

ADF Code Corner

105. How to auto-dismiss af:popup dialogs

ORACLE
CODE CORNER



twitter.com/adfcodecorner

Abstract:

The two use cases for the af:popup are to show content for edit or for information. In the latter case you may want to implement a functionality that auto-hides a popup after a certain period of time. Users should have the option to close the popup by pressing a close icon, a close button, or by a click outside of the popup. This article shows you how to implement auto popup dismissal using ADF Faces declarative on-board functionality.

Author:

Frank Nimphius, Oracle Corporation
twitter.com/fnimphiu
04-DEC-2012


Oracle ADF Code Corner is a loose blog-style series of how-to documents that provide solutions to real world coding problems.

Disclaimer: All samples are provided as is with no guarantee for future upgrades or error correction. No support can be given through Oracle customer support.

Please post questions or report problems related to the samples in this series on the OTN forum for Oracle JDeveloper: <http://forums.oracle.com/forums/forum.jspa?forumID=83>

Introduction

In the sample you can download for this article, a note window (contained in an `af:popup`) is opened by a click on the "Show More" link provided for each table row.

View ▾  Detach

DepartmentId	FirstName	LastName	
50	Donald	OConnell	Show More
50	Douglas	Grant	Show More
10	Jennifer	Whalen	Show More
20	Michael	Hartstein	Show More
20	Pat	Fay	Show More
40	Susan	Mavris	Show More
70	Hermann	Baer	Show More
110	Shelley	Higgins	Show More
110	William	Gietz	Show More
90	Stevens	King	Show More
30	Neena	Kochhar	Show More
90	Lex	De Haan	Show More
60	Alexander	Hunold	Show More
60	Bruce	Ernst	Show More

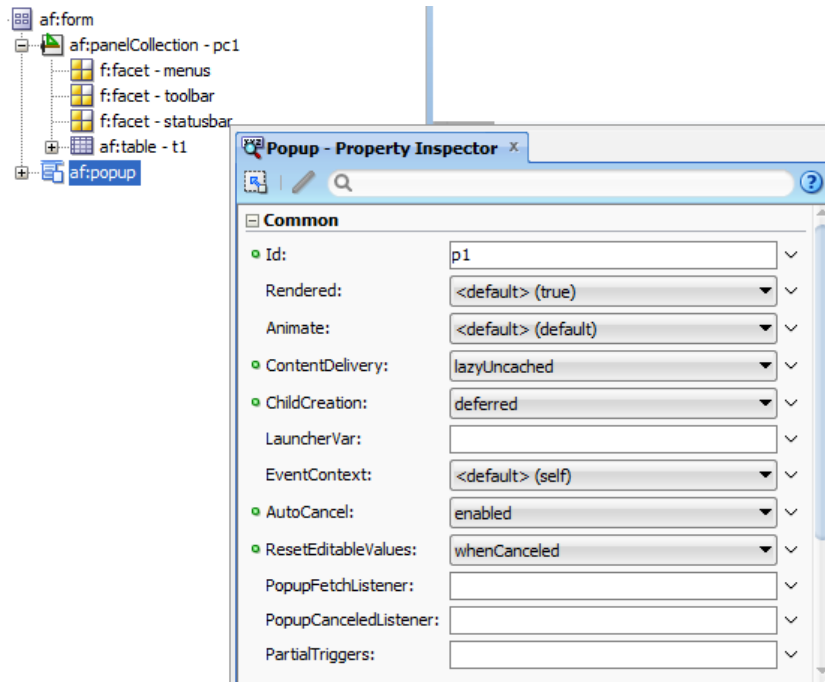
The note window shows additional information for the selected table row. If the user does not click outside of the notw window, the window closes within 5 seconds enforced by a one-time poll.

DepartmentId	FirstName	LastName	
50	Donald	OConnell	Show More
50	Douglas	Grant	Show More
10	Jennifer	Whalen	Show More
20	Michael	Hartstein	Show More
20	Pat	Fay	Show More
40	Susan	Mavris	Show More
70	Hermann	Baer	Show More
110	Shelley	Higgins	Show More
110	William	Gietz	Show More
90	Stevens	King	Show More
30	Neena	Kochhar	Show More
90	Lex	De Haan	Show More
60	Alexander	Hunold	Show More
60	Bruce	Ernst	Show More
60	David	Austin	Show More
60	Valli	Pataballa	Show More
60	Diana	Lorentz	Show More
100	Nancy	Greenberg	Show More
100	Daniel	Faviet	Show More
100	John	Chen	Show More
100	Ismael	Sciarra	Show More

EmployeeId 206
 FirstName William
 LastName Gietz
 Email WGIETZ
 PhoneNumber 515.123.8181
 HireDate 07/06/2002
 JobId AC_ACCOUNT
 Salary 8300
 CommissionPct
 ManagerId 205
 DepartmentId 110

Implementation

As mentioned earlier, the implementation is declarative and does **not** require client side JavaScript. The `af:popup` component that hosts the `af:noteWindow` component is configured to refresh its content upon opening (`ContentDelivery > lazyUncached`). Unless the information in a dialog doesn't change, the **lazyUncached** value is what you always should use. The child creation of the popup is set to "deferred" for better dialog opening performance. Using this setting, the content is fetched in subsequent partial requests.



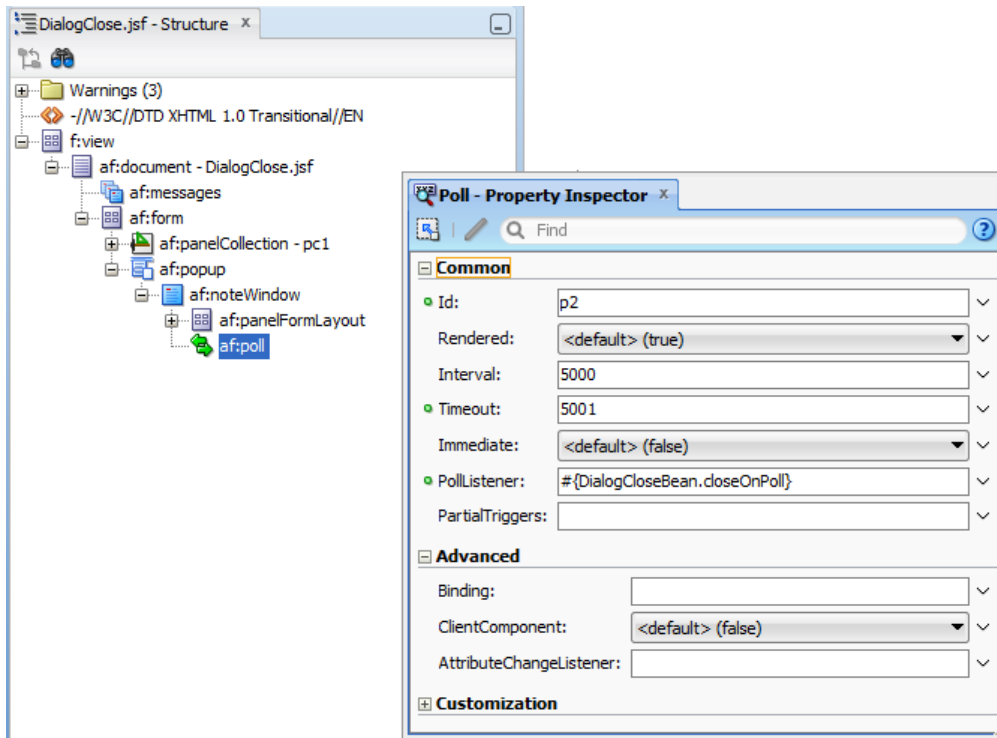
Set the **AutoCancel** property to enabled allows the user to close the note window by clicking anywhere outside of the dialog.

Note: if you use `af:dialog` instead of `af:noteWindow` you can also use the dialog listeners or the OK and CANCEL buttons to allow users to actively close the dialog.

Also note the **ResetEditableValues** property which is set to **whenCancelled**. If the popup content wasn't read only then setting the **ResetEditableValues** property ensures data to be released when the dialog closes so that next time you open the dialog, it displayed the records of the newly selected row.

The `af:poll` component, shown in the image below, is added to the `af:popup` component, which implicitly means that the poll is not started before the popup opens (one of the benefits you get from deferred child loading).

The important property settings of the `af:poll` dialog are **Interval**, which you use to determine how long the dialog should stay open, **Timeout**, which you set slightly larger than Interval to ensure a one-time poll and **PollListener**, which references a managed bean that closes the popup using a component binding created from the `af:popup` "Bindings" property to this managed bean (request scope or backing bean scope for that reason).



The managed bean code of interest is shown below

```
public void closeOnPoll(PollEvent pollEvent) {
    handleClose();
}

//close dialog in private method so you can reuse logic
//if content e.g. is shown in dialog so popup can be closed
//by buttons
private void handleClose() {
    employeePopup.hide();
}
```

Download

The sample to this article can be downloaded as sample 105 from the ADF Code Corner Website

<http://www.oracle.com/technetwork/developer-tools/adf/learnmore/index-101235.html#CodeCornerSamples>

The sample comes as a JDeveloper 11.1.2.3 workspace for which you need to configure the database connection to point to the HR schema of a local database of yours

RELATED DOCUMENTATION

<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	