

ADF - Делаем дерево более симпатичным

При создании проекта использовался **Weblogic 12.1.2**, возможно в более ранних версиях используются другие названия стилей.

Было	Стало
 <p>Ппки можно выбрать</p> <p>У конечных узлов есть кнопка раскрытия</p>	 <p>Ппки при выборе визуально не выделяются</p> <p>Скрыта ненужная кнопка</p>

Как можно добиться такого результата

В **nodeStamp** дерева у надписи я определил класс:

```
<af:tree value="#{bindings.SMenuView1.treeModel}" var="node" rowSelection="single" id="t1"
selectionListener="#{WorkMenuBean.selectMenuItem}" binding="#{WorkMenuBean.menuTree}"
inlineStyle="margin-left: 5px;" emptyText="#{viewControllerBundle['Message.NO_DATA_TO_DISPLAY']}">
  <f:facet name="nodeStamp">
    <af:image source="/images/menu/type#{node.MenuTypeId}.png" id="i2"
inlineStyle="padding-right: 5px;vertical-align: middle;"/>
    <af:outputText value="#{node.Descr}" id="ot1"
inlineStyle="#{node.Status eq '0' ? 'color:Red;' : ''}"
styleClass="#{node.MenuTypeId == 1 ? 'mainMenuFolderNode':'mainMenuItemNode'}">
  </f:facet>
</af:tree>
```

Классы **mainMenuFolderNode** и **mainMenuItemNode** нигде прописывать не нужно, они будут служить только как метки для поиска данного элемента на странице. У меня в данном случае используется атрибут из базы данных, но вы можете использовать другие методы, для определения является ли данный узел конечным, например, `"#{node.children != null ? 'mainMenuFolderNode':'mainMenuItemNode' }`".

Для облегчения процесса я использовал **jQuery** (а куда сейчас без него?). Подключаем его:

```
<af:document title="#{viewControllerBundle['Label.PORTAL TITLE']}" id="d1">
  <af:resource type="javascript" source="/js/jquery-2.1.1.min.js"/>
</af:document>
```

Пишем **JavaScript** функцию, наводящую красоту:

```
<af:resource type="javascript">
  function prepareMainMenu(){
    // запретить выбирать каталоги
    $('span.mainMenuFolderNode').parent('span.xv9').removeClass('xv9');
    // скрыть развертывание у конечных элементов
    $('span.mainMenuItemNode').parent('span.xv9').siblings('span.xvj').children('a.xv3').css('visibility','hidden');
  }
</af:resource>
```

Эту функцию будем дергать при помощи PagePhaseListener:

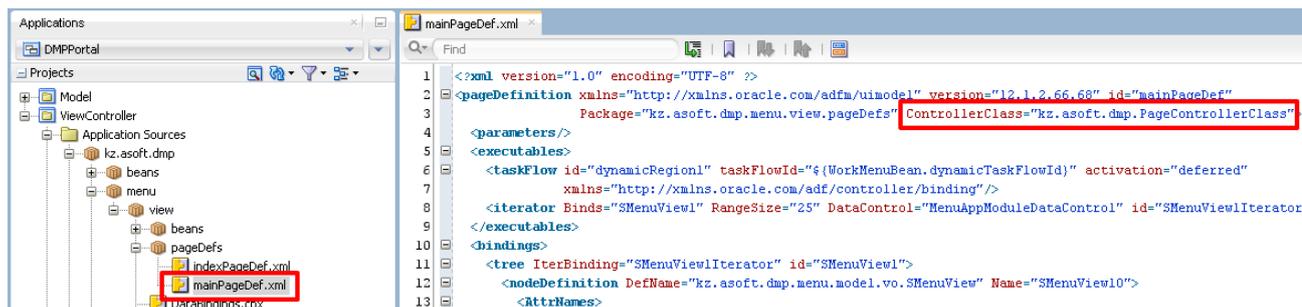
```
public class PageControllerClass implements PagePhaseListener {
    private static Logger LOGGER = Logger.getLogger(PageControllerClass.class);

    public PageControllerClass() {
        super();
    }

    @Override
    public void afterPhase(PagePhaseEvent pagePhaseEvent) {
        if (pagePhaseEvent.getPhaseId() == Lifecycle.PREPARE_RENDER_ID) {
            FacesContext fctx = FacesContext.getCurrentInstance();
            ExtendedRenderKitService erks = Service.getRenderKitService(fctx, ExtendedRenderKitService.class);
            //invoke JavaScript
            erks.addScript(fctx, "prepareMainMenu();");
        }
    }

    @Override
    public void beforePhase(PagePhaseEvent pagePhaseEvent) {
        // TODO Implement this method
    }
}
```

Остается прописать данный класс в определении страницы:



Все, можно запускать и радоваться красоте, которая у нас получилась. 😊