## **ASP.NET 应用程序生命周期**

应用程序生命周期有以下阶段：

* 用户请求访问应用程序的资源，即一个页面。浏览器发送此请求到 web 服务器。
* 一个统一管道接收第一个请求，并发生以下事件：
  + 一个 ApplicationManager 类的对象创建。
  + 一个 HostingEnvironment 类的对象创建从而提供信息资源。
* 创建响应对象。应用程序对象如 HttpContext，HttpRequest 和 HttpResponse 被创建并初始化。
* 一个 HttpApplication 对象的实例被创建并被分配到请求。
* 请求由 HttpApplication 类所处理。不同的事件引发此类处理请求。

## **ASP.NET 页面生命周期**

当请求一个页面时，页面被加载到服务器内存，然后处理并送达到浏览器中。然后再从内存中卸载。在这些步骤中的每一步，方法和事件都是可用的，可以根据应用程序所需进行改写。换言之，你可以编写自己的代码从而去置换缺省代码。

页面类创建了页面上所有控件的等级树。页面上所有的组件，除了指令，其余都是控件树的一部分。你可以通过在页面指令上添加 trace = "true" 来看到控制树。我们会覆盖页面指令，然后在 'directives' 和 'event handling' 下追踪。

以下是一个 ASP.NET 页面的不同阶段：

* ****页面请求**** - 当 ASP.NET 得到一个页面请求，它决定是否解析和编译页面，或者会有一个页面的缓存版本；相应地进行回应。
* ****页面生命周期的开启**** - 在这个阶段，设置请求和回应对象。如果是一个旧的请求或者是回发的，页面 IsPostBack 属性设置为正确。页面 ULCulture 属性同样也被设置。
* ****页面初始化**** - 在此阶段，页面上的控件通过设置 UniqueID 属性被分配到独特的 ID 并应用主题。对于一个新的请求，加载回发数据并且控件属性被重新储存到视图状态下的值。
* ****页面加载**** - 在此阶段，设置控件属性，使用视图状态和控件状态值。
* ****验证**** - 调用验证控件的校验方法并成功执行，页面的 IsValid 属性设置为 true。
* ****回发事件处理**** - 如果请求是一个回发（旧请求），相关事件处理程序被调用。
* ****页面显示**** - 在此阶段，页面的视图状态和所有控件被保存。页面为每一个控件调用显示方法，并且呈现的输出被写入页面响应属性中的 OutputStream 类。
* ****卸载**** - 显示过的页面被送达客户端以及页面属性，例如响应和请求，被卸载并全部清除完毕。

## **ASP.NET 页面生命周期事件**

在页面生命周期的每一阶段，页面引发一些事件，会被编码。一个事件处理程序基本上是一个函数或子程序，绑定到事件，使用声明式如 OnClick 属性或处理。

以下是页面生命周期事件：

* ****PreInit**** - PreInit 是页面生命周期的第一个事件。它检查 IsPostBack 属性并决定页面是否是回发。它设置主题及主版页，创建动态控件，并且获取和设置值配置文件属性值。此事件可通过重载 OnPreInit 方法或者创建一个 Page\_PreInit 处理程序进行处置。
* ****Init**** - Init 事件初始化控件属性，并且建立控件树。此事件可通过重载 OnInit 方法或者创建一个 Page\_Init处理程序进行处置。
* ****InitComplete**** - InitComplete 事件允许对视图状态的跟踪。所有的控件开启视图状态下的跟踪。
* ****LoadViewState**** - LoadViewState 事件允许加载视图状态信息到控件中。
* ****LoadPostData**** - 在此阶段期间，对所有由 \ 标签定义的输入字段的内容进行处理。
* ****PreLoad**** - PreLoad 在回发数据加载在控件中之前发生。此事件可以通过重载 OnPreLoad 方法或者创建一个 Pre\_Load 处理程序进行处置。
* ****Load**** - Load 事件被页面最先引发，然后递归地引发所有子控件。控件树中的控件就被创建好了。此事件可通过重载 OnLoad 方法或者创建一个 Page\_Load 处理程序来进行处置。
* ****LoadComplete**** - 加载进程完成，控件事件处理程序运行，页面验证发生。此事件可通过重载 OnLoad 方法或者创建一个 Page\_LoadComplete 处理程序来进行处置。
* ****PreRender**** - PreRender 事件就在输出显示之前发生。通过处理此事件，页面和控件可以在输出显示之前执行任何更新。
* ****PreRenderComplete**** - 当 PreRender 事件为所有子控件被循环引发，此事件确保了显示前阶段的完成。
* ****SaveStateComplete**** - 页面控件状态被保存。个性化、控件状态以及视图状态信息被保存。
* ****UnLoad**** - UnLoad 阶段是页面生命周期的最后一个阶段。它为所有控件循环引发 UnLoad 事件，最后为页面自身引发。最终完成清理和释放所有资源和引用，比如数据库连接。此事件可通过调整 OnLoad 方法或者创建一个 Page\_UnLoad 处理程序来进行处置。