

# Launching a Modal Popup Window from Server Code

Christian Wenz

## Overview

The ModalPopup control in the AJAX Control Toolkit offers a simple way to create a modal popup using client-side means. However some scenarios require that the opening of the modal popup is triggered on the server-side.

## Steps

First of all, an ASP.NET Button web control is required to demonstrate how the ModalPopup control works. Add such a button within the <form> element on a new page:

```
<asp:Button ID="ClientButton" runat="server" Text="Launch Modal  
  Popup (Client)" />
```

Then, you need the markup for the popup you want to create. Define it as an **<asp:Panel>** control and make sure that it includes a Button control. The ModalPopup control offers the functionality to make such a button close the popup; otherwise there is no easy way to let it vanish.

```
<asp:Panel ID="ModalPanel" runat="server" Width="500px">  
  ASP.NET AJAX is a free framework for quickly creating a new  
  generation of more efficient,  
  more interactive and highly-personalized Web experiences that  
  work across all the  
  most popular browsers.<br />  
  <asp:Button ID="OKButton" runat="server" Text="Close" />  
</asp:Panel>
```

Next add the ModalPopup control from the ASP.NET AJAX Toolkit to the page. Set properties for the button which loads the control, the button which makes it disappear, and the ID of the actual popup.

```
<ajaxToolkit:ModalPopupExtender ID="mpe" runat="server"  
  TargetControlId="ClientButton" PopupControlID="ModalPanel"  
  OkControlID="OKButton" />
```

As with all web pages based on ASP.NET AJAX; the Script Manager is required to load the necessary JavaScript libraries for the different target browsers:

```
<asp:ScriptManager ID="asm" runat="server" />
```

Run the example in the browser. When you click on the button, the modal popup appears. In order to achieve the same effect using server-side code, a new button is required:

```
<asp:Button ID="ServerButton" runat="server" Text="Launch Modal  
Popup (Server)" OnClick="ServerButton_Click" />
```

As you can see, a click on the button generates a postback and executes the **ServerButton\_Click()** method on the server. In this method, a JavaScript function called **launchModal()** is executed—to be exact, the JavaScript function will be executed once the page has been loaded:

```
<script runat="server">  
protected void ServerButton_Click(object sender, EventArgs e)  
{  
    ClientScript.RegisterStartupScript(this.GetType(), "key",  
    "launchModal();", true);  
}  
</script>
```

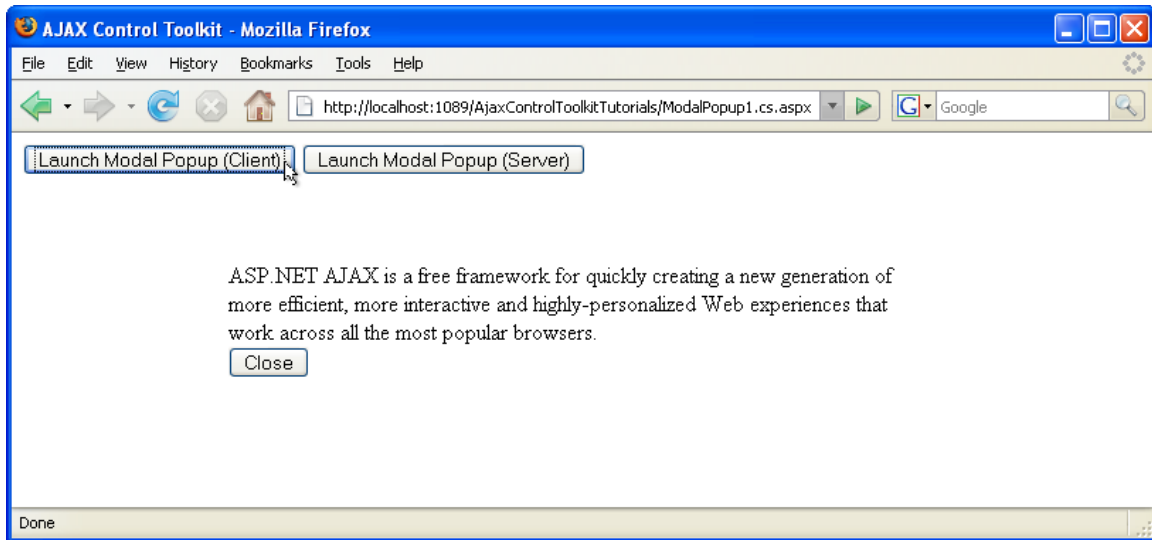
The job of **launchModal()** is to display the ModalPopup. The **launchModal()** function is executed once the complete HTML page has been loaded. At that moment, however, the ASP.NET AJAX framework has not been fully loaded yet. Therefore, the **launchModal()** function just sets a variable that the ModalPopup control must be shown later on:

```
<script type="text/javascript">  
var launch = false;  
function launchModal()  
{  
    launch = true;  
}
```

The **pageLoad()** JavaScript function is a special function that gets executed once ASP.NET AJAX has been fully loaded. Therefore we add code to this function to show the ModalPopup control, but only if **launchModal()** has been called before:

```
function pageLoad() {  
    if (launch)  
    {  
        $find("mpe").show();  
    }  
}  
</script>
```

The **\$find()** function is looking for a named element on the page and expects the server-side ID as a parameter. Therefore, **\$find("mpe")** returns the client representation of the ModalPopup control; its **show()** method lets the popup appear.



**The modal popup appears when either of the buttons is clicked**