

目录

1	大概思路.....	1
2	设计数据库.....	1
3	编写递归树.....	2
4	EXTJS 注册若干通用组件.....	3
5	EXTJS 类图概览与代码.....	7
6	JSON 序列化.....	11
7	运行效果.....	12
8	总结.....	14

1 大概思路

- 设计数据库
- 编写递归树
- EXTJS 注册若干通用组件与代码
- EXTJS 类图概览
- JSON 序列化
- 运行效果

2 设计数据库

详细查看 db_script

SYSTEM_MODULE		
SM_ID	NUMBER	<pk>
SM_PARENTID	NUMBER	
SM_NAME	VARCHAR(30)	
SM_DESCRIPTION	VARCHAR(50)	
SM_REMARK	VARCHAR(300)	
SM_EXPANDED	VARCHAR(1)	
SM_LEAF	VARCHAR(1)	
UPDATE_DATE	DATE	
UPDATE_BY	VARCHAR(30)	
CREATE_DATE	DATE	
CREATE_BY	VARCHAR(30)	

```

/*=====*/
/* Table: SYSTEM_MODULE */
/*=====*/
create table SYSTEM_MODULE
(
  SM_ID          NUMBER          not null,
  SM_PARENTID   NUMBER,
  SM_NAME       VARCHAR(30),
  SM_DESCRIPTION VARCHAR(50),
  SM_REMARK     VARCHAR(300),
  SM_EXPANDED   VARCHAR(1)      default '0' not null,
  SM_LEAF       VARCHAR(1)      default '0',
  UPDATE_DATE   DATE,
  UPDATE_BY     VARCHAR(30),
  CREATE_DATE   DATE,
  CREATE_BY     VARCHAR(30),
  constraint PK_SYSTEM_MODULE primary key (SM_ID)
);
comment on table SYSTEM_MODULE is

```

```

'模块表';
comment on column SYSTEM_MODULE.SM_ID is
'当前节点';
comment on column SYSTEM_MODULE.SM_PARENTID is
'父节点';
comment on column SYSTEM_MODULE.SM_NAME is
'模块名称';
comment on column SYSTEM_MODULE.SM_DESCRIPTION is
'描述或访问页面、对象';
comment on column SYSTEM_MODULE.SM_REMARK is
'备注';
comment on column SYSTEM_MODULE.SM_EXPANDED is
'0 不展开,1 展开';
comment on column SYSTEM_MODULE.SM_LEAF is
'0 不是子节点 ,1 是子节点';
comment on column SYSTEM_MODULE.UPDATE_DATE is
'更新时间';
comment on column SYSTEM_MODULE.UPDATE_BY is
'更新人';
comment on column SYSTEM_MODULE.CREATE_DATE is
'创建时间';
comment on column SYSTEM_MODULE.CREATE_BY is
'创建人';
create sequence SEQ_SYSTEM_MODULE
increment by 1
start with 1
  nomaxvalue
  nocycle;
/* 若有数据 */
INSERT      INTO      system_module(sm_id,SM_PARENTID,sm_name,SM_EXPANDED)
VALUES(seq_system_module.nextval,'0','产品一','1');

```

3 编写递归树

```

/// <summary>
/// 获取树
/// </summary>
/// <param name="obj"></param>
/// <returns></returns>
public SYSTEM_TREE GetTree(SYSTEM_TREE obj)
{
    try

```

```
{
    // 查询子节点
    string strNextSQL = @"SELECT SM.SM_ID as id, SM.SM_PARENTID as parentid,
SM.SM_NAME as text, SM.SM_EXPANDED as expanded
        FROM SYSTEM_MODULE SM
        WHERE SM.SM_PARENTID = :P_SM_PARENTID
        ORDER BY SM.SM_ID";
    OracleParameter[] parasNext = new OracleParameter[1];
    parasNext[0] = new OracleParameter("P_SM_PARENTID",
OracleType.Number);
    parasNext[0].Value = obj.id;
    DataView dvNext = OracleHelper.ExecuteView(this.connectString,
System.Data.CommandType.Text, strNextSQL, parasNext);
    DataTable dtNext = dvNext.Table;

    obj.children = new List<SYSTEM_TREE>();
    // 子节点添加到当前节点
    foreach (DataRow dr in dtNext.Rows)
    {
        SYSTEM_TREE st = new SYSTEM_TREE();
        st.id = Convert.ToInt32(dr["id"].ToString());
        st.text = dr["text"].ToString();
        st.expanded = dr["expanded"].ToString() == "0" ? false : true;

        // 递归调用
        st = this.GetTree(st);

        st.leaf = st.children.Count > 0 ? false : true;
        obj.children.Add(st);
    }
    return obj;
}
catch
{
    throw;
}
}
```

4 EXTJS 注册若干通用组件

ux.js 文件

```
Ext.ns('Pub.ux');

/*
create by zhyongfeng in 2013.05.23
注册 RadioGroup 扩展组件
new Pub.ux.RadioGroup({
.....
})
*/
Pub.ux.RadioGroup = Ext.extend(Ext.form.RadioGroup, {
    getValue : function () {
        var v;
        if (this.rendered) {
            this.items.each(function (item) {
                if (!item.getValue())
                    return true;
                v = item.getRawValue();
                return false;
            });
        } else {
            for (var k in this.items) {
                if (this.items[k].checked) {
                    v = this.items[k].inputValue;
                    break;
                }
            }
        }
        return v;
    },
    setValue : function (v) {
        if (this.rendered)
            this.items.each(function (item) {
                item.setValue(item.getRawValue() == v);
            });
        else {
            for (var k in this.items) {
                this.items[k].checked = this.items[k].inputValue == v;
            }
        }
    }
});
Ext.reg('ux.RadioGroup', Pub.ux.RadioGroup);
```

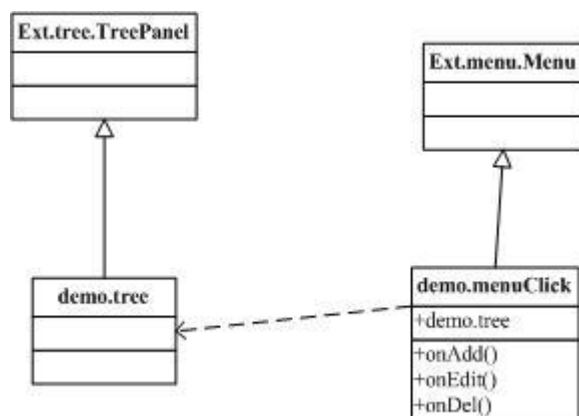
```
/*
create by zhyongfeng in 2013.05.23
注册 FormPanel 扩展组件
new Pub.ux.FormPanel({
.....
})
*/
Pub.ux.FormPanel = Ext.extend(Ext.FormPanel, {
    frame : true,
    layout : 'form',
    border : false,
    lableWidth : 20,
    constructor : function (config) {
        // 对象不存在,则返回
        if (!config) {
            Pub.ux.FormPanel.superclass.constructor.apply(this);
            return;
        }

        Ext.apply(this, config);
        Pub.ux.FormPanel.superclass.constructor.apply(this);
    }
});
Ext.reg('ux.FormPanel',Pub.ux.FormPanel);

/*
create by zhyongfeng in 2013.05.23
注册 Window 扩展组件
new Pub.ux.Window({
.....
})
*/
Pub.ux.Window = Ext.extend(Ext.Window, {
    frame : true,
    border : false,
    autoDestroy : true,
    modal:true,
    resizable : false,
    layout:'fit',
    buttonAlign : "center",
    width : 250,
    height : 150,
    constructor : function (config) {
        // 对象不存在,则返回
```

```
if (!config) {
    Pub.ux.Window.superclass.constructor.apply(this);
    return;
}
this.buttons = [{
    text : "提交",
    scope : this,
    handler : this.onSave
}, {
    text : "退出",
    scope : this,
    handler : this.close
}
];
Ext.apply(this, config);
Pub.ux.Window.superclass.constructor.apply(this);
},
onSave : function () {
    if (this.items.length == 0)
        return;
    // 针对表单
    var formPanel = this.getFormPanel();
    if (formPanel.getForm().isValid())
        formPanel.getForm().submit({
            url:formPanel.url,
            waitMsg : '正在保存...',
            success : function (re, v) {},
            failure : function () {
                JsHelper.ShowError("响应文本错误");
            }
        });
},
getFormPanel:function(){
    if (this.items.length == 0)
        return;
    return this.items.items[0];
}
})
Ext.reg('ux.Window',Pub.ux.Window);
```

5 EXTJS 类图概览与代码



```

Ext.ns('demo');
Ext.onReady(function () {
    Ext.QuickTips.init();
    demo.run();
});

demo.formItems = [{
    xtype : 'textfield',
    fieldLabel : 'ID',
    name : 'id',
    anchor : '100%',
    readOnly : true,
    style : {
        background : '#E6E6E6'
    }
}, {
    xtype : 'textfield',
    fieldLabel : '名称',
    name : 'parentid',
    anchor : '100%',
    allowBlank : false
}, {
    xtype : 'ux.RadioGroup',
    fieldLabel : '展开',
    name : 'expandedgroup',
    items : [{
        xtype : 'radio',
        name : "expanded",
        inputValue : true,
        boxLabel : "是",
        checked : true
    }
  ]
}];
  
```



```
        }, {
            xtype : 'radio',
            name : "expanded",
            inputValue : false,
            boxLabel : "否"
        }
    ]
}
];

/*
声明 menu
new demo.menuClick({
    treePanel : null,
    .....
})
*/
demo.menuClick = Ext.extend(Ext.menu.Menu, {
    constructor : function (config) {
        this.items = [{
            text : '添加节点',
            scope : this,
            handler : this.onNew
        }, {
            text : '编辑节点',
            scope : this,
            handler : this.onEdit
        }, {
            text : '删除节点',
            scope : this,
            handler : this.onDel
        }
    ];
    // 对象不存在,则返回
    if (!config) {
        demo.menuClick.superclass.constructor.apply(this);
        return;
    }
    Ext.apply(this, config);
    demo.menuClick.superclass.constructor.apply(this);
},
```

```
onNew : function () {
    var formPanel = new Pub.ux.FormPanel({
        url:'Handler.ashx',
        items:demo.formItems
    });
    var window = new Pub.ux.Window({
        height:300,
        width:300,
        title : '新增',
        items : [formPanel]
    });
    window.show();
},
onEdit : function () {
    var selectNode = this.treePanel.getSelectionModel().getSelectedNode();
    if (selectNode.id == 0) {
        JsHelper.ShowWarning("根节点不提供编辑");
        return;
    }

    var formPanel = new Pub.ux.FormPanel({
        url:'Handler.ashx',
        items:demo.formItems
    });
    var window = new Pub.ux.Window({
        title : '编辑',
        items : [formPanel]
    });
    // 加载数据
    var json = {
        "id" : selectNode.id,
        "parentid" : selectNode.text,
        "expandedgroup" : selectNode.expanded
    };
    formPanel.getForm().setValues(json);
    window.show();
},
onDel : function () {
    var selectNode = this.treePanel.getSelectionModel().getSelectedNode();
    if (selectNode.id == 0) {
        JsHelper.ShowWarning("根节点不提供删除");
        return;
    }
    JsHelper.OK("当前节点为" + selectNode.id + "<br />当前文本为" +
```

```
selectNode.text);
    }
});

demo.treePanel = Ext.extend(Ext.tree.TreePanel, {
    constructor : function (config) {
        // 默认加载
        Ext.apply(this, {
            enableDD : false,
            allowDrag : false,
            useArrows : false,
            lines : true,
            border : false,
            rootVisible : true,
            root : new Ext.tree.AsyncTreeNode({
                id : "0",
                text : "超市商场",
                expanded : true, //展开
                loader : new Ext.tree.TreeLoader({
                    url : "handler.ashx"
                })
            })
        })
    };
    Ext.apply(this, config);
    demo.treePanel.superclass.constructor.apply(this);
});

demo.run = function () {
    var treePanel = new demo.treePanel();
    var rightClick = new demo.menuClick({
        treePanel : treePanel
    });

    // 右键监听
    treePanel.on('contextmenu', function (node, event) {
        // 阻止浏览器默认右键菜单显示
        event.preventDefault();
        node.select();
        // 取得鼠标点击坐标,展示菜单
        rightClick.showAt(event.getXY());
    });

    var panelMain = new Ext.Panel({
```

```
        title : "系统管理",
        width : 300,
        height : 450,
        autoScroll : true,
        layout : 'fit',
        iconCls : "form-window",
        items : treePanel,
        collapsible : false
    });

    panelMain.render(document.body);
}
```

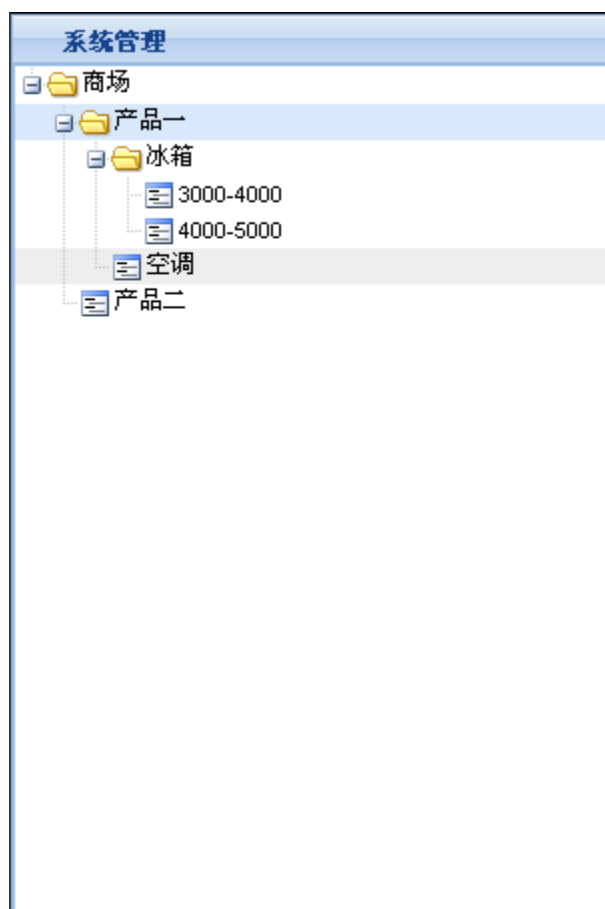
6 JSON 序列化

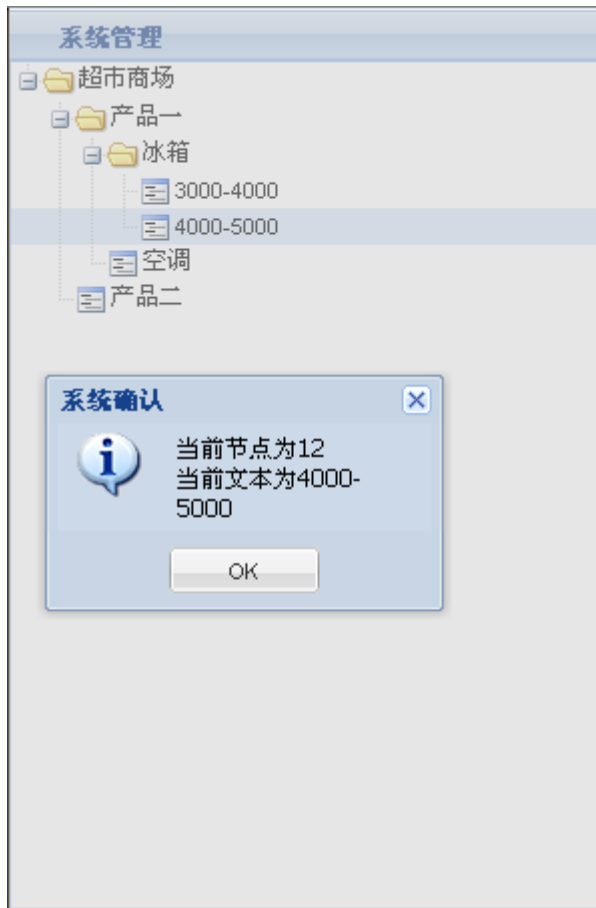
需要引入 `Newtonsoft.Json.dll`，进行 JSON 序列化。

```
using Newtonsoft.Json;
using Newtonsoft.Json.Converters;

SYSTEMMANAGER sysmanager = new SYSTEMMANAGER();
SYSTEM_TREE obj = new SYSTEM_TREE();
obj.id = int.Parse(context.Request["node"]);
obj = sysmanager.GetTree(obj);
string json = JsonConvert.SerializeObject(obj.children, Formatting.None, new
Newtonsoft.Json.Converters.IsoDateTimeConverter() { DateTimeFormat = "yyyy-MM-dd
HH:mm:ss" });
```

7 运行效果





8 总结

对于项目系统目录级数不限，可采用递归树的解决方案。

Newtonsoft.Json.dll 可对 DataTable 与 List<Object>等泛型进行序列化。

源代码下载：

http://files.cnblogs.com/yongfeng/EXTJS_TREE.rar