# datagrid组件

jsp

|  |
| --- |
| <table id=*"sysUserGrid"*></table>  **var** $grid = $('#sysUserGrid');  $grid.datagrid({  idField:'id' , //只要创建数据表格 就必须要加 ifField  fit:**true** ,//填充父容器  fitColumns:**true** , //自动展开/收缩列的大小，以适应网格的宽度，防止水平滚动。  striped: **true** , //隔行变色特性  loadMsg: '数据正在加载,请耐心的等待...' ,  rownumbers:**true** ,  pagination: **true** , //启用分页  //singleSelect:true , //单选模式  url:'${path}/sysUser/list',  columns:[[  {field:'ck', width:50, checkbox: **true**},  {field:'name',title:'名称'},  {field:'loginname',title:'登录名'},  {field:'pwd',title:'密码'},  {field:'sex',title:'性别'},  {field:'icon',title:'图标'},  {field:'phonenum',title:'手机号'},  {field:'phonenum2',title:'备用号'},  {field:'address',title:'地址'},  {field:'states',title:'用户状态'},  {field:'deptid',title:'部门id'},  {field:'createtime',title:'入职时间'},  {field:'leavetime',title:'离职时间'},  {field:'usertype',title:'用户类型'},  {field:'sotid',title:'排序号'},  ]] ,  }); toolbar:[{  text:'新 增' ,  iconCls:'icon-add' ,  handler:**function**(){  initAdd();  }  }] |

后台

|  |
| --- |
| @RequestMapping("/list")  @ResponseBody  **public** String list(HttpServletRequest request) {  Map<String, Object> map = **new** HashMap<String, Object>();  //easyUI的grid分页参数,具体处理在service层  map.put("pageNo", request.getParameter("page"));  map.put("pageSize", request.getParameter("rows"));  //查询域的查询条件  //map.put("deptid", request.getParameter("deptid"));  map.put("name", request.getParameter("name"));  PageBean<SysUser> list = sysUserService.select2PageBean(map);    **return** JsonUtil.*obj2JsonStr*(list);  }  **public** PageBean<SysUser> select2PageBean(Map<String, Object> map) {  PageBean<SysUser> pageBean = **new** PageBean<>();  pageBean.setPageNo(Integer.*parseInt*(map.get("pageNo").toString()));  pageBean.setPageSize(Integer.*parseInt*(map.get("pageSize").toString()));  //注意map要先设置pageBean,拦截器里面要获取其值  map.put("pageBean", pageBean);  map.put("needPage", **true**);//是否分页，默认是false不分页  pageBean.setRows(sysUserMapper.selectAll4Page(map));  **return** pageBean;  } |

由pageBean将需要返回的json封装了，实际只需的格式为：

{

"total":21,

"rows":[{"id":1,"loginname":"xm","name":"xiaoming","pwd":"123456","sex":"1","states":"1"},{"id":2,"loginname":"xm2","name":"xiaoming2","pwd":"123456","sex":"1","states":"1"}]

}

## grid的高级(其他)用法

见示例

冻结列（jsp和js2种实现）

## 注意事项

# tree组件

jsp

|  |
| --- |
|  |

后台

|  |
| --- |
|  |

## 注意事项

# combo组件

jsp

|  |
| --- |
|  |

后台

|  |
| --- |
|  |

## 注意事项

# combotree组件

jsp

|  |
| --- |
| <input id=*"pidCombotree"* name=*"pid"* />  **var** $combotree = $('#pidCombotree');  $combotree.combotree({  url:treeURL ,  required: **true** ,  missingMessage: '请选择上级部门',  onLoadSuccess : **function**(node, data){  **var** treeObj = $combotree.combotree('tree');  **var** rooNode = treeObj.tree('getRoot');  //console.log(rooNode);  //展开根节点  treeObj.tree('expand',rooNode.target);  //生效,但是是展开所有  //treeObj.tree('expandAll');  }  }); |

展开某个节点，先获得tree对象，然后用tree的方法来展开。

后台与tree组件实现相同

# treegrid组件

jsp

|  |
| --- |
| <table id=*"sysDeptTreeGrid"*></table>  **var** $treeGrid = $('#sysDeptTreeGrid');  $treeGrid.treegrid({  url:'${path}/sysDept/treeGrid',  idField:'deptid',  treeField:'name',  fit:**true** ,//填充父容器  rownumbers:**true** ,  //singleSelect:false , //true为单选模式 ,false为多选  columns:[[  {field:'ck', width:50, checkbox: **true**},  {field:'deptid',title:'部门id'},  {field:'name',title:'部门名称'},  {field:'pid',title:'父id'},  {field:'levels',title:'层级'},  {field:'createtime',title:'创建时间'},  {field:'description',title:'描述'},  {field:'sotid',title:'排序号'}  ]],  toolbar:[{  text:'新 增' ,  iconCls:'icon-add' ,  handler:**function**(){  initAdd();  }  },{  text:'刷新' ,  iconCls:'icon-reload' ,  handler:**function**(){  $treeGrid.treegrid('reload');  }  }],  onLoadSuccess : **function**(node, data){  **var** rooNode = $treeGrid.treegrid('getRoot');  //console.log(rooNode);  //展开根节点  $treeGrid.treegrid('expand',rooNode.deptid);//指定的是idField对应的值  //生效,但是是展开所有  //$treeGrid.treegrid('expandAll');  }  }); |

需要注意treeGrid展开节点与tree不同。

后台

|  |
| --- |
| @RequestMapping("/treeGrid")  @ResponseBody  **public** String treeGrid(HttpServletRequest request) {  Map<String, Object> map = **new** HashMap<String, Object>();  String pid = request.getParameter("id");  **if** (pid == **null**) {  pid = "0";  }  List<SysDept> list = sysDeptService.selectByPid(pid);  // //treegrid不适合做搜索查询  **return** JsonUtil.*obj2JsonStr*(sysDeptService.select2treeGrid(list));  }  //这里必须自己来判断state的值，从表直接取的值会出现bug  **public** List<SysDept> select2treeGrid(List<SysDept> deptList) {  **for** (SysDept dept : deptList) {  **if**(dept != **null**){  **if**(selectByPid(dept.getDeptid()).size() >0){  dept.setState("closed");  }  }  }  **return** deptList;  } |

# layout组件

jsp

|  |
| --- |
|  |

后台

|  |
| --- |
|  |

# href和iframe引入页面的区别

Href引入页面时，只解析body部分

iframe引入页面时，全部页面内容直接嵌入，如a页面引入了b页面，b页面用不了a页面的样式和js

# Formmatter的问题

return '<a href="javascript:void(0)" class="easyui-linkbutton" onclick="showDetail('+index+')">详情</a>'; 或者

return "<a href='javascript:void(0)' class='easyui-linkbutton' onclick='showDetail("+index+")'>详情</a>";

如果是string类型

return '<a href="javascript:void(0)" class="easyui-linkbutton" onclick="showDetail(\''+index+'\')">详情</a>';

当然搜全局了。formatter中的拼接index，只是在渲染dom的时候用一下而已，index未定义。