This page last changed on Jun 20, 2013 by tyler.johnson.

Menu Confirmation Dialog Tutorial

This tutorial will demonstrate the use of the <mobi:menuButton>, <mobi:panelConfirmation> and <mobi:submitNotification> components. The menuButton drop down will allow the user the option of selecting several actions and then confirm their selection via a confirmation dialog. The submit notification component will then prevent further user interaction until the backing bean process has completed. We will have cover two use cases:

- Single dialog
- Dialog for each menu item

The source code for both tutorials can be found at the bottom of the tutorial here. Additional information regarding these components can be found here:

- WIKI
- TLD

Add <mobi:fieldsetGroup> layout component

We will layout our form using the <mobi:fieldsetGroup> and <mobi:fieldsetRow> components. Please paste the following into the <h:body> of your ICEmobile page:

```
<h:form>
   <mobi:fieldsetGroup>
      <mobi:fieldsetRow>
      .......
      </mobi:fieldsetRow>
   </mobi:fieldsetGroup>
</h:form>
```

Add <mobi:menuButton>

The menuButton allows a user to select and execute a menu item while ActionListener methods can be assigned for each of the MenuButtonItem's in the menu. We will also tie the menuButton with a panelConfirmation and submitNotification component. Please add the following within the <mobi:fieldsetRow>:

```
<mobi:menuButton id="mnu" value="#{menu.menuData}" var="item">
</mobi:menuButton>
```

The menuButton value binding menu.menuData, will consist of a list of <SelectItem> objects that will populate our menuButtonItems below.

Example 1: Single Dialog

Add <menuButtonItem>

The menuButton will iterate over the list supplied by the value binding and create the menuButtonItems. The menuButtonItem options will 'add', 'edit' and 'delete' values. Please add the following code within the <mobi:menuButton>:

```
<mobi:menuButtonItem value="#{item.value}" label="#{item.label}" panelConfirmation="pc1"
submitNotification="sn1" actionListener="#{menu.performAction}" />
```

The <SelectItem> values provided by the menuButton value binding will supply our menuButtonItems value and label attributes. When clicking a MenuButtonItem, the <mobi:panelConfirmation> with id="pc1" will be triggered followed by the actionListener method performAction and lastly the <mobi:submitNotification> blocker.

Add <mobi:panelConfirmation> <mobi:submitNotification>

The panelConfirmation will prompt the user to either accept or decline the action while the submitNotifcation will prevent any further user interaction until page processing has completed. Please add the following code below the closing </mobi:menuButton> tag:

```
<mobi:panelConfirmation id="pc1" message="Perform Action?" />
<mobi:submitNotification id="sn1">
      <h:outputText value="Working..." />
</mobi:submitNotification>
```

Add action output

Add the following <mobi:fieldSetRow> below the row containing our panelConfirmation and submitNotification components:

```
<mobi:fieldsetRow>
Action: <h:outputText value="#{menu.action}"/>
</mobi:fieldsetRow>
```

Add Managed Bean

Please add the following managed bean code in addition to generating getters and setters:

```
@ManagedBean(name = "menu")
@ViewScoped
public class MenuBean implements Serializable {

	List<SelectItem> menuData = new ArrayList<SelectItem>();
	private String action;

	public MenuBean() {
		this.menuData.add(new SelectItem("Add", "Add"));
		this.menuData.add(new SelectItem("Edit", "Edit"));
		this.menuData.add(new SelectItem("Delete", "Delete"));
	}

	public void performAction(ActionEvent ae) {

		try {
			MenuButtonItem item = (MenuButtonItem) ae.getSource();
			action = (String) item.getValue();

			System.out.println("Action performed: " + action);

			Thread.sleep(5000);

		} catch (Exception e) {
		}
	}

	//TODO GENERATE GETTERS/SETTERS
```

Our actionListener performAction, will get the user's menu item selection from the ActionEvent and call Thread.sleep which is used to simulate a long running process in order to demonstrate the panelConfirmation functionality.

Example 2 - Multiple confirmation dialogs

Modify the ICEmobile page

The menuButtonItems will have parameterized panelConfirmation and SubmitNotification attributes. We will keep track of these values in a custom object in our managed bean. Please replace the previous menuButton with the following:

```
<mobi:menuButton id="mnu" value="#{menu.menuData}" var="item">
    <mobi:menuButtonItem value="#{item.value}" label="#{item.label}"
 panelConfirmation="#{item.panelConfId}"
     submitNotification="#{item.submitNotif}" actionListener="#{menu.performAction}" />
</mobi:menuButton>
```

Add additional panelConfirmation dialogs

Each menuButtonItem will now have a specific panelConfirmation component. Please add the following below the closing </mobi:menuButton> tag:

```
<mobi:panelConfirmation id="pc1" message="Add Item?" />
<mobi:panelConfirmation id="pc2" message="Edit Item" />
<mobi:panelConfirmation id="pc3" message="Delete Item" /
<mobi:submitNotification id="sn1">
	<h:outputText value="Working..." />
</mobi:submitNotification>
```

Add Managed Bean code

We will use a custom object to hold our menuButtonItem's value, label, and corresponding panelConfirmation and submitNotification ids. Please add the following managed bean code in addition to generating the getters and setters:

```
@ManagedBean(name = "menu")
@ViewScoped
public class MenuBean implements Serializable {

	List<ModelData> menuData = new ArrayList<ModelData>();
	private String action;

	public MenuBean() {
		this.menuData.add(new ModelData("Add", "Add","pc1","sn1"));
		this.menuData.add(new ModelData("Edit", "Edit","pc2","sn1"));
		this.menuData.add(new ModelData("Delete", "Delete","pc3","sn1"));
	}

	public void performAction(ActionEvent ae) {

		try {
			MenuButtonItem item = (MenuButtonItem) ae.getSource();
			action = (String) item.getValue();

			System.out.println("Action performed: " + action);

			Thread.sleep(5000);

		} catch (Exception e) {
		}
	}

	//TODO GENERATE GETTERS/SETTERS

	public class ModelData implements Serializable {
		private String value;
		private String label;
		private String panelConfId;
		private String submitNotif;

		public ModelData(String val, String label, String pcId, String snId) {
			this.value = val;
			this.label = label;
			this.submitNotif = snId;
			this.panelConfId = pcId;
		}
```

```
        //TODO GENERATE GETTERS/SETTERS
    }
}
```

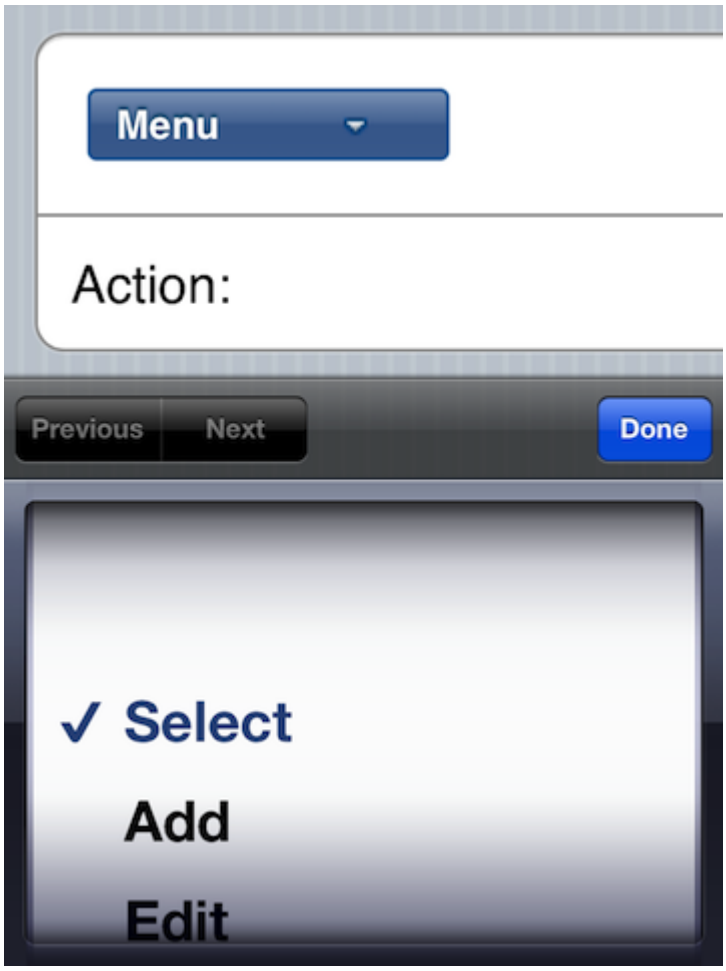
Add action output

Add the following <mobi:fieldSetRow> below the row containing our panelConfirmation and submitNotification components:

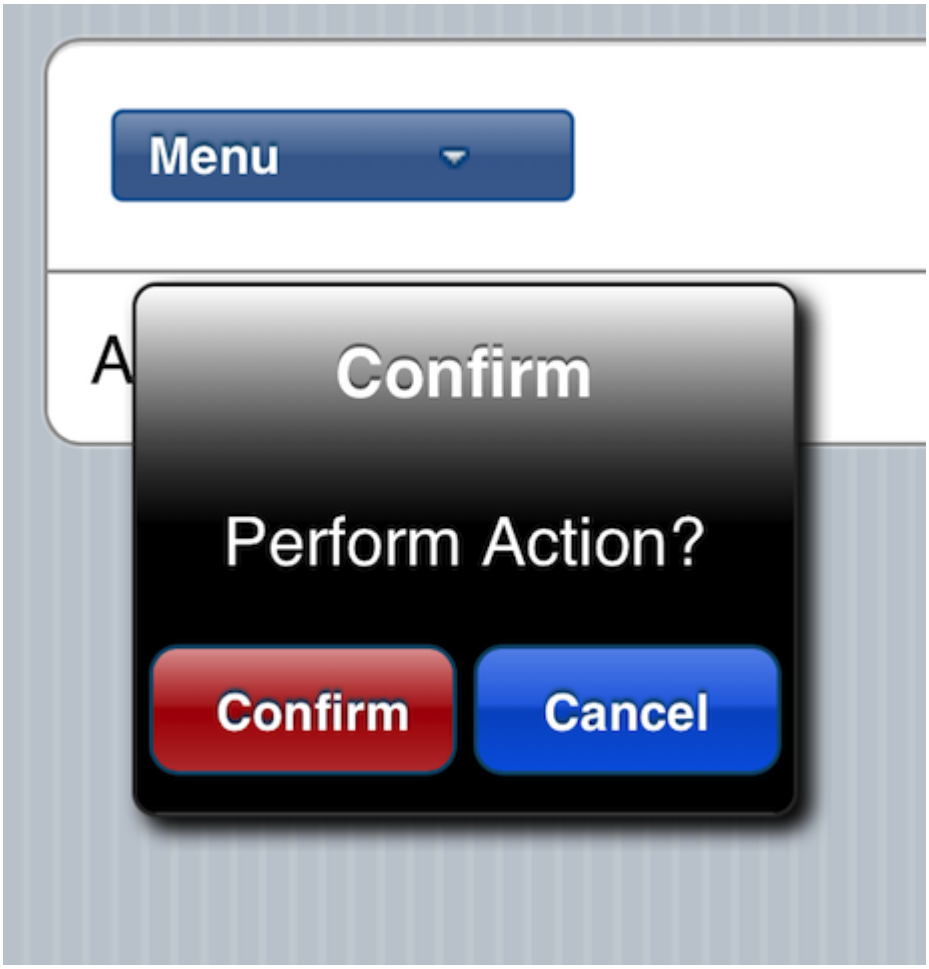
```
<mobi:fieldsetRow>
Action: <h:outputText value="#{menu.action}"/>
</mobi:fieldsetRow>
```

Run Application

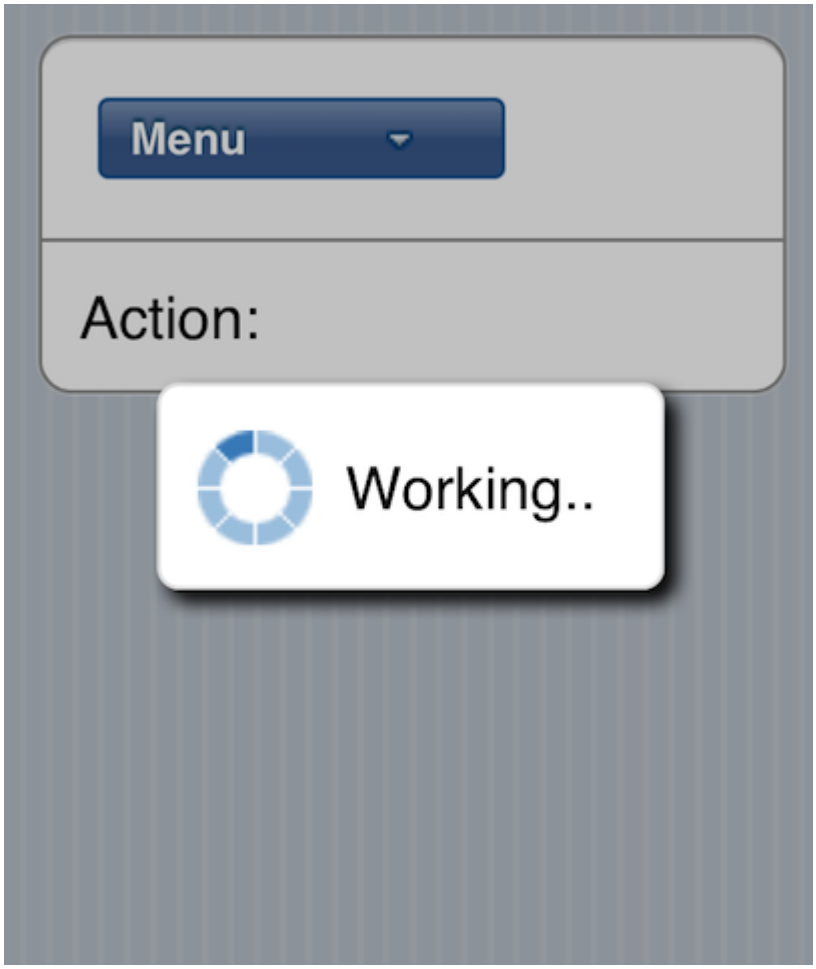
Select an item from the <mobi:menuButton> drop down:



Click to confirm or cancel the <mobi:panelConfirmation>:



The <mobi:submitNotification> will prevent user interaction until our actionListener completes:



Source

1. menu-basic-tutorial.zip
2. menu-advanced-tutorial.zip