## **管理状态**

超文本传输协议(HTTP)是一种无状态协议。当客户端从服务器断开连接时，ASP.NET 引擎将丢弃页面对象。这样一来，每个 Web 应用程序能够扩展到同时用于大量请求，但是不会耗尽服务器内存。

ASP.NET 管理四种状态：

* 视图状态 viewstate
* 控制状态
* 会话状态 session
* 应用程序状态

## **视图状态**

视图状态是页面及其所有控件的状态。它通过 ASP.NET 框架的反馈保持不变。

ViewState类具有以下属性：

| **属性** | **描述** |
| --- | --- |
| Item(name) | 具有指定名称的视图状态的值，是 StateBag 的默认属性。 |
| Count | 状态集合中的项目名称。 |
| Keys | 集合中所有项目的密钥集合。 |
| Values | 集合中所有项目的值的集合。 |

StateBag 类具有以下方法：

| **方法** | **描述** |
| --- | --- |
| Add(name, value) | 添加一个项目到视图状态集合，更新现有项目。 |
| Clear | 移除集合中所有项目。 |
| Equals(Object) | 确定指定的对象是否等于当前对象。 |
| Finalize | 允许释放资源并执行其他清理操作。 |
| GetEnumerator | 返回存储在 StateBag 对象中重复的 StateItem 对象的密钥/值对的计数器。 |
| GetType | 获取当前实例的类型。 |
| IsItemDirty | 检查存储在 StateBag 对象以确认其是否已被修改。 |
| Remove(name) | 移除制定项目。 |
| SetDirty | 设置 StateBag 对象的状态以及每个由其包含的 StateItem 对象的 Dirty 属性。 |
| SetItemDirty | 为在 StateBag 对象中的指定 StateItem 对象设置 Dirty 属性。 |
| ToString | 返回代表状态包对象的字符串。 |

该实例的后台代码文件如下所示：

**public** partial **class** \_Default : System.Web.UI.Page

{

**public** **int** counter

{

get

{

**if** (ViewState["pcounter"] != null)

{

**return** ((**int**)ViewState["pcounter"]);

}

**else**

{

**return** 0;

}

}

set

{

ViewState["pcounter"] = value;

}

}

**protected** **void** **btnIncrement\_Click**(object sender, EventArgs e)

{

lblCounter.Text = counter.ToString();

counter++;

}

}

## **控制状态**

控制状态不能被直接修改，存取或禁用。

## **会话状态**

HttpSessionState 类具有以下属性：

| **属性** | **描述** |
| --- | --- |
| SessionID | 唯一的会话标识符。 |
| Item(name) | 具有指定名称的会话状态项的值，是 HttpSessionState 类的默认属性。 |
| Count | 会话状态集合中项的数量。 |
| TimeOut | 获取和设置时间量，几分钟内，在供应商停止会话状态前在请求间被允许。 |

HttpSessionState 类有以下方法：

| **方法** | **描述** |
| --- | --- |
| Add(name, value) | 添加新的项到会话状态集合。 |
| Clear | 移除会话状态集合中所有项。 |
| Remove(name) | 移除会话状态集合中的指定项。 |
| RemoveAll | 移除会话状态集合中所有密钥和值。 |
| RemoveAt | 从会话状态集合中删除指定索引处的项。 |

## **应用程序状态**

HttpApplicationState 类具有以下属性：

应用程序状态主要被用于存储计数器，其他统计数据及税率，折扣率等所有应用程序数据，并存储用户访问网站的路径。

| **属性** | **描述** |
| --- | --- |
| Item(name) | 具有指定名称的应用程序项的值，是 HttpApplicationState 的默认属性。 |
| Count | 应用程序状态集合中项的数量。 |

HttpApplicationState 类具有以下方法：

| **方法** | **描述** |
| --- | --- |
| Add(name, value) | 添加新的项目到应用程序状态集合。 |
| Clear | 移除应用程序状态集合中的所有项。 |
| Remove(name) | 移除应用程序状态集合中的指定项。 |
| RemoveAll | 移除一个 HttpApplicationState 集合中所有对象。 |
| RemoveAt | 移除从由索引找到的集合中的一个 HttpApplicationState 对象。 |
| Lock() | 锁定应用程序状态集合以便只有当前用户可以访问。 |
| Unlock() | 解锁应用程序状态集合以便所有用户可以访问。 |

应用程序状态的数据通常是由为事件编写的处理程序维护：

* 应用程序开启
* 应用程序结束
* 应用程序错误
* 会话开始
* 会话结束

以下代码片段展示了用于存储应用程序状态信息的基本语法：

Void **Application\_Start**(**object** sender, EventArgs e){

Application["startMessage"] = "The application has started.";

}

Void **Application\_End**(**object** sender, EventArgs e){

Application["endtMessage"] = "The application has ended.";

}