

Introducing ViewState

Mohamad Halabi

Microsoft Integration MVP

Ex- ASP.NET MVP

www.pluralsight.com



Contents

- ➡ ▪ **Introducing ViewState**
- ➡ ▪ **ViewState for ASP.NET**

Introducing ViewState

- **ViewState is key/value pair indexer:**

```
ViewState["x"] = 1;  
ViewState["y"] = "str";
```

- **ViewState is a property of System.Web.UI.Control**
 - All ASP.NET controls and pages derive from System.Web.UI.Control
- **ViewState is of type System.Web.UI.StateBag**

```
...protected virtual StateBag ViewState { get; }
```

- **StateBag class can “Track” changes**
 - Tracking enabled by calling “TrackViewState()”
 - A tracked member is said to be “Dirty”
- **Why is this tracking behavior?**
 - This will become clear in the Page lifecycle

ViewState for ASP.NET

- Technique used by ASP.NET to remember change of state between multiple requests
- Saved in __VIEWSTATE hidden HTML field
- Two serialized data types are stored in __VIEWSTATE:
 - Programmatic (not Static!) change of page and control state (i.e. "Dirty" items)
 - Explicit data stored using ViewState[""] indexer

ViewState for ASP.NET, Cont'd

- ASP.NET Server Controls use ViewState to store property values
- Ex: Text property of TextBox control:

```
public virtual string Text
{
    get
    {
        return (string) this.ViewState["Text"] ?? string.Empty;
    }
    set
    {
        this.ViewState["Text"] = (object) value;
    }
}
```

- Note: this does NOT mean that value is stored in __VIEWSTATE.
- For example: `<asp:TextBox runat="server" ID="txt" Text="sometext" />`
 - "sometext" is not stored in __VIEWSTATE. Design-time data is NOT serialized into __VIEWSTATE