

# 树形控件详解

DNA专题交流会之一

保密

软件研究院 二零一二年

控件介绍

常用操作

常见问题

小结



树形控件提供以树形方式展示或操作数据。DNA中树控件主要有Tree和LWTree，树结构的实现方式主要可以通过Tree控件、LWTree控件和TreeViewer，通过各个方式实现的树结构从在外观上并无差别，都有三种风格：JWT.SINGLE、JWT.MULTI、JWT.CHECK。



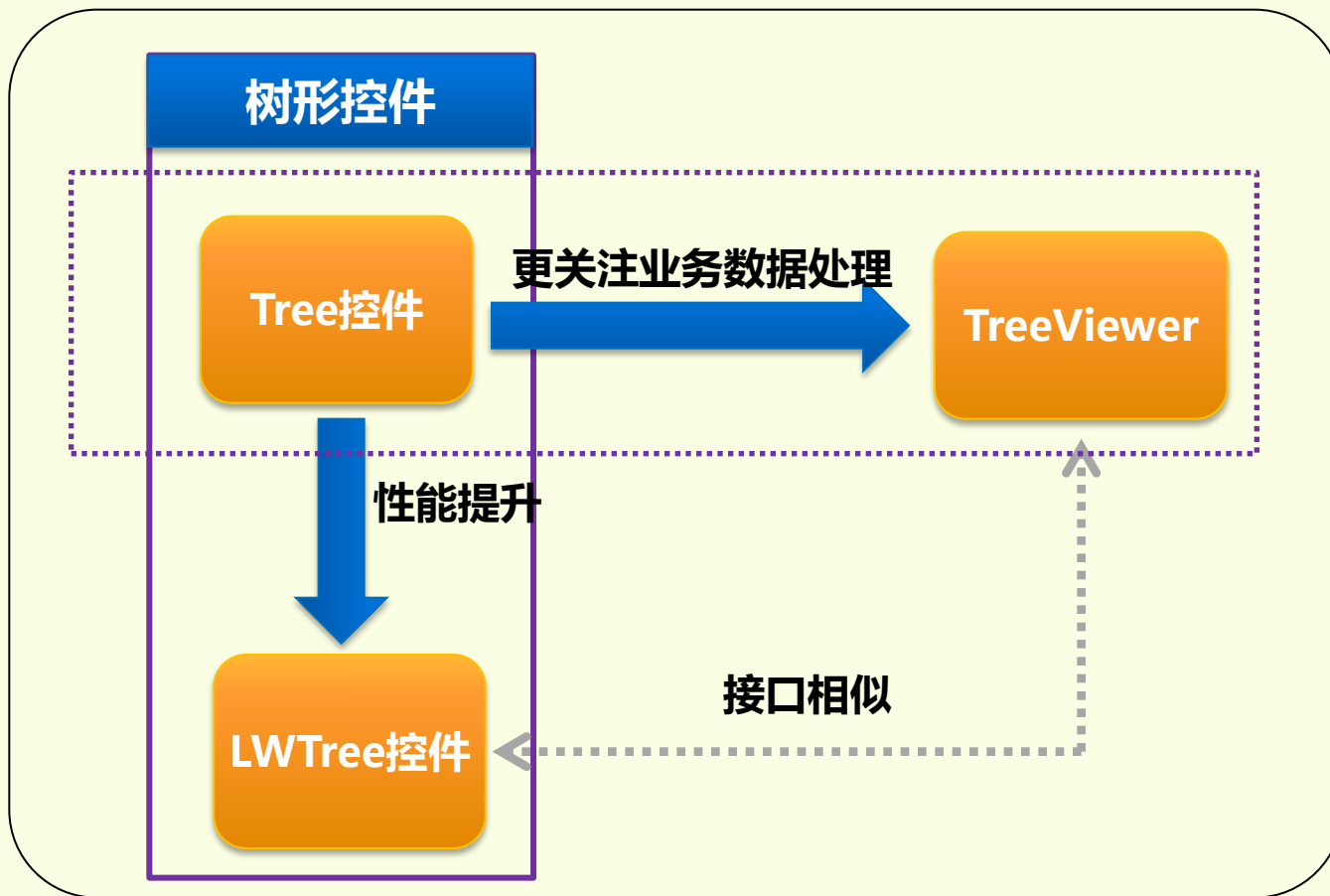
单选(JWT.SINGLE)



多选(JWT.MULTI)

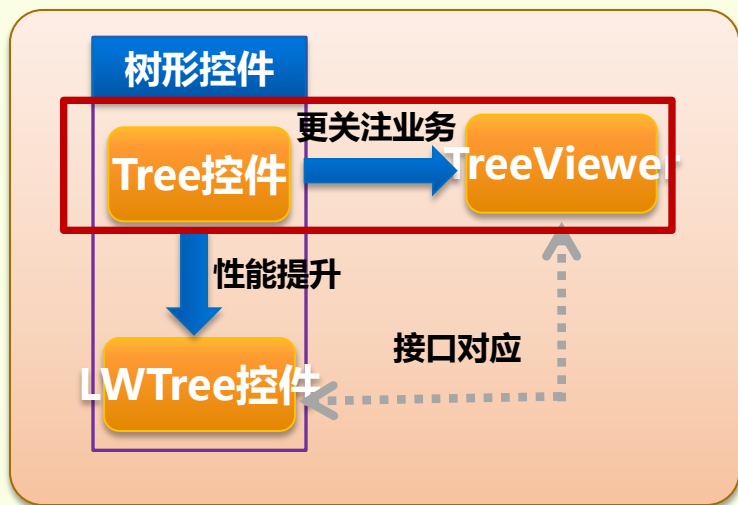


Check(JWT.CHECK)

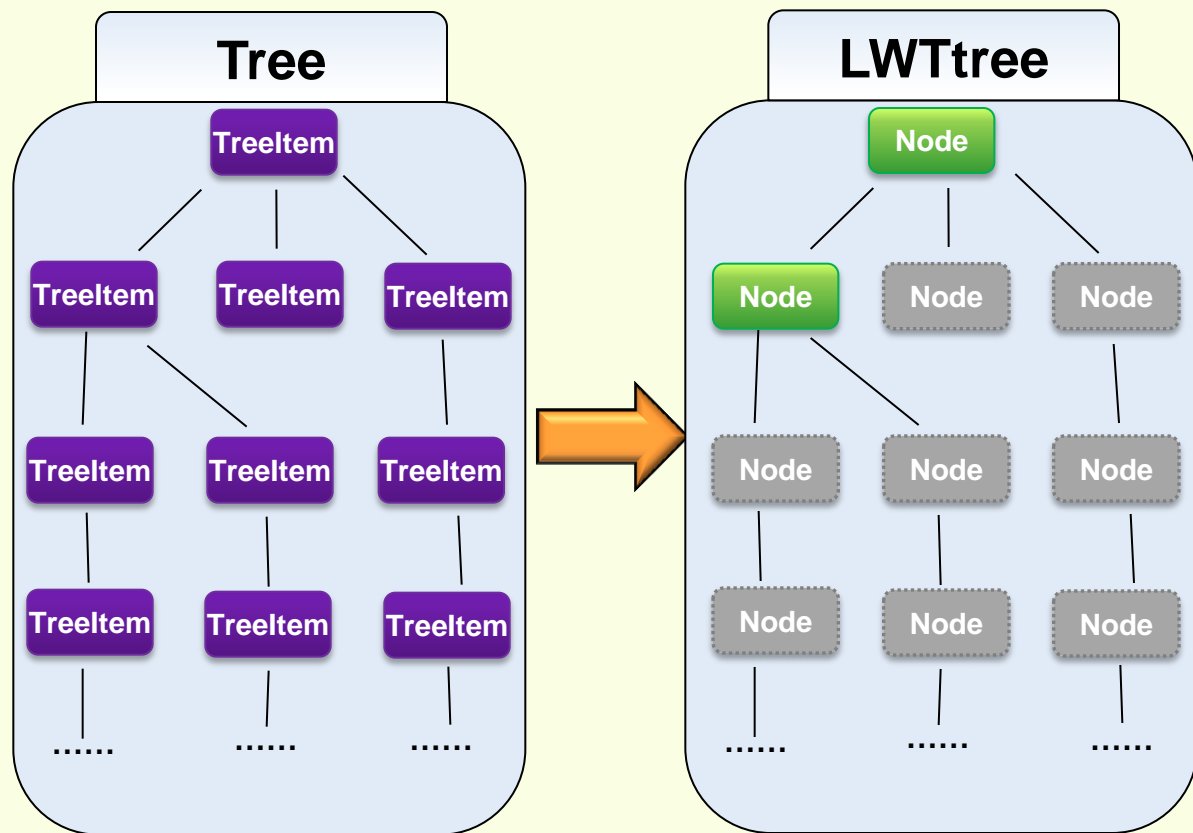
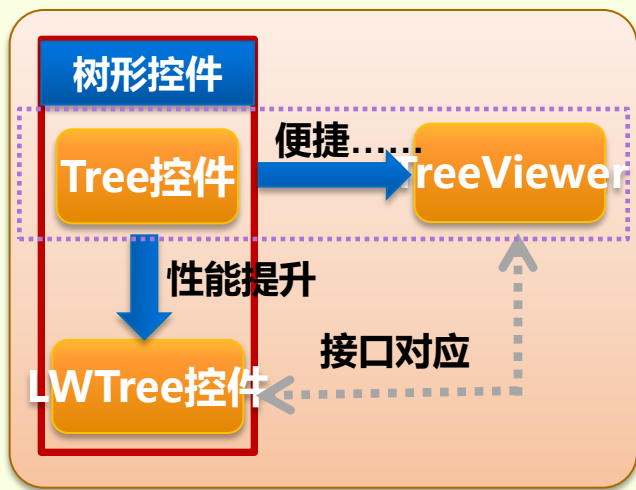


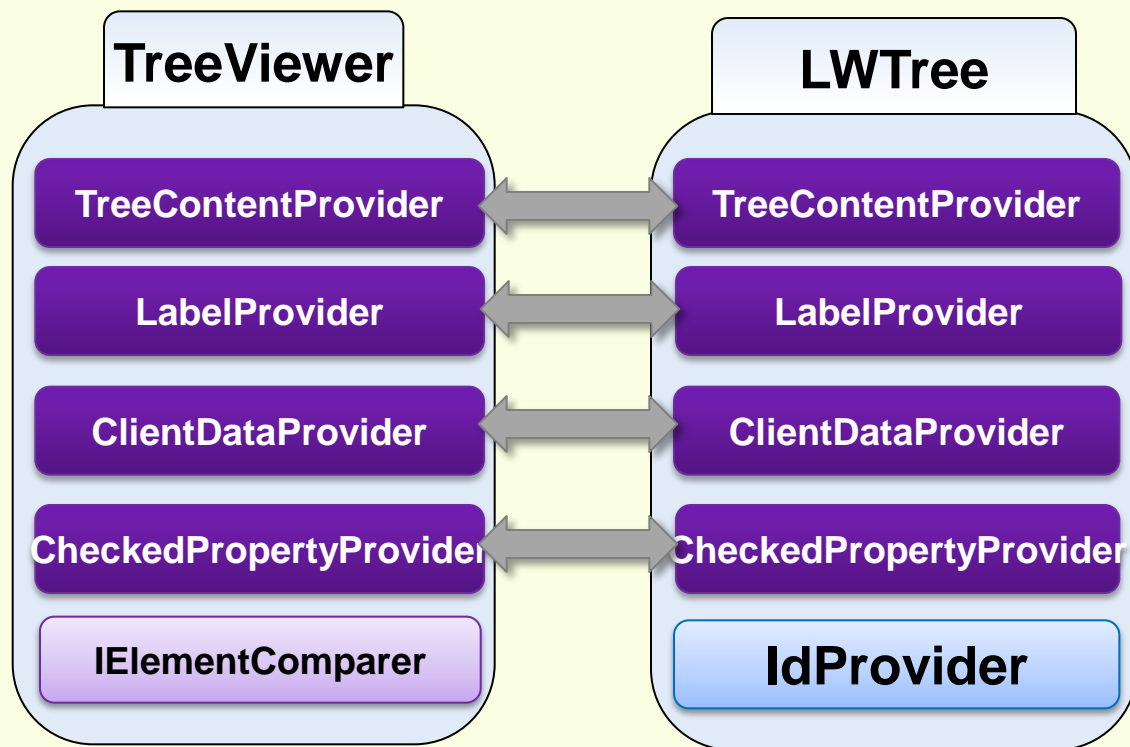
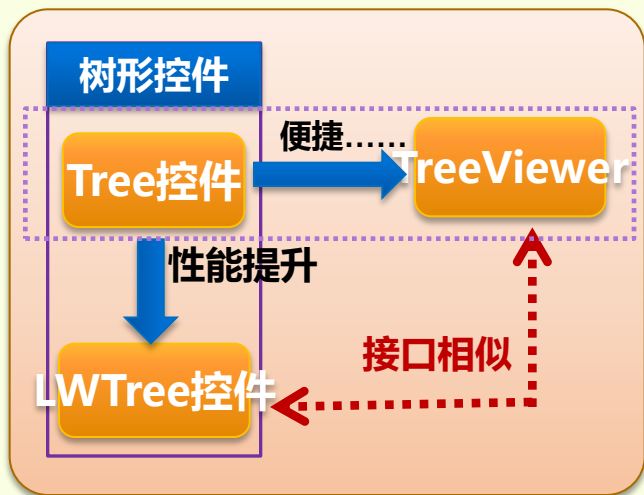


- TreeContentProvider
- LabelProvider
- CheckedProvider
- ClientDataProvider
- **IElementComparer**



- 节点不再是item对象。
- 树只保留了部分状态的同步。





控件介绍

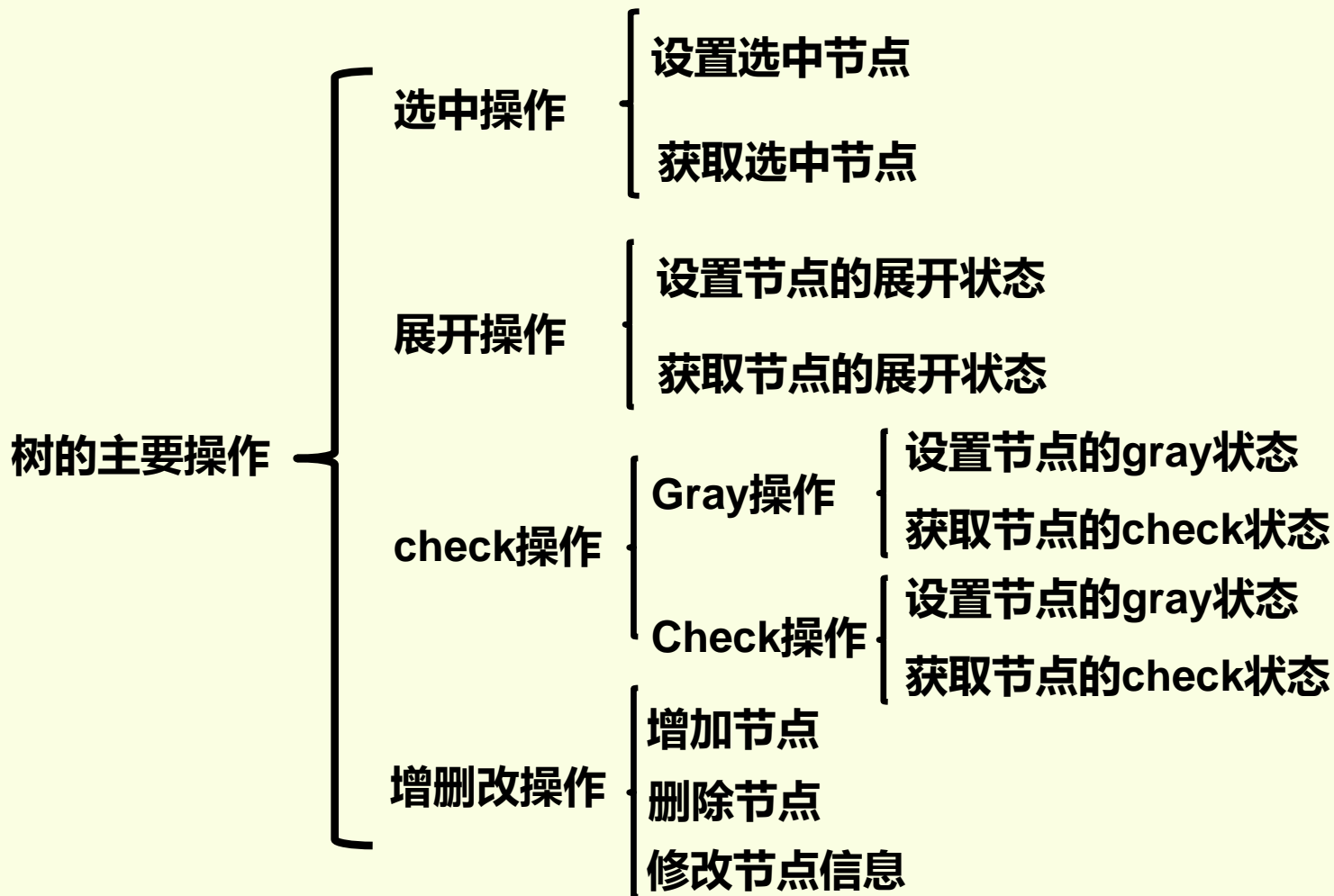
常用操作

常见问题

小结









select



内容管理

删除 剪切 粘贴 上移 下移 搜索 预览 关闭

信息管理

- 所有文档
  - 小知识
  - 问题报告
  - Test

小知识

序号	标题	类型	发布用户	发布日期	新
1	【界面框架】文本变化事件DocumentEvent中的	文档	芦星	2012-08-22	N
2	【逻辑框架】为什么UI界面在没有与业务逻辑层	文档	刘敬伟	2012-08-16	N
3	【关于封装的感悟总结】由Service中的结果提	文档	刘华西	2012-08-15	N
4	【界面框架】关于消息机制中的死循环	文档	刘敬伟	2012-08-15	N
5	【Portal】如何获取公共配置中的数据	文档	刘敬伟	2012-08-15	N
6	【DNA Developer】Java代码提示找不到的问题	文档	芦星	2012-08-15	N
7	【接口工程】关于接口工程中接口（Facade）、	文档	刘华西	2012-08-15	N
8	【基础应用平台】主体的删除。	文档	芦星	2012-08-14	N
9	【界面框架】DNA平台中Button的风格	文档	芦星	2012-08-14	N
10	【基础应用平台】如何刷新单据列表	文档	芦星	2012-08-10	N
11	【界面框架】LWTree获取选中节点或check节点	文档	吴娟	2012-08-09	N
12	【Portal】什么是单点登录？	文档	芦星	2012-08-09	N
13	【Portal】viewpart的五种类型。	文档	芦星	2012-08-09	N
14	【界面框架】在TreeView中如何正确使用get	文档	刘华西	2012-08-09	N
15	【数据处理】关于ORM脚本的使用	文档	刘华西	2012-08-09	N
16	【界面框架】showPage的用法	文档	刘敬伟	2012-08-07	N
17	【接口工程】请明确facade的发音。	文档	芦星	2012-08-07	N
18	【界面框架】如何打开一个功能界面？	文档	芦星	2012-08-06	N

当前显示 18 条，共 18 条

共 1 页 第 1 页

实现方法	服务端方法	客户端方法
Tree	<pre>setSelection(int index) ; setSelection(int[] indices) ; setSelection(TItem item) ; setSelection(TItem[] items) ; selectAll() ; deselectAll() ;</pre>	<pre>setSelection(Items);</pre>
TreeViewer	<pre>setSelections(Object[] selection) ; setSelection(Object selection) ; 通过内部Tree里的方法 ;</pre>	-----
LWTree	<pre>setSelection(Object object) ; setSelection(Object[] objects) ; setSelection(String objectId) ; setSelection(String[] selection) ;</pre>	no

实现方法	服务端方法	客户端方法
Tree	<pre> getSelectionCount() ; getSelectionIndex() ; getSelectionIndices() ; isSelected(int index) ; TItem[] getSelection() ;                     </pre>	<pre> getSelectedItems();                     </pre>
TreeViewer	<pre> Object getSelection(); Object[] getSelections(); Int getSelectionIndex();                     </pre>	<p>—————</p>
LWTree	<pre> String[] getSelectedIds() ; String getSeleted() ;                     </pre>	<pre> getSelectedItems();                     </pre>

- 在手动选择或者使用setSelection时会触发选择事件。

实现方法	服务端事件	客户端事件
Tree	SelectionListener	CLIENT_EVENT_SELECTION
TreeViewer	SelectionListener	-----
LWTree	SelectionListener	CLIENT_EVENT_SELECTION

- selectionChanging将会先于selection执行,如果要在selectionChanging中取消selection的相应,可以控制SelectionChangingEvent的doIt属性来实现。

实现方法	服务端事件	客户端事件
Tree	SelectionChangingListener	no
TreeViewer	No ( 但可以通过内部Tree操作 )	-----
LWTree	SelectionChangingListener	no

多语言管理

多语言分组

搜索

- 单据
- 基础数据
  - 自动生成设置
  - 基础数据
    - 民族**
    - 假别
  - 多组织机构类型自定义分组
  - 请假单
- 工作流
- 所有功能
- 所有文档
- 所有PORTLET
- 数据查询
- 数据查询工具按钮
- 数据建模
- 单据列表定义

多语言信息

序号	标题	中文(简体)	中文(繁体)	英语	维
1	民族	发生地			

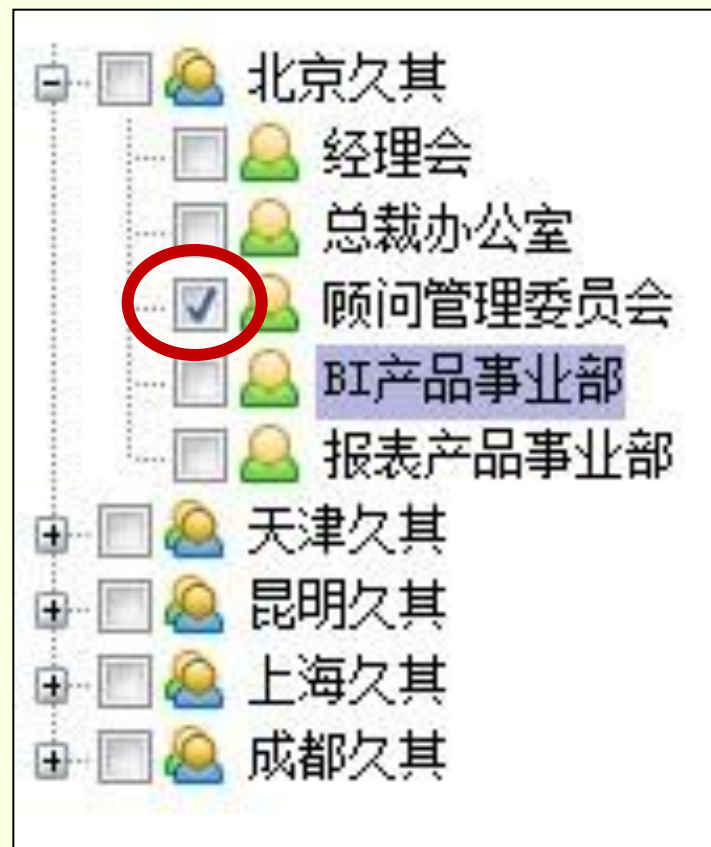
信息提示

是否保存当前数据?





check →



中国注册会计师行业管理信息系统

<
首页
首页子页面
注册管理
继续教育（执业）
财务报表
业务监管
继续教育（非执

📁 专家库维护
📁 日常管理
📁 专家库名录
📁 统计分析

专家入库出库(申请专家初选)×
历史资料备案×
专家聘用与评价(聘用考核)×

姓名

身份证号

性别 全部

专业 报表审计, 报表审阅, 民

职称 全部

聘用状态 全部

导出EXCEL

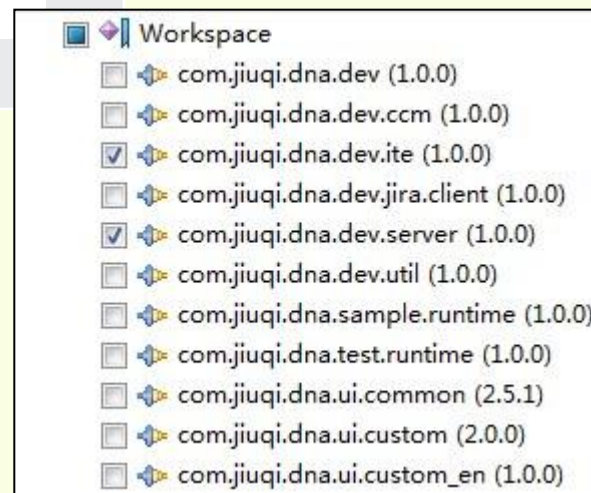
	姓名	性别	年龄	职务	单位	专业	聘用状态

- 会计
- 审计
  - 审计实务
    - 报表审计
    - 报表审阅
    - 其他鉴证业务
  - 审计理论
    - 注册会计师审计
    - 企业内部审计
    - 国家审计
- 法律
  - 法理学
  - 民商法

实现方法	服务器端方法	客户端方法
TreeItem	setChecked(Boolean value) ;	setChecked(checked);
TreeViewer	setChecked(Object element, boolean state) ; setCheckedElements(Object[] elements) ; 使用内部Tree ;	-----
LWTree	setChecked(String objectId, boolean state) ; setChecked(Object object, boolean state) ;	no

实现方法	服务器端方法	客户端方法
TreeItem	isChecked();	isChecked()
TreeViewer	isChecked(Object element); 使用内部Tree;	-----
LWTree	no	isChecked(id); isCheckedItems ();

	true	false
check		
gray		



	服务端方法	客户端方法
TreeItem	setGrayed(Boolean grayed) ;	setGrayed(grayed)
TreeViewer	setGrayed(Object element, boolean state); setGrayedElement(Object[] elements);	————
LWTree	setGrayed(Object object, boolean state) ; setGrayed(String objectId, boolean state) ;	no

实现方法	服务端方法	客户端方法
TreeItem	getGrayed() ;	isGrayed()
TreeViewer	getGrayed(Object element); Object[] getGrayedElement();	————
LWTree	no	isGrayed (id)

- Check事件在点击复选框或使用setChecked方法时都会触发。

实现方法	服务端事件	客户端事件
Tree	CheckListener	JWT. CLIENT_EVENT_CHECK;
TreeViewer	CheckListener	-----
LWTree	CheckListener	JWT. CLIENT_EVENT_CHECK;





expand

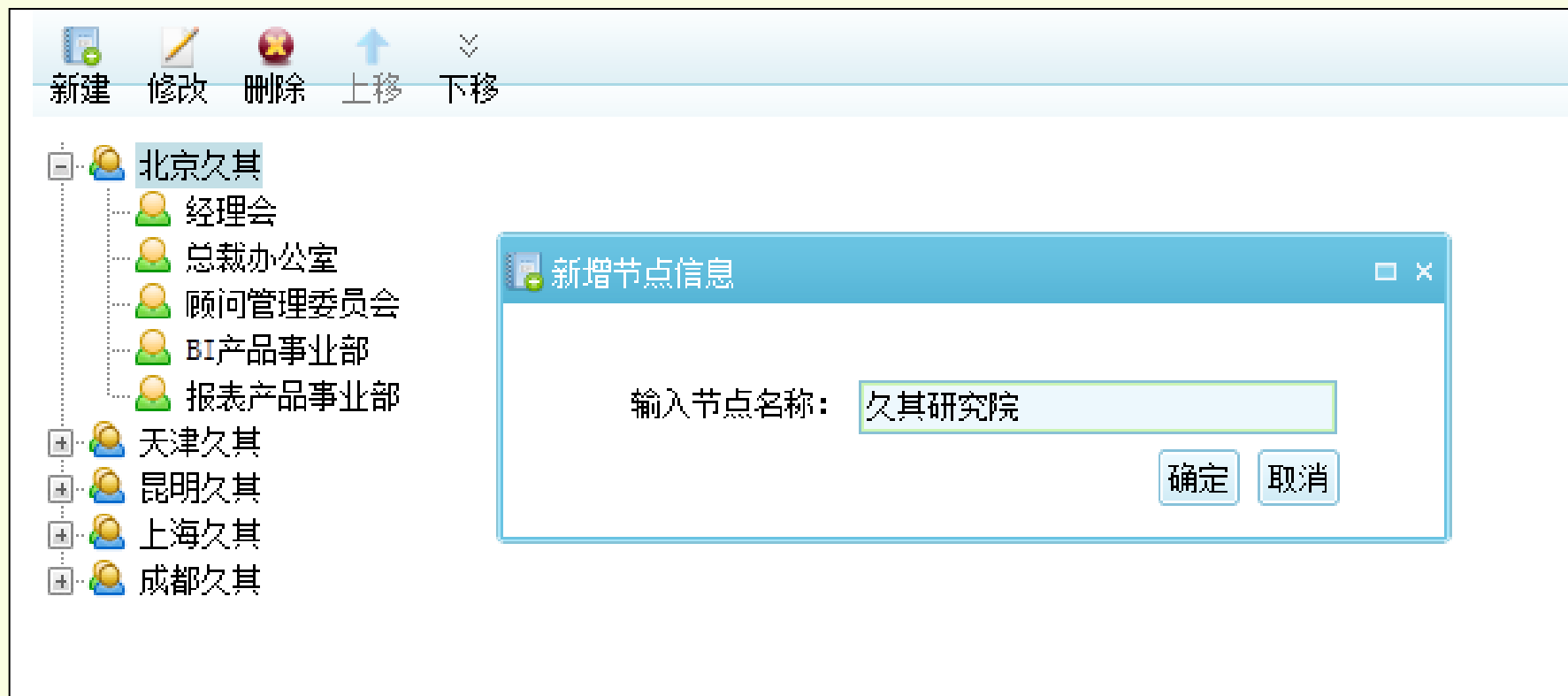


实现方法	服务端方法	客户端方法
TreeItem	setExpanded(Boolean newValue) ;	setExpanded(expanded);
TreeViewer	setExpandedState(Object element, boolean expanded) ; 通过内部Tree ;	-----
LWTree	setExpand(Object object, boolean expand) ; setExpand(String objectId, boolean expand)	no

实现方法	服务端方法	客户端方法
TreeItem	getExpanded() ;	isExpanded();
TreeViewer	通过内部Tree ;	-----
LWTree	isExpanded(Object object) ; isExpanded(String objectId) ;	no

- 展开和收缩事件在点击展开节点或调用setExpanded方法时都会触发。
- TreeListener的treeCollapsed()和treeExpanded() 用来处理收缩和展开的事件响应。

实现方法	服务端事件	客户端事件
Tree	TreeListener	JWT.CLIENT_EVENT_TREE_EXPANDED; JWT.CLIENT_EVENT_TREE_COLLAPSED
TreeViewer	不能使用	————
LWTree	no	JWT.CLIENT_EVENT_TREE_EXPANDED; JWT.CLIENT_EVENT_TREE_COLLAPSED

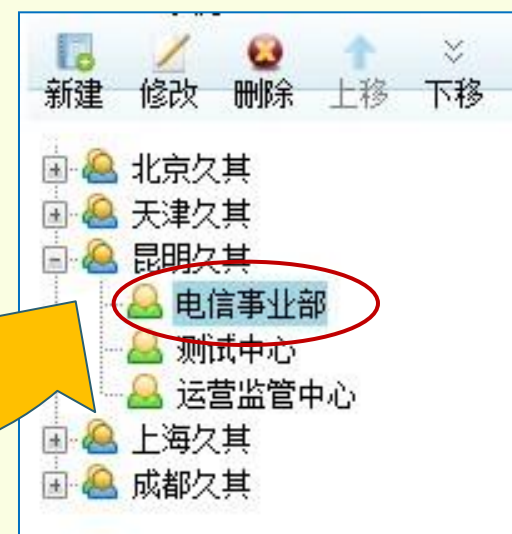


实现方法	增加	删除
Tree	New TreeItem(tree);	remove(int index); remove(int[] indices); remove(TItem item); removeAll(); 可以通过TreeItem的dispose方法。
TreeViewer	add(Object parent, Object child); add(Object parent, Object[] children); insert(Object parent, Object child, int index); 通过内部Tree ;	remove(Object element); remove(Object[] elements); remove(Object parent, Object[] children); 通过内部Tree ;
LWTree	add(Object[] objects) ; add(Object object); insert(Object object, Object insertBefore); insert(Object object, String insertBefore);	remove(Object[] objects); remove(Object object) ; remove(String[] objectIds); remove(String objectId) removeAll() ;

The screenshot displays a software interface for managing organizational nodes. At the top, there is a toolbar with five icons: a document for '新建' (New), a pencil for '修改' (Modify), a red 'X' for '删除' (Delete), an upward arrow for '上移' (Move Up), and a downward arrow for '下移' (Move Down). Below the toolbar is a tree view of nodes. The root node is '北京久其', which has five sub-nodes: '经理会', '总裁办公室', '顾问管理委员会', 'BI产品事业部', and '报表产品事业部'. Below this are four more root nodes: '天津久其', '昆明久其', '上海久其', and '成都久其'. The 'BI产品事业部' node is selected. A dialog box titled '修改节点信息' (Modify Node Information) is open in the foreground. It contains a label '输入节点名称:' (Enter node name:) followed by a text input field containing 'BI产品事业部'. At the bottom right of the dialog are two buttons: '确定' (OK) and '取消' (Cancel).

实现方法	服务端方法	客户端方法
TreeItem	setText(String Text); setImage(ImageDescriptor image);	setText(text); setImage(imageId, width, height);
TreeViewer	通过提供器更新数据对象; 通过内部Tree;	-----
LWTree	通过提供器更新数据对象; 更新数据对象;	setItemText(id, text);





实现方法	方法
TreeItem	通过TreeItem直接修改。
TreeViewer	refresh(); update(Object obj); update(); 通过内部Tree;
LWTree	refresh(); refresh(Object object) ; refresh(String objectId) ; update(Object object) ;

控件介绍

常用操作

常见问题

小结



## 1. 有时无法展开或选中LWTree或TreeView的节点？

检查提供器或比较器是否正确。

## 2. setGray和setEnabled有什么区别？

gray是在树控件的check下多加了两个状态。

**3. 树节点的上移下移操作需要自己实现吗？**

可以直接使用TreeItem上的相关移动方法。

**4. 为什么有时候进行树节点遍历会很慢？**

尽量使用非递归的方法进行遍历。

控件介绍

常用操作

常见问题

小结



**1** DNA中的树控件有Tree、LWTree，TreeView不是控件。

**2** LWTree是轻量级树，没有Item对象。

**3** 树控件在服务器端和客户端对事件的支持情况是不一样的。

**4** 出现问题时先检查提供器和使用方法。

控件	适用场景
Tree	简单的数据展示
TreeViewer	复杂业务场景的处理
LWTree	适用于性能要求较高的业务场景





# DNA

久其期待与您共创未来

久其软件  
钻石品质

诚信协作  
奉献超越