Mitch Meyer

Assignment 8.2

Assessing DataList and GridView controls

ASP.net is a powerful web development language and provides many tools for a developer to manipulate and customize data to be displayed on the Internet. The DataList and GridView controls are two tools that make use of HTML tables to display information stored on a database. Both make use of the DataBind() method to bind a data source object to the controls. They are also both derived from the WebControl class, meaning that stylistic settings such as BackColor, ForeColor, CssClass, and BorderStyle can be customized. These similarities aside, each control has advantages and disadvantages so neither one is the perfect solution for every situation.

The GridView control is very versatile in its features as it comes with many functions that greatly increase the usability of the data being displayed. GridView has an AllowSorting property that allows users to sort the data displayed in its table by clicking on the header of the column to be sorted by. The column clicked on first sorts the data items ascendingly and with the second click of the column header, the items are sorted as descending. GridView also allows a developer to allow for paging of the data to prevent extraordinary page load times and it also allows for inline editing of the data being displayed. GridView also has a relatively short development time as all that is necessary is to add the DataGrid to the webpage and to write two lines of code binding the DataSource to the grid and calling the method to populate the data to the grid. Of course development time increases as features are added to the GridView controls such as paging and sortability. These features grant great usability to the GridView controls but there are some disadvantages to using this control.

There are two main disadvantages to using GridView to display data stored in a database. Gridview is very limited in its customization of the rendered HTML markup. There is some flexibility in adjusting the font of the text displayed by the GridView and by adjusting the size of the columns, but ultimately, data displayed in GridView will always be displayed as an HTML table. The second disadvantage to using GridView is that it is has the worst performance of the three data web controls in ASP.Net and the ViewState produced by the GridView can be very large in the case of a dataset with a large number of rows.

DataList controls can be used to display data just like GridView controls are used, but DataList controls give a developer the ability to use templates to determine how the data will be displayed. This gives a developer a much wider range of options by allowing them to specify a mix of both HTML syntax and databinding syntax. There are seven templates that DataList supports including ItemTemplate, AlternateEndingTemplate, EditItemTemplate, FooterTemplate, HeaderTemplate, SelectedItemTemplate, and SeparatorTemplate. By default, the DataList displays each item as a row in an HTML table, but that can be changed to have multiple DataItems displayed in a single row or to have the data displayed using span tags instead. DataList provides better performance over GridView and also supports inline data editing as well as data paging and sorting. These functions are substantially more difficult to implement in a DataList than in a GridView. Paging and column sorting is not a built in feature for the DataList controls. The developer would have to custom code those functions in order to create that functionality while maintain the customization available using DataList.

When it comes down to choosing how to display data on an ASP.Net web page, it would be wise to consider a series of questions about what that data requires. A prudent developer should consider the potential volume of the site being developed. Will the data being displayed be accessed by many users creating a performance concern? Will the users need to be able to sort through the data displayed? How much development time is available to this project? By thoroughly considering the various needs of the users and the recipients of the completed project, a developer can make a proper decision. For a project with a limited time frame, that needs sort and paging functionality, a GridView approach would be the best choice. For a project that requires an eye catching look but the user doesn’t need to be able to manipulate the data, the DataList would the method of choice. There are many other combinations of circumstances that would make either tool seem like the proper choice for the job, but ultimately, it is up to the developer and what they are most comfortable working with.

Source

Deciding When to Use the DataGrid, DataList or Repeater. (n.d.). Retrieved October 7, 2015, from https://msdn.microsoft.com/en-us/library/aa479015.aspx

Delamater, M., & Boehm, A. (2013). *Murach's ASP.NET 4.5 Web programming with C# 2012* (5th ed.). Fresno, Calif.: Mike Murach & Associates.