1. 菜单树展开问题

期望效果: 窗口打开时, 调用后端接口, 返回一个完整树的json结构的数据包, Extjs解析该数据包并将整棵树完整展现

遇到的困难:

* Tree控件在构造成功后会自动load数据, 导致此刻userId和session不能作为查询参数, 进而后端无法校验安全性

解决方法:

* + 去掉校验功能, 目前采用该方法
  + 在Tree的store中不定义proxy, 等到界面构造完成后再setProxy.
* AjaxProxy在请求URL时, 可以拼装一个node的ID, 如果想同时拼装两个参数, 不知道怎么做. 比如要展开一个节点, 希望按照该节点的type和ID去服务端查询其子节点, 此时Extjs框架只能附带一个ID参数

解决办法:

把ID作为TYPE+ID的字符串, 每次需要时, 都是拿到这个改造后的ID, 按需拆分. 这种方式使用起来及其繁琐, 影响前端和后端, 需要更改.

* Tree控件使用local的数据源, 可以展开多级树, 但是用remote的store, 只能展开第一级, 尽管已经从服务端获得了全部的树节点.

解决办法:

目前由服务端拼装expanded参数和leaf参数, Tree可以达到想要的效果. 但是理论上前端应该可以控制树形展开的效果. 服务端只能逐级给前端返回数据, 如果把整个树的数据都发送出来, 就会展示出错, 导致前端不断的刷新.

Treestore里面有个loaded参数, 要对加载过的目录节点设置其为TRUE, 否则会一直加载该节点, 尤其该节点child是空的情况

* 创建/修改/删除后, 树节点未刷新

解决办法:

创建后, 获得创建成功的对象ID, 生成一个Node加入tree中;

修改后, replaceNode, 如果node修改了树结构, 需要将原有的父节点删除node, 并将新的父节点增加node

删除后, 根据原节点找到父节点, 并remove该node

* Tree 控件有两个保留字id和parent, 也许还有其他保留字. 用一个store使用id或者parent作为field就会导致树结构打开关闭时出错

1. Hibernate 的缓存功能

目前每次查询, 即使同样的查询都会到数据库重新检索.

1. Menu maintain 功能里, 点击左边的树, 右边的两个combobox无法显示数据, 因为现在store还没有设定, 所以setValue其实是无效的.
2. 文件上传问题

文件上传必须把filefiled控件放到form中, 通过调用form.submit方法来提交. 如果直接使用Ext.Ajax.request, 无法根据这个filefiled控件获得整个文件的数据.

Filefield控件显示的文件路径是个虚拟的, 这个不需要深入研究, 只要隐藏路径框即可.

在springmvc中, 要先配置一个multipart Resolver

<!-- 配置MultipartResolver 用于文件上传 使用spring的CommosMultipartResolver -->

<beans:bean id=*"multipartResolver"* class=*"org.springframework.web.multipart.commons.CommonsMultipartResolver"*

p:defaultEncoding=*"UTF-8"*

p:maxUploadSize=*"5400000"*

p:uploadTempDir=*"fileUpload/temp"* >

</beans:bean>

1. 菜单树上修改菜单价格, 通过一个弹出框修改了价格, 保存成功后如何更新原来的节点数据?
2. Android 中设置Layout的background会把控件拉伸, 目前未找到好的解决方案. 如果使用ImageView的src指定图片, 可以防止被拉伸. 但是菜单上的那个浮动选择按钮无法在ImageView的顶层显示, 显示效果会有折扣. 采取的办法还是用server段重新生成指定大小的图片, 客户端还是用background属性显示.

解决办法:

这里应该先把layout组件加入activity, 然后再setBackground. 因为在加入的时候已经指定了大小, 就不会再被图片大小影响. 如果先setBackground, 因为大小未确定, 所以就使用了图片的大小.

1. Indent 查询的地方, web端, 当下面的page控件改变页码时, 即切换到某一页, 会给grid的proxy指定一个page和start参数, 但是这会影响后面的查询. 比如说已经翻页到第三页, 此时page=3, 如果此时改了查询条件, 点”查询”按钮, 此时的查询参数依然会把page=3发给服务端, 导致查询数据不准确.

解决办法:

调用store.loadPage(1);