Home
Específicos
Flex
Getting Started
Java
Processo Datasul
Progress
Tools
Trilhas
User
Page Operations
Browse Space

SDK - Software Development Kit - v.1.0

DatasulZoom - DataGrid (Estilo ZoomManager)

Added by Marcelo Paes Rech, last edited by Marcelo Paes Rech on Jun 23, 2010 (view change)
Labels
incubado, componente

Visão Geral

Em alguns momento é necessário utilizar o componente de zoom sem a interface gráfica, ou mesmo utilizar as consultas do zoom em um componente de interface customizado DataGrid por exemplo).
O DatasulZoom do metadados possui as funcionalidades para efetuar consultas sem a camada UI. Neste documento, estaremos propondo a implementação da consulta utilizan componente customizado dentro de um DataGrid

Localização

Quando utilizar o Componente?

Quando houver a necessidade de utilizar as consultas do DatasulZoom se a interface gráfica ou mesmo surgindo a necessidade de utilizar as consultas em um componente cust

Pré-Requisitos

Conhecimentos em Flex e conhecimentos no componente DatasulZoom do metadados.

Exemplo de Uso

Em nosso exemplo vamos utilizar o componente DatasulZoom. Para a utilização criamos uma função que cria a instância do DatasulZoom. Abaixo segue o fonte:

```
private function createZoom():DatasulZoom{

                                var dzZoom:DatasulZoom = new DatasulZoom();
                                dzZoom.entityHandler = "prgfin/cfl/cfl908we.py";
                                dzZoom.entityName = "emsfin.UnidControlFinanc";
                                dzZoom.fieldCode = "primaryKey.codUnidControlFinanc";
                                dzZoom.fieldLabel = "desUnidControlFinanc";
                                dzZoom.queryColumnName = "primaryKey.codUnidControlFinanc";
                                dzZoom.typeColumn = "String";
                                dzZoom.xmlName = "view.ems5.CRUDUnidControlFinanc";
                                return dzZoom;

                        }
```

Após a criação do componente podemos criar as funções que irão efetuar as chamadas para as consultas. Abaixo segue o fonte:

```
private function findObect(e:AdvancedDataGridEvent) : void {
                                var str : String = e.currentTarget.itemEditorInstance.value;
                                if(this.oldValue != str && str != ""){
                                        this.selectedItem = e.currentTarget.selectedItem;
                                        var manager:DatasulZoom  = this.createZoom();
                                        //Adiciona o listener de resposta
                                        manager.addEventListener(ZoomEvent.ZOOM_FIND_OBJECT_RESULT, zoomHandler);
                                        manager.addEventListener(ZoomEvent.ZOOM_FIND_OBJECT_FAULT, faultHandler);
                                        manager.findObject(str);
                                }
                        }

                        public function openZoom() : void {
                                var manager:DatasulZoom = this.createZoom();
                                this.selectedItem = this.adg.selectedItem;

                                //Adiciona o listener de resposta
                                manager.addEventListener(ZoomEvent.ZOOM_SELECT, zoomHandler);
                                manager.addEventListener(ZoomEvent.ZOOM_BACK, cancelHandler);
                                //Abre o DatasulZoomPopUp
                                manager.openZoom(DisplayObject(Application.application));
                        }
```

Com as chamadas criadas, implementamos o DataGrid. Na implementação do DataGrid existe a necessidade de criar um ItemEditor, para incluir o componente customizado qu dados informados pelo usuário para efetuar a chamada ao DatasulZoom. Abaixo segue o fonte do DataGrid e do ItemEditor necessário:
DataGrid

```
<mx:AdvancedDataGrid id="adg" width="100%" height="100%" dataProvider="{content}"
                itemEditEnd="this.findObect(event)" itemEditBegin="this.updateOldValue(event)" editable="true">
                <mx:groupedColumns>
                        <mx:AdvancedDataGridColumn headerText="ID" dataField="id" editable="false"/>
                        <mx:AdvancedDataGridColumn headerText="Task" dataField="task" editable="false"/>
                        <mx:AdvancedDataGridColumnGroup headerText="Customer" editable="true">
                                <mx:AdvancedDataGridColumn headerText="ID"  editorDataField="value" editable="true">
                                        <mx:itemEditor>
                                                <mx:Component>
                                                        <zoommanager:ZoomEditor open="{parentDocument.openZoom}" value="{data.unidControl['primaryKey.codUnidControlF
                                                </mx:Component>
                                        </mx:itemEditor>
                                        <mx:itemRenderer>
                                                <mx:Component>
                                                        <mx:Label text="{data.unidControl['primaryKey.codUnidControlFinanc']}"/>
                                                </mx:Component>
                                        </mx:itemRenderer>
                                </mx:AdvancedDataGridColumn>
                                <mx:AdvancedDataGridColumn headerText="Short Name" editable="false">
                                        <mx:itemRenderer>
                                                <mx:Component>
                                                        <mx:Label text="{data.unidControl.desUnidControlFinanc}"/>
                                                </mx:Component>
                                        </mx:itemRenderer>
                                </mx:AdvancedDataGridColumn>
                        </mx:AdvancedDataGridColumnGroup>
                </mx:groupedColumns>
        </mx:AdvancedDataGrid>
```

ItemEditor

```
<?xml version="1.0" encoding="utf-8"?>
<mx:HBox xmlns:mx="http://www.adobe.com/2006/mxml" width="100%" height="100%">

	<mx:Script>
		<![CDATA[
			[Bindable]
			public var value:String = "";
			public var open:Function;
			private function onRelease():void{
				value = txi.text;
			}
		]]>
	</mx:Script>

	<mx:TextInput id="txi" width="50" text="{value}" focusOut="onRelease()" enter="onRelease()"/>
	<mx:LinkButton label="find" click="open()"/>

</mx:HBox>
```

A alteração no retorna da consulta é feito através de referência. Portanto, nos métodos anteriores foi armazenado o item que efetuou a consulta. No retorno, este item armazena sofre a alteração que, após o refresh da lista, permuta as alterações para a tela. Abaixo segue exemplo de como recuperar os dados do item selecionado:

```
private function zoomHandler(e:ZoomEvent):void{
				if(e.object != null){
					var position:int = adg.verticalScrollPosition;//recupera a posição do scroll atualmente
					this.selectedItem.unidControl = e.object;// Atualiza o dataprovider
					this.content.refresh();//Força a lista a permutar as alterações
					adg.verticalScrollPosition = position;// devolve o grid para a mesma posição que estava antes da atualização ocorrer (scroll)
				}
			}
```

Para melhorar a visualização do exemplo, abaixo segue a lista que serviu como dataProvider do DataGrid:

```
[Bindable]
			private var oldValue:Object = null;

			[Bindable]
			private var selectedItem:Object = null;

			[Bindable]
			private var content:ArrayCollection = new ArrayCollection();

			private function startup():void{
				for(var x:int = 0; x < 10; x++){
					var obj:Object = new Object();
					obj["id"] = x+1;
					obj["task"] = "Sample Zoom";
					obj["unidControl"] = new Object();
					obj["unidControl"]["primaryKey.codUnidControlFinanc"] = "";
					obj["unidControl"]["desUnidControlFinanc"] = "";
					content.addItem(obj);
				}
			}
```

Pode-se notar que o atributo unidControl, é um objeto complexo dentro de nosso dataProvider. Assim, recuperamos o objeto que referência a row selecionada e alteramos da re antigo unidControl, pelo que foi retornado na consulta.

Referências

As refencias deste documento foram retiradas da documentação do componente ZoomManager. As princípios do componente ZoomManager podem ser aplicado através do co DatasulZoom do metadados.

Ver Também

Antigo componente ZoomManager

Informações sobre este Documento

Versão	Estado	Compatibilidade	Criador	Criação	Último Revisor	Última Revisão
4	incubado		Marcelo Paes Rech	Jun 23, 2010 10:52	Marcelo Paes Rech	Jun 23, 2010 15: