

您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

ASP.NET 页面框架

讲师：邵志东

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

课前准备

- Dot Net Framework
- VS.NET 2002/2003
- C#/VB.NET
- Level 200

议程

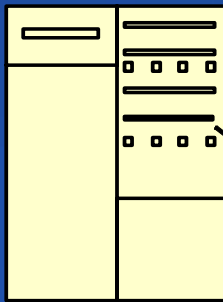
- ASP.NET 页面简介
- 页面生存周期
- 页面事件
- 页面指令
- 内联编程模型与代码隐藏编程模型

ASP.NET工作原理

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

Web 服务器



②Web 服务寻找指令文件(.aspx)

③ASP.NET 代码被发送给公共语言运行时进行编译

④HTML 流返回给浏览器和指令

①客户请求 Web 页

⑤浏览器处理 HTML 并显示页面



客户机

aspnet_isapi.dll

MSDN Webcasts

ASP.NET 页面

- ASP.NET 页框架是一种编程框架，它在 Web 服务器上运行以动态地生成和管理 Web 窗体页。可以使用 Web 窗体页来创建可编程的 Web 页，这些 Web 页用作 Web 应用程序的用户界面。
- Web 窗体页在任何浏览器或客户端设备中向用户提供信息，并使用服务器端代码来实现应用程序逻辑。
- Web 窗体页输出几乎可以包含任何支持 HTTP 的语言（包括 HTML、XML、WML 和 ECMAScript (JScript, JavaScript)）。

Web窗体页的特点

- 基于 Microsoft ASP.NET 技术。在该技术中，在服务器上运行的代码动态地生成到浏览器或客户端设备的 Web 页输出。
- 兼容所有浏览器或移动设备。Web 窗体页自动为样式、布局等功能呈现正确的、符合浏览器的 HTML。
- 兼容 .NET 公共语言运行库所支持的任何语言
- 基于 Microsoft .NET 框架生成
- 在 Visual Studio 中通过强大的快速应用程序开发 (RAD) 工具受到支持
- 具有灵活性，因为您可以向它们添加用户创建的控件和第三方控件。

Web窗体页的组成



单文件 Web 窗体页

- 除了由 **.aspx** 文件和单独的类文件组成的页之外，**ASP.NET** 结构还支持“单文件”模型，在此模型中，UI 元素和代码位于同一文件中。
- 单文件页的处理方式存在一些不同之处：
 - 不会将该页的代码编译成一个单独的类，然后从该类派生 **.aspx** 文件。相反，**.aspx** 文件将直接从 **Page** 类派生。
 - 当部署该页时，由于源代码在物理上位于 **.aspx** 文件中，所以源代码将与 Web 窗体页一起部署

您的潜力. 我们的动力

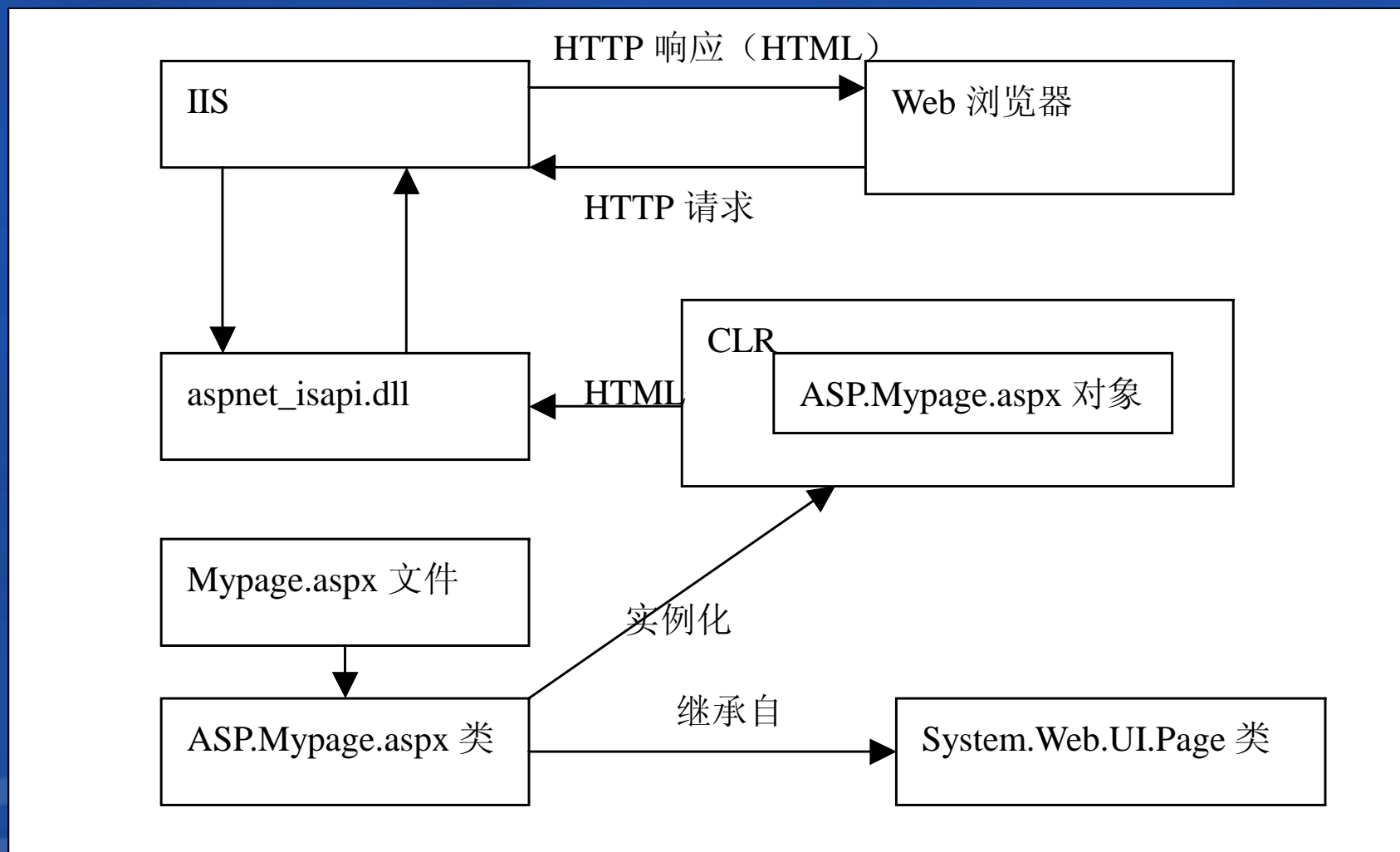
Microsoft
微软(中国)有限公司

议程

- ASP.NET页面简介
- 页面生存周期
- 页面事件
- 页面指令
- 内联编程模型与代码隐藏编程模型

ASP.NET页面的执行过程

Microsoft®
微软(中国)有限公司



WEB页面处理过程

- 页面的一次往返处理 :用户对**Server Control**的一次操作, 就可能引起页面的一次往返处理: 页面被提交到服务器端, 执行响应的事件处理代码, 重建页面, 然后返回到客户端
- 页面重建 :每一次页面被请求, 或者页面事件被提交到服务器, **asp.net**运行环境将执行必要的代码, 重建整个页面, 把结果页面送到浏览器, 然后抛弃页面的变量、控件的状态和属性等等页面信息。

议程

- ASP.NET 页面简介
- 页面生存周期
- 页面事件
- 页面指令
- 内联编程模型与代码隐藏编程模型

页面事件

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

以下列表按激发顺序提供运行时连线的代理实例:

- **Page_Init**: 初始化值或连接
- **Page_Load**: 出现此事件期间, 您可以执行一系列的操作来首次创建 **ASP.NET** 页面或响应由投递引起的客户端事件。在此事件之前, 已还原页面和控件视图状态。使用 **IsPostBack** 页面属性检查是否为首次处理该页面。如果是首次处理, 请执行数据绑定。此外, 请读取并更新控件属性。
- **Page_DataBind**: 在页面级别调用 **DataBind**, 也可在单个控件上调用
DataBindPage_PreRender: 恰好在保存视图状态和呈现控件之前激发 **PreRender** 事件。
- **Page_Unload**: 此事件是执行最终清理工作的合适位置。

非确定性事件

- **Page_Error:** 如果在页面处理过程中出现未处理的例外, 则激发 **Error** 事件。错误事件为您提供了解决处理错误的机会。
- **Page_AbortTransaction:** 如果要指明交易是成功还是失败, 交易事件非常有用。此事件通常用于购物车方案, 其中此事件可以指示订购是成功还是失败。如果已终止交易, 则激发此事件。
- **Page_CommitTransaction:** 如果已成功提交交易, 则激发此事件。

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

DEMO1

公用外观示例

Global.asax

Global.asax是干什么的？

- ◆ Global.asax文件也称为ASP.NET应用程序文件，它一般被放在根目录下。此文件中的代码不产生用户界面，也不相应单个页面的请求。
- ◆ 它主要是负责处理Application_Start, Application_End, Session_Start和Session_End事件的。

Application和Session事件

- 当Application 对象的生命周期开始时, Application_Start 事件会被启动, 当Application 对象的生命周期结束时Application_End 事件会被启动
- 当有一个新用户访问应用程序时, 就会立刻触发 Session_Start事件。当某个用户停止了访问或者程序执行了Session.Abandon方法, 就会触发 Session_End事件。
- 一个Application_End事件肯定发生在 Session_End事件之后, Application_End事件只有在服务器停止工作或Application_End事件卸载时才触发。

Global.asax中的事件

Microsoft
微软(中国)有限公司

事件函数	描述
Application_Start	应用程序启动时激发
Application_BeginRequest	HTTP请求开始时激发
Application_AuthenticateRequest	应用程序批准HTTP请求时激发
Session_Start	会话启动时激发
Application_EndRequest	HTTP请求结束时激发
Session_End	会话结束时激发
Application_End	应用程序结束时激发
Application_Error	发生错误时激发

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

DEMO2

网站计数器

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

议程

- ASP.NET 页面简介
- 页面生存周期
- 页面事件
- 页面指令
- 内联编程模型与代码隐藏编程模型

页面指令

- 页面指令指定页面编译器在处理文件时使用的可选设置。页面指令位于页面文件的顶部，使用以下语法：
 - `<%@ directive {attribute=value}* %>`
- ASP.NET 页面支持下列指令：
 - **@ Page**
 - **@ Implements**
 - **@ Import**
 - **@ Register**
 - **@ Assembly**
 - **@ Reference**
 - **@ OutputCache**

@import

- 命名空间的导入
- 示例:
 - `<%@ import namespace="System.NET"%>`
 - 导入类库
 - 有点类似于 `#include <stdio.h>`

@register

- 自定义控件声明
- 示例如下:
 - `<%@ Register TagPrefix="saidy" TagName="info" Src="con01.ascx" %>`

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

DEMO3

@register示例

@page

- 语法形式如下:
 - `<%@ Page Language="VB"
ContentType="text/html"
ResponseEncoding="gb2312" %>`
- 参数
 - Language
 - 示例: language="vb"
 - 示例: language="c#"

@page

- 参数

- **Debug**

- 指示是否应使用调试符号编译该页。如果应使用调试符号编译该页，则为 **true**；否则为 **false**。
 - 示例: debug="true"

- **Buffer**

- 确定是否启用 HTTP 响应缓冲。如果启用页缓冲，则为 **true**；否则为 **false**。默认值为 **true**。
 - 示例: buffer=true

@page

- 参数

- **AutoEventWireup**

- 指示页的事件是否自动连网。如果启用事件自动连网，则为 **true**；否则为 **false**。默认值为 **true**。

- **ErrorPage**

- 定义在出现未处理页异常时用于重定向的目标 URL。
 - `Errorpage=error.htm`

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

DEMO3

@page 示例

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

议程

- ASP.NET 页面简介
- 页面生存周期
- 页面事件
- 页面指令
- 内联编程模型与代码隐藏编程模型

内联编程模型与代码隐藏编程模型

ASP.NET 支持两种模式的页面开发:

- 在 **.aspx** 文件的 **<script runat=server>** 块内写入页面逻辑代码, 然后在服务器上首次请求该页面时动态编译代码。
- 在外部类中写入页面逻辑代码, 在服务器上进行部署之前编译该外部类, 并且运行时在 **.aspx** 文件“背后”链接该类。

内联编程模型的限制

- 不能直接在 Visual Studio 中创建单文件 Web 窗体页
- 不能通过从工具箱中拖动将非可视组件（如数据组件）添加到该页上
- 在 HTML 视图中而不是在代码编辑器中编写代码。
- 您得不到语法检查或语句结束、制表位设置或代码格式设置等功能。
- 必须手动将事件绑定到事件处理程序
- 不支持某些调试功能
- 在运行该页之前不会捕获编译时错误。

CodeBehind技术

- Code Behind就是所谓的代码分离，自从Microsoft公司推出了ASP.NET以后，Code Behind就是一个热门的话题。在一般的ASP.NET文件中，Code Behind主要是用二个文件来创建一个ASP.NET的页面，其中一个设计文件，一般以.aspx或者.ascx做为扩展名，而另外一个程序代码文件，一般以.vb或者.cs做为扩展名，其程序设计语言主要是VB.Net或者是C#。

CodeBehind的优点

- Code Behind把界面设计代码和程序设计代码以不同的文件分开，对于代码的重复使用，程序的调试和维护都是革命性的
- 当发布网站的时候，可以利用这种技术来有效的保护代码。这对于程序的安全性是一个提高。

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

DEMO4

内联编程模型示例 CodeBehind技术示例

小结

- ASP.NET页面简介
- 页面生存周期
- 页面事件
- 页面指令
- 内联编程模型与代码隐藏编程模型

您的潜力，我们的动力


Microsoft
微软(中国)有限公司

更多信息.....


- MSDN网站: <http://msdn.microsoft.com>
- 程序员大本营: www.csdn.net

Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问 

提问(A)

删除(D)

问题管理器(Q)

您的潜力，我们的动力

Microsoft®
微软(中国)有限公司

Microsoft®

msdn


MSDN Webcasts