



链滴

ElementUI 中 tree 控件使用说明

作者: [alakis](#)

原文链接: <https://ld246.com/article/1519116316882>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

一句话总结，就是很好很强大。但是ElementUI中树的加载可以通过load属性一个一个懒加载，在官的案例中有这种用法，个人非常不推荐这种用法，很麻烦，节点的动态删除和修改都很麻烦，我一开就使用了这个，后来发现动态添加节点、删除节点、节点过滤统统都很麻烦，然后换个思路，使用data，所有问题一下就都解决了。使用data思路如下：在tree中使用data属性加载数据，后期所有涉及到点动态添加删除的事，我们统统只去操作data就可以了，这也符合数据驱动视图的思想。

1.在页面加载时，我在mounted方法中就去获取了所有的部门数据，然后直接加载到树上（这种方式较省事，特别是处理后面动态添加删除时很方便，小伙伴在工作中可以根据实际情况选取合适的策略。核心代码如下：

```
this.getRequest("/system/basic/dep/-1").then(resp=> {
  this.treeLoading = false;
  if (resp && resp.status === 200) {
    this.treeData = resp.data;
  }
})
```

2.添加节点时，当服务端添加成功后，会返回刚刚添加的节点的json，通过递归找到这一段json所在位置，动态假如到树中。递归过程如下：

```
setDataToTree(treeData,pId,respData){
for(var i=0;i<treeData.length;i++) {
  var td = treeData[i];
  if(td.id==pId) {
    treeData[i].children=treeData[i].children.concat(respData);
    return;
  }else{
    this.setDataToTree(td.children, pId, respData);
  }
}
}
```

这是通过递归找到添加的位置，动态添加。

3.删除节点时，当服务端返回删除成功时，一样找到删除节点的位置，动态从树中删除。核心代码如下：

```
deleteLocalDep(treeData,data){
for(var i=0;i<treeData.length;i++) {
  var td = treeData[i];
  if(td.id==data.id) {
    treeData.splice(i, 1);
    return;
  }else{
    this.deleteLocalDep(td.children, data);
  }
}
}
```

这也是在删除成功后，通过递归找到删除的位置，动态删除数据。

4.由于我采用了一次性加载所有部门的策略，所以节点过滤这块特别容易，基本上完全参考官方案例能实现。

其他一些琐碎的技术就不值得介绍了，大家在源码中自行研究，有问题欢迎留言讨论。