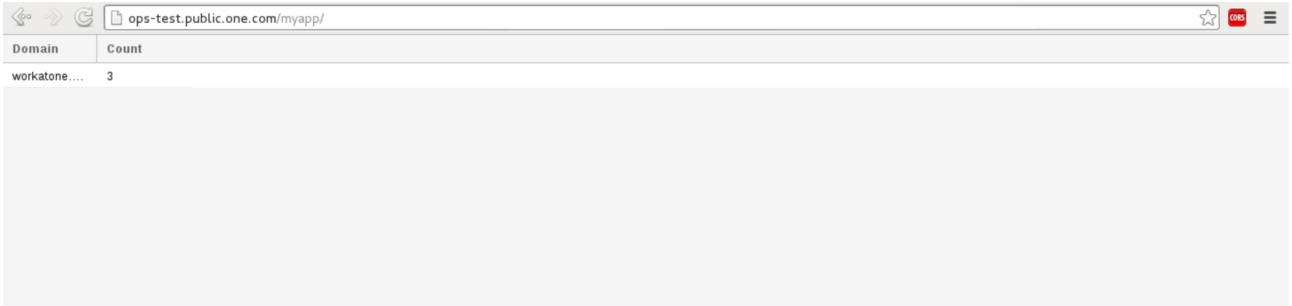
```
// This is the store that will fetch data from a PHP script, we can have local data as well.

Ext.define('SpamMonitor.store.Main', {

    extend: 'Ext.data.Store',
    model: 'SpamMonitor.model.Main',
    proxy: {
        type: 'ajax',           // We are using ajax call to load data from a php script.
        url: 'fetch-data.php',  //
    },
  autoLoad: true                // This will load data automatically on web loading time.
});
```

8. vim /var/www/myapp/fetch-data.php

```
// Sample scripting

<?php

$data = array('domain' => 'workatone.com', 'count' => '3');

echo json_encode($data);

?>
```

Load the application

http://localhost/myapp



ops-test.public.one.com/myapp/

Domain	Count
workatone....	3