
第 25 章 新闻模块设计

现在的大部分网站都需要使用新闻模块进行网站信息交流，新闻模块是网站之中最传统的交流模块。管理人员能够通过后台进行新闻的发布和修改，用户就能够在前台页面中进行新闻的访问和评论，新闻模块是网站必不可少的模块，例如新浪、腾讯、搜狐等大型网站都离不开新闻模块。

25.1 学习要点

新闻模块需要涉及到一些 ASP.NET 3.5 的基本知识，如果要仔细学习新闻模块的开发，需要详细了解本书的一些章节知识，这些章节如下所示：

- ☐ ASP.NET 的网页代码模型。
- ☐ Web 窗体基本控件。
- ☐ 数据库基础。
- ☐ ADO.NET 常用对象。
- ☐ Web 窗体数据控件。
- ☐ ASP.NET 内置对象。
- ☐ 生成静态的概念

基本了解了以上章节的知识点后，就能够熟练学习和开发此模块。

25.2 系统设计

新闻模块对于网站而言是非常重要的，虽然今天 Web 2.0 的概念大行其道，但是新闻还是作为网站应用的基础内容而存在，新闻能够提供最简单的用户信息交互，对于新闻信息的筛选和投放同样能够吸引访问者。

25.2.1 模块功能描述

新闻模块对于网站开发而言是最简单也是最重要的，对于网站而言，作为一个信息媒体，需要向用户，也就是网站的使用者进行信息传递。现在的各大门户网站，如新浪、腾讯和搜狐等，依旧使用的是新闻作为网站主导，而对于大行其道的 Web 2.0，同样也是基于新闻模块的形式进行信息呈现。

新闻模块的开发相对于广告模块而言从技术上实现比较的简单，并没有广告模块实现起来复杂和繁琐，也不需要使用自定义控件。但是新闻模块如果要制作好，还是有一定的难度的，其最主要的难度就在于生成静态和伪静态化。

对于不需要生成静态或伪静态化的新闻，其功能模块抽象起来比较的简单，在新闻使用之前，管理员可以在后台添加新闻分类，用于分类新闻。在添加新闻分类完毕后，就可以添加新闻并选择相应的分类进行新闻分类，分类后的新闻将能够呈现在不同的页面中以显示不同的分类的新闻。从一定的意义上

来说，新闻模块的功能对于管理员而言，就只是添加分类和发布新闻，如图 25-1 所示。

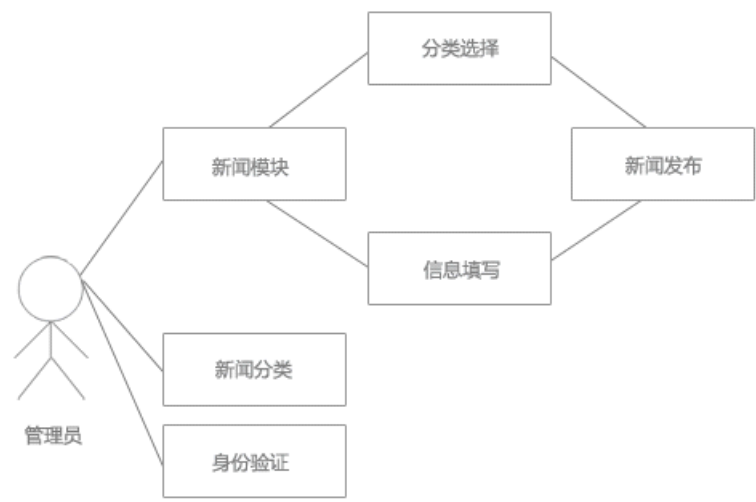


图 25-1 新闻模块基本流程分析

正如图 25-1 所示，管理员能够在后台进行新闻模块中的分类选择和信息填写进行新闻发布，管理员还可以对新闻分类进行管理。但是在管理员进行操作前，首先需要对管理员进行身份验证，以判断管理员是否有合法的权限进行身份验证。

身份验证可以使用登录模块进行身份验证，但是这里的登录模块没有网站的登录控件复杂。这里只需要实现对管理员进行判断，如果判断是管理员则能够通过，如果不是管理员则不允许通过的功能即可。从上述模块功能描述中可以规划成以下几个页面：

- ❑ 登录页面：管理员登录页面，为管理员提供身份验证。
- ❑ 新闻分类添加页面：为管理员提供新闻添加功能。
- ❑ 新闻分类管理页面：为管理员提供新闻分类的添加和管理。
- ❑ 新闻页面：用于显示新闻。
- ❑ 首页调用：用于进行新闻列表的显示，方便用户进行新闻查阅。

这些页面能够为管理员的新闻发布和更新进行操作提供，管理员首先需要在登录页面进行登录操作并进行身份验证。如果验证通过，就能够在新闻分类页面和新闻页面进行新闻分类操作和新闻操作，管理员可以通过新闻分类操作和新闻操作进行新闻的发布和归类，这样有助于在前台的页面中进行调用。

在前台显示中，同样还需要新闻显示页面和首页，新闻显示页面用于显示单个新闻，而首页用于显示新闻相应的列表，如在新浪、腾讯等网站的首页，都是调用最新的一些新闻列表来呈现的，这样有助于用户对新闻信息的筛选和分类。

25.2.2 模块流程分析

在各种类型的网站中，例如腾讯，都可以看到首页被各种新闻版块内容所填充，包括时事、体育、娱乐等等，这些新闻和内容版块都是在后台相关人员进行采编并纳入数据库和页面中的。可以想象，一个大型的门户网站每天会有多少的访问量，如果每次的用户访问都需要从数据库中读取数据，那么一天下来可能有几百万的读取次数，这样无疑会对 Web 应用带来极大的挑战。

可以观察各种门户的新闻，可以看得出来这些门户的新闻的 URL 地址的后缀都是.html 或者是.shtml 的，那么是不是这些网站的开发人员和采编人员当有一条新闻时就手动进行页面编写呢？显然答案是否

定的，新闻网站可以将一些新闻静态化，这样就能够保证服务器只需要承受较少的压力依旧可以承担百万级的访问量。

生成静态就是将数据库中的数据或相应的字段进行静态化，例如将.aspx 页面的文件进行静态化生成成为.html 页面。 .html 是静态页面，当用户访问.html 页面时无需进行数据操作和逻辑操作，对于服务器而言只需要将.html 文本发送到浏览器就能够显示页面的内容。这样无疑增加了访问速度。如果网站要生成静态，其基本模块流程如图 25-2 所示。

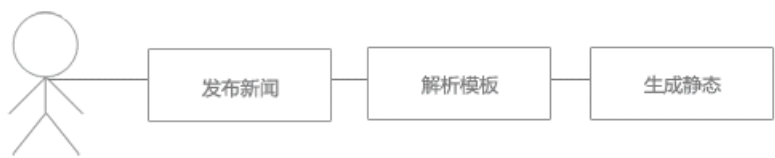


图 25-2 生成静态解决方案

虽然静态化能够降低服务器的压力，但是静态化同样会牺牲很多的空间。如果将新闻数据中的每个数据进行静态化，这也就是说每一条新闻就会生成一个.html 页面，那么有十万条新闻就会生成十万个.html 页面，这对服务器操作系统和 I/O 读写也有更高的要求，如果文件太多，打开文件夹的速度还不如读取数据库。虽然静态化是一个解决方案，但是很多情况下也可以不使用静态化。如果系统不使用静态化，可以使用非静态化的解决方案，如图 25-3 所示。

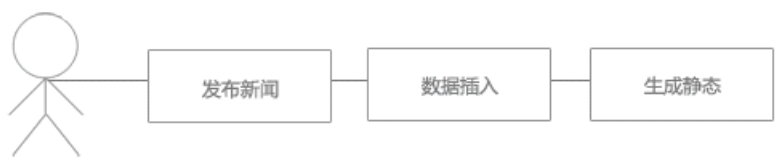


图 25-3 非静态化的解决方案

相比之下，非静态化的解决方案在实现上来说更加容易，因为静态化的实现方案还需要解析模板。在新闻模块的编写中，可以事先考虑是选择静态化的解决方案还是选择非静态化的解决方案，静态化的解决方案和非静态化的解决方案在开发过程中虽然可以替换，但是也有一定的开发风险。

而对于管理员而言，无需关心是否是静态化的解决方案还是非静态化的解决方案。在后台的操作过程中，管理员只关心自己如何能够快速的进行添加新闻和修改新闻等操作，在执行了相应的操作后，管理员就能够在前台进行新闻显示。

25.3 数据库设计

新闻模块同样需要多个表进行新闻描述和新闻操作，同样，为了安全起见和模块的可扩展性，还需要其他的表进行数据存储，这些表能够进行新闻的存储、身份验证、新闻分类的增删以及静态化生成保存等操作。

23.3.1 数据库设计

在新闻模块设计中，需要多个表进行新闻描述，同时为了保证管理用户的安全性，还需要设计管理员表，这些表包括 `news`、`newsclass` 和 `admin` 三个表，这三个表分别存储新闻、新闻分类和管理员信息。在创建表之前，首先需要创建数据库 `news`，创建完成后就能够创建相应的表。在对新闻模块进行流程分析之后，就能够大概的设计出这三个表中所需要的字段，其中 `news` 表所包含的字段如下所示。

- ☐ 新闻编号：用于标识新闻，为自动增长的主键。
- ☐ 新闻标题：用于表示新闻的标题。
- ☐ 发布时间：用于表示新闻发布的事件。
- ☐ 新闻作者：用于表示新闻的作者。
- ☐ 新闻内容：用于表示新闻的内容。
- ☐ 发布天气：用于表示新闻发布的天气。
- ☐ 新闻等级：用于表示新闻的等级。
- ☐ 阅读次数：用于表示新闻的阅读次数。
- ☐ 新闻分类：用于表示新闻的分类，为整型字段。

对于新闻分类表而言，可以使用少数字段进行新闻分类的描述，新闻分类表的字段如下所示。

- ☐ 分类编号：用于标识新闻的分类，为自动增长的主键。
- ☐ 分类名称：用于显示新闻分类的名称。

在管理员进行新闻操作之前，首先需要验证身份，如果管理员是合法用户则通过验证，否则就不允许进行后续操作，管理员的身份验证和登录模块基本相同，但是其功能要比登录模块少很多，`admin` 表结构中的字段如下所示。

- ☐ 管理员编号：用于标识管理员信息，为自动增长的主键。
- ☐ 管理员用户名：用于标识管理员用户名。
- ☐ 管理员密码：用于标识管理员的密码，通常情况下和管理员用户名一起进行身份验证。

对于新闻表而言，其功能并不是十分的复杂，而新闻模块的难度不在于数据库的设计上，而在于前台显示和静态生成，静态生成主要是要利用模板解析技术进行静态生成，模板解析技术可以使用编程的方法进行编写也可以使用数据库进行模板技术的支持，这里使用 `htm` 文本作为数据库进行技术解析处理，将在后面的章节进行介绍。

25.3.2 数据表的创建

创建表可以通过 SQL Server Management Studio 视图进行创建也可以通过 SQL Server Management Studio 查询使用 SQL 语句进行创建。新闻模块同样需要创建多个表进行模块功能的实现，首先最重要的是 `news` 表，`news` 表的字段如下所示。

- ☐ `id`：用于标识新闻，为自动增长的主键。
- ☐ `title`：用于表示新闻的标题。
- ☐ `time`：用于表示新闻发布的事件。
- ☐ `author`：用于表示新闻的作者。
- ☐ `content`：用于表示新闻的内容。
- ☐ `weather`：用于表示新闻发布的天气。
- ☐ `level`：用于表示新闻的等级。

- ❑ hits: 用于表示新闻的阅读次数。
- ❑ classname: 用于表示新闻的分类，为整型字段。

确定好 news 表的各个字段后，就能够创建一个 news 表，news 表结构如图 25-4 所示。



	列名	数据类型	允许空
	id	int	<input type="checkbox"/>
	title	nvarchar(50)	<input checked="" type="checkbox"/>
	time	datetime	<input checked="" type="checkbox"/>
	author	nvarchar(50)	<input checked="" type="checkbox"/>
	[content]	nchar(10)	<input checked="" type="checkbox"/>
	weather	nvarchar(50)	<input checked="" type="checkbox"/>
	[level]	int	<input checked="" type="checkbox"/>
	hits	int	<input checked="" type="checkbox"/>
	classname	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

图 25-4 news 表结构

图中的字段描述了相应的字段在实际应用中的意义，创建表的 SQL 语句如下所示。

```
USE [news]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[news](                                //创建 news 表
    [id] [int] IDENTITY(1,1) NOT NULL,
    [title] [nvarchar](50) COLLATE Chinese_PRC_CI_AS NULL,
    [time] [datetime] NULL,
    [author] [nvarchar](50) COLLATE Chinese_PRC_CI_AS NULL,
    [content] [nvarchar](3000) COLLATE Chinese_PRC_CI_AS NULL,
    [weather] [nvarchar](50) COLLATE Chinese_PRC_CI_AS NULL,
    [level] [int] NULL,
    [hits] [int] NULL,
    [classname] [int] NULL,
    CONSTRAINT [PK_news] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

news 表中的 classname 字段为整型字段，这也就是说 classname 字段为另一个表的外键，另一个表 newsclass 用于描述新闻的分类的信息，newsclass 字段如下所示。

- ❑ id: 用于标识新闻的分类，为自动增长的主键。
- ❑ classname: 用于显示新闻分类的名称。

上述字段描述了 newsclass 表中需要使用的字段，可以使用 SQL 语句进行表和字段的创建，创建 newsclass 表的 SQL 语句如下所示。

```
USE [news]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[newsclass](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [classname] [nvarchar](50) COLLATE Chinese_PRC_CI_AS NULL,
    CONSTRAINT [PK_newsclass] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

//创建 newsclass 表

上述代码创建了 newsclass 表, 创建完成后, 还需要创建 admin 表, 通过上述字段描述可以了解 admin 表只需要保存管理员的用户名和密码即可, 则其字段可以描述为如下所示。

- ❑ id: 用于标识管理员信息, 为自动增长的主键。
- ❑ admin: 用于标识管理员用户名。
- ❑ password: 用于标识管理员的密码, 通常情况下和管理员用户名一起进行身份验证。

上述字段描述了 admin 表中需要使用的字段, 可以使用 SQL 语句进行表和字段的创建, 创建 newsclass 表的 SQL 语句如下所示。

```
USE [news]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[admin](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [admin] [nvarchar](50) COLLATE Chinese_PRC_CI_AS NULL,
    [password] [nvarchar](50) COLLATE Chinese_PRC_CI_AS NULL,
    CONSTRAINT [PK_admin] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

//创建 admin 表

上述代码创建了 admin 表, 用于进行管理员的身份验证, 创建完成后的 admin 表和 newsclass 表如图 25-5 和图 25-6 所示。



	列名	数据类型	允许空
	id	int	<input type="checkbox"/>
	admin	nvarchar(50)	<input checked="" type="checkbox"/>
	password	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

图 25-5 admin 表结构

	列名	数据类型	允许空
	id	int	<input type="checkbox"/>
	classname	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

图 25-6 newsclass 表结构

创建完成 admin 表之后就需要插入一个管理员, 在 SQL 中可以新建查询并执行 SQL 语句进行管理员表中数据的插入, 示例代码如下所示。

```
INSERT INTO admin (admin,password) VALUES ('guojing','0123456')
```

执行上述代码就能够进行 admin 表的数据插入, 插入一个新管理员之后, 就能够在后面的登录操作中使用该表的管理员信息。

25.4 界面设计

新闻模块包括众多的页面，这些页面包括登录页面、后台框架集、新闻发布页面、新闻删除页面等页面，这些页面都需要进行界面设计。在后台的开发过程中，虽然对后台的界面设计并没有苛刻的要求，但同样需要良好的用户体验，本章使用 Microsoft Expression Web 2 进行页面设计。

25.4.1 登录界面

登录界面用于进行管理员的身份验证，管理员可以在后台进行登录执行相应的新闻操作，如果管理员为合法用户，则允许进行新闻操作，否则不允许进行新闻操作，登录界面 HTML 代码见光盘中源代码\第 25 章\25-1\25-1\admin\login.aspx。

上述代码使用了 TextBox 控件以及验证控件和按钮控件，这些控件用于验证用户输入的是否正确并且判断用户是否为合法管理员，管理员可以通过该页面进行登录操作。如果登录成功，系统会跳转到后台管理框架集中，如果登录不成功，则会提示相应的错误信息。

25.4.2 后台框架集

后台操作中，为了提高页面的友好度，可以使用框架集进行后台开发，框架集是多个网页组成的一个页面，使用框架集能够在不刷新的情况下进行页面跳转，使用 Microsoft Expression Web 2 可以制作框架集。在 Microsoft Expression Web 2 中，单击【文件】选项，在下拉菜单中单击【新建】选项，单击【网站】选项，在弹出窗口中选择框架集，如图 25-7 所示。

框架集可以将多个页面放在同一个页面，在 Microsoft Expression Web 2 中可以创建框架集并为框架集中的页面进行指定或新建，如图 25-8 所示。

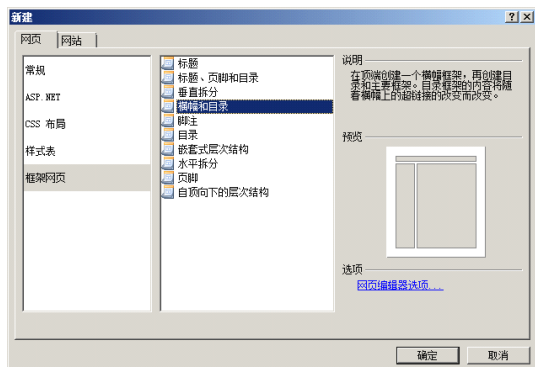


图 25-7 创建框架集

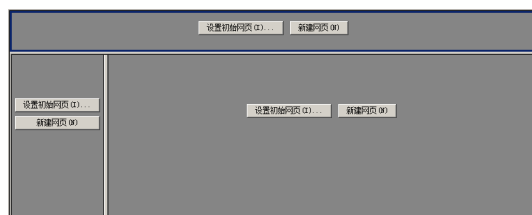


图 25-8 设置初始网页或新建网页

开发人员可以在框架集中创建网页或选择设置初始网页，这里创建三个网页，头部的网页用于显示后台管理的基本信息，包括这是什么后台管理系统；左侧的边栏用于显示操作，这里使用 TreeView 控件进行显示；中间为主操作区，该操作区用于后台中主要的页面操作。设置完成后示例代码见光盘中源代码\第 25 章\25-1\25-1\admin\default.aspx。

页面中的代码使用了一个框架集。在该框架集中包括三个页面，这三个页面分别为 top.aspx、left.aspx 和 center.aspx，其中 top.aspx 用于显示相应的信息，主要是用来作为导航或者后台提示，left.aspx 用于显示导航，使用 TreeView 控件能够为该页面制作相应的导航，而 center.aspx 用于呈现相应的操作页面，

在这里可以被成为主工作区。

开发人员能够在不同的页面进行布局，控件拖动和事件等操作，当用户访问框架集时，各个页面之间互不影响，可以在框架集之间进行页面跳转，其中 `left.aspx` 代码见光盘中源代码\第 25 章\25-1\25-1\admin\left.aspx。

`Left.aspx` 页面代码使用了 `TreeView` 控件在 `left.aspx` 页面中添加了导航信息，但是上述代码并没有配置 `TreeView` 控件中相应字段的 URL 属性，开发人员可以通过 `TreeView` 控件的属性进行配置。这里只提供 `left.aspx` 代码，对于其他页面的代码可以自行布局显示。如图 25-9 所示。

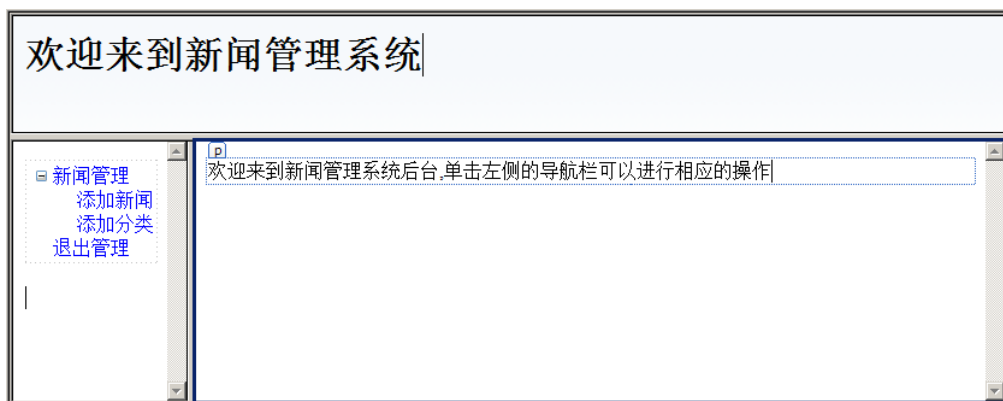


图 25-9 框架集布局

25.4.3 新闻发布页面

新闻发布页面是新闻系统中最为重要的页面，新闻发布页面主要使用 ADO.NET 进行新闻的发布和提交等操作，管理员能够在该页面进行新闻填写、新闻分类选择，然后管理员就能够进行新闻数据操作，新闻发布页面示例代码见光盘中源代码\第 25 章\25-1\25-1\admin\news_add.aspx。

`News_add.aspx` 页面代码使用了基本的文本框控件用于文本的输入。在一些用户数据输入时，为了保证用户输入的是完整的、符合规范的以及安全的数据，就需要使用下拉菜单控件进行数据呈现，示例代码见光盘中源代码\第 25 章\25-1\25-1\admin\news_add.aspx 中下拉菜单所示。

`News-add.aspx` 页面代码声明了多个文本框控件和下拉菜单控件用于文本的输入和呈现，管理员还需要通过数据源控件进行数据绑定并通过按钮控件进行数据提交。按钮控件用于数据提交，而数据绑定控件主要用于绑定下拉菜单方便管理员选择，示例代码见光盘中源代码\第 25 章\25-1\25-1\admin\news_add.aspx 中数据源配置代码所示。

新闻发布页面使用了数据源控件进行新闻分类的绑定，这也就是说明了在新闻添加之前，必须要选择新闻分类，否则新闻分类没有被填写，系统就会提示错误。在新闻页面设计中，使用 `TextBox` 控件和验证控件对管理员的操作进行验证和控制，如果管理员没有填写相应的信息，则系统会提示管理员填写，当管理员填写完成后，就可以单击控件进行提交。

25.4.4 新闻修改页面

新闻修改页面可以使用控件进行编写，新闻修改页面的数据获取同样需要从传递的参数中进行选择和判断，在 ASP.NET 3.5 中提供了一些数据绑定控件能够进行相应的数据的查询和更新，这里使用 `DetailsView` 控件，示例代码见光盘中源代码\第 25 章\25-1\25-1\admin\modi.aspx。

上述代码使用 DetailsView 控件进行数据绑定并能够使用 DetailsView 控件自带的更新功能进行数据更新。由于新闻更改页面需要通过获取的参数进行查询和更新，在配置 DetailsView 控件使用的数据源时，SELECT 查询语句必须配置参数，如图 25-10 所示。

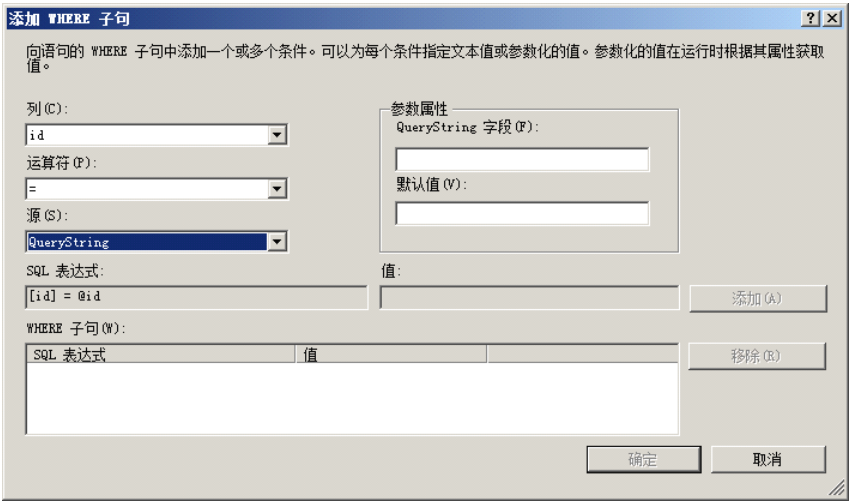


图 25-10 配置 WHERE 子句

配置 WHERE 子句就能够让数据源通过传递的参数进行相应的数据更新而不会涉及到其他的新闻数据，数据源源代码见光盘中源代码第 25 章\25-1\25-1\admin\modi.aspx 中数据源配置代码。

在配置数据源时，同样需要配置能够自动生成“插入、更新、删除”等操作，这样数据绑定控件才能够支持数据的插入、更新和删除。在新闻修改页面，只需要进行新闻的更新即可，在配置数据绑定控件时，无需选择“插入、删除”等操作。

25.4.5 新闻管理页面

新闻管理页面可以使用 GridView 控件进行编程，这样不仅能够简化开发人员的开发操作，还能够提高开发效率，因为 GridView 控件能够支持数据的更新和删除，使用 GridView 控件能够直接执行数据的更新和删除操作。由于新闻修改页面是一个单独的页面，而且在单独的页面中进行新闻修改能够提高用户体验，所以在新闻管理页面中就不再使用修改功能，GridView 控件示例代码见光盘中源代码第 25 章\25-1\25-1\admin\manage.aspx。

使用 GridView 控件可以进行相应字段的筛选，在管理页面中，并不需要每个字段都显示，例如 content 新闻内容字段。如果在管理页面同样要呈现 content 字段的话，那么当 content 字段的数据很多，例如是一篇很长的文章，那么页面就会被压缩的很难看，甚至变形。所以在新闻管理页面可以选择显示相应的字段而不显示一些不常用的字段，GridView 控件配置后如图 25-11 所示。

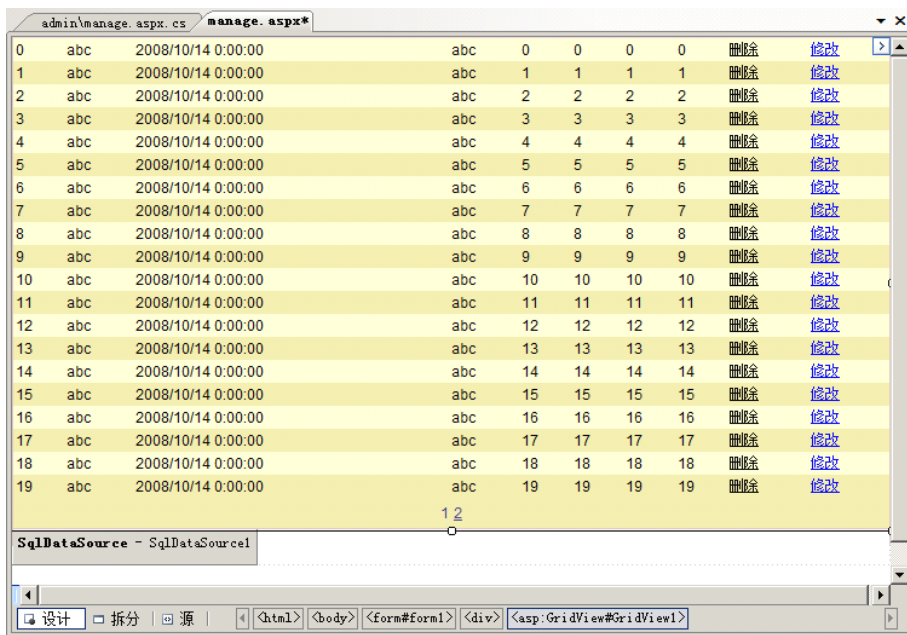


图 25-11 GridView 控件

GridView 控件自己能够支持删除操作。在 GridView 控件中，需要添加 HypeLink 列进行页面跳转。当管理人员单击【修改】连接时，就能够跳转到新闻修改页面进行新闻修改，而无需关心新闻修改页面的开发和维护。

25.4.6 新闻分类管理页面

新闻分类管理页面用于管理新闻分类，管理员可以在新闻分类管理页面进行新闻分类的添加和删除，在新闻分类的管理和添加页面中，同样可以使用 ASP.NET 3.5 提供的 ListView 控件进行分页、添加、修改、删除等操作，示例代码见光盘源代码第 25 章\25-1\25-1\admin\classmanage.aspx。

ListView 控件能够提供数据的插入、更新和删除等功能，对于简单的数据操作可以使用 ListView 控件进行功能实现，而如果需要复杂的数据操作和页面布局，使用 ListView 控件就不能很好的完成。ListView 控件通常情况下可以使用到数据较少，数据字段较短的情况下，如果对页面布局要求不是很高，也可以使用 ListView 控件。

25.5 代码实现

在新闻模块开发中包含了很多的页面，这些页面有些可以使用控件进行实现，而有些需要使用编程的方法进行代码实现，例如登录时的身份验证，新闻模块中的新闻发布甚至是新闻显示和生成静态等功能。这些功能没有现有的控件提供功能实现，而需要开发人员进行代码实现。

25.5.1 导航菜单配置

在后台管理的框架集中，可以配置导航菜单进行页面跳转，管理人员可以通过导航菜单进行页面的管理和跳转操作。当管理员单击导航菜单上的新闻管理时，就应该在主工作区中间显示新闻管理页面而

不会刷新其他页面，同样当管理员点击其他操作时，会跳转到不同的页面。导航菜单示例代码见光盘中原代码\第 25 章\25-1\25-1\admin\left.aspx。

Left.aspx 页面代码使用了 TreeView 控件中节点的 NavigateUrl 属性配置了单击该节点时应该跳转的页面，并使用了节点的 Target 属性进行了连接应该跳转的框架集。

注意：在超链接中包含 target 属性，而 TreeView 控件同样包含 Target 属性。在框架集中，需要指定相应的连接所需要跳转的框架，否则跳转的框架会在自身页面中执行。

25.5.2 身份验证页面

在管理员操作之前，首先需要进行身份验证，如果管理员是合法的用户，那么系统就能够使管理员进行添加和删除等操作。如果系统验证操作人员不是合法用户，那么就不应该为用户赋予权限，并阻止用户的登录和管理操作。打开登录页面，双击按钮编写相应的登录事件，示例代码如下所示。

```
protected void Button1_Click(object sender, EventArgs e)
{
    try
    {
        SqlConnection con = new SqlConnection("Data Source=(local);
                                                Initial Catalog=news;Integrated Security=True"); //创建连接
        con.Open();
        string strSql = "select * from admin where admin='" + TextBox1.Text + "' and password='"
                        + TextBox2.Text + "'"; //创建 SQL
        SqlDataAdapter da = new SqlDataAdapter(strsql, con); //创建适配器
        DataSet ds = new DataSet(); //创建数据集
        int count = da.Fill(ds, "table"); //填充数据集
        if (count > 0) //如果存在用户
        {
            Session["admin"] = TextBox1.Text; //配置一个 Session
            Response.Redirect("default.aspx"); //页面跳转
        }
        else
        {
            Label1.Text = "无法登录,请检查用户名和密码"; //提示无法登录
        }
    }
    catch
    {
        Label1.Text = "无法进行数据连接"; //抛出异常
    }
}
```

上述代码使用了 ADO.NET 中的数据对象进行数据操作，其操作和登录控件一样，在数据库中查询相应的信息，如果查询的结果大于 0 则说明该查询在数据库中是有效的，也就是说存在这样一个管理员能够进行登录，而如果查询的结果小于 0 则说明不存在相应的管理员。

在管理页面的各个页面都需要进行用户身份判断，当管理员登录成功后，系统会配置一个 Session 对象给用户，如果管理员操作超时或者操作者是一个非法用户，那么就没有 Session 对象。在各个页面判断 Session 对象是否存在，就能够判断是否是合法的管理员，示例代码如下所示。

```
if (Session["admin"] == null) //如果不为管理员
```

```

    {
        Response.Redirect("login.aspx");           //登录跳转
    }

```

如果相应的 Session 对象为空，就说明正在操作的用户不具备管理权限，则应该跳转到 login.aspx 页面进行重新进行登录操作。

25.5.3 新闻发布页面

在新闻发布页面，管理员只需要在相应的字段进行新闻内容的填写，包括新闻的标题、内容、作者等就能够进行新闻的发布。对于非静态生成的新闻发布而言，只需要进行数据的插入即可，示例代码如下所示。

```

protected void Page_Load(object sender, EventArgs e)
{
    if (Session["admin"] == null)                //如果不为管理员
    {
        Response.Redirect("login.aspx");         //登录跳转
    }
    TextBox2.Text = DateTime.Now.ToString();      //初始化字段
}
protected void Button1_Click(object sender, EventArgs e)
{
    try
    {
        SqlConnection con = new SqlConnection("Data Source=(local);
                                                Initial Catalog=news;Integrated Security=True"); //创建连接
        con.Open();                               //打开连接
        string strsql = "insert into news (title,time,author,content,weather,level,hits,classname) values
                        (" + TextBox1.Text + "," + TextBox2.Text + "," + TextBox3.Text + "," +
                        TextBox5.Text + "," + TextBox4.Text + "," + DropDownList1.Text + ",0," +
                        DropDownList2.Text + ")";    //SQL 语句
        SqlCommand cmd = new SqlCommand(strsql, con); //创建 Command
        cmd.ExecuteNonQuery();                     //执行 SQL 语句
        Response.Redirect("manage.aspx");          //页面跳转
    }
    catch(Exception ee)
    {
        Response.Write(ee.ToString());             //抛出异常
    }
}

```

新闻发布页面只需要执行相应的 SQL 语句进行数据插入即可，如果新闻发布只需要进行动态读取，那么只需要进行静态插入新闻即可，而不需要进行模板解析操作；如果需要生成静态页面，那么就需要进行模板编写再生成纯静态页面。

25.5.4 静态生成功能

静态生成听上去非常的复杂，但是其实静态生成非常的简单。当管理员发布一条新闻，就会在数据库中插入数据，数据插入后就应该解析模板进行静态生成。静态生成的是一个文件，这个文件可以是一

个.html 文件或者.shtml 文件。在生成文件之前，可以运行以下代码，示例代码如下所示。

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)                                     //判断加载
    {
        string str = "<b>*title*</b>";                    //模板代码
        string database = "新闻标题";                     //数据库字段
        string output = str.Replace("*title*", database); //替换操作
        Response.Write(output);                           //输出字符串
    }
}
```

上述代码简单的定义了一个模板代码，其中 str 变量被定义为标签，database 变量被定义为新闻标题，当需要生成静态时，可以将模板中代码的关键字进行替换，例如这里的*title*替换成为 database 变量中的标签，替换完成后输出替换后的字符串生成即可。在模板代码中，模板代码如下所示。

```
string str = "<b>*title*</b>";
```

其中*title*是定义的标签，用于替换关键字，*title*可以替换新闻标题数据进行呈现。在了解了静态生成的原理之后，就能够使用模板解析进行静态生成，示例代码如下所示。

```
protected void Button1_Click(object sender, EventArgs e)
{
    try
    {
        SqlConnection con = new SqlConnection("Data Source=(local);
                                                Initial Catalog=news;Integrated Security=True"); //创建连接
        con.Open();                                     //打开连接
        string strsql = "insert into news (title,time,author,content,weather,level,hits,classname) values
                        (" + TextBox1.Text + "," + TextBox2.Text + "," + TextBox3.Text + "," +
                        TextBox5.Text + "," + TextBox4.Text + "," + DropDownList1.Text + ",0," +
                        DropDownList2.Text + ")";         //SQL 语句
        SqlCommand cmd = new SqlCommand(strsql, con);   //创建执行对象
        cmd.ExecuteNonQuery();                          //执行 SQL
        StreamReader aw = File.OpenText(Server.MapPath("template.htm")); //打开模板
        string template = aw.ReadToEnd();               //读取模板
        aw.Close();                                     //关闭对象
        template = template.Replace("[新闻标题]", TextBox1.Text); //替换标签
        template = template.Replace("[发布时间]", TextBox2.Text); //替换标签
        template = template.Replace("[新闻作者]", TextBox3.Text); //替换标签
        template = template.Replace("[新闻天气]", TextBox4.Text); //替换标签
        template = template.Replace("[新闻内容]", TextBox5.Text); //替换标签
        StreamWriter sw = File.CreateText(Server.MapPath("../html/" + DateTime.Now.Year.ToString()
            + DateTime.Now.Month.ToString() + DateTime.Now.Day.ToString()
            + DateTime.Now.Hour.ToString() + DateTime.Now.Second.ToString()
            + ".htm")); //生成文件
        sw.Write(template);                             //编写内容
        sw.Close();                                     //关闭对象
        Response.Redirect("manage.aspx");               //页面跳转
    }
    catch (Exception ee)
    {
        Response.Write(ee.ToString());                 //抛出异常
    }
}
```

```
}
}
```

上述代码首先使用 ADO.NET 进行数据插入操作，在数据操作成功后，就进行静态生成。静态生成的模板为 template.htm，通过编写 template.htm 文件能够为静态文件编辑相应的模板。在模板读取之后，系统会将一些关键字进行替换，例如将 “[新闻标题]” 这个字符串替换为数据库中的新闻标题，当 template.htm 模板中包含 “[新闻标题]” 字符串时，就能够通过程序将该字符串替换成相应的数据库中的数据。

在使用 File 类之前，需要使用 System.IO 命名空间，以提供对文件的读取操作。当读取模板并替换了关键字字符串之后，就能够通过 File 类的 CreateText 进行文件的存储，示例代码如下所示。

```
StreamWriter sw = File.CreateText(Server.MapPath("../html/" + DateTime.Now.ToString() + ".htm"));
sw.Write(template);                                     //编写内容
sw.Close();                                             //关闭对象
```

上述代码生成了一个以时间为文件名的 htm 静态文件，当文件生成后，就会保存在 html 文件夹中。在编写静态标签时，需要注意的是就是静态标签应该比较复杂，例如不能够将 title 作为标签进行替换，因为可能文章中的很多字段都包含 title，而 html 页面本身就包含 title 字符串，如果将 title 进行替换，很有可能会将不应该替换的字符串替换成数据库文件，这样很有可能会造成模板输出错误。

在制作模板时，尽量使用一些符号进行标签规则，例如标题可以编写成为 { \$title\$ } 或者 [*title*] 等，就不会与现有的字符串中的字符进行冲突，替换了不该替换的字符串。

25.5.5 新闻显示页面

新闻显示页面的作用在于显示新闻，当用户单击一条新闻时就会跳转到新闻显示页面。例如在新浪网站上点击一个新闻，打开的页面就是新闻显示页面。新闻显示页面可以使用 CSS 和 HTML 布局，新闻显示页面的设计能够提高用户体验、吸引用户，新闻显示页面代码见光盘中源代码第 25 章\25-1\25-1\articles.aspx。

上述代码编写了一个 articles.aspx 文件，当用户访问新闻时，会打开该文件并通过参数获取新闻的值，然后将值填充到 Label 等控件中。如图 25-12 所示。



图 25-12 新闻显示页面

在参数获取和填充的过程可以在 Page_Load 方法中编写相应代码，示例代码如下所示。

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Request.QueryString["id"] != "")
    {
        SqlConnection con = new SqlConnection("Data Source=(local);
                                                Initial Catalog=news;Integrated Security=True");
        con.Open();                                     //创建连接
                                                        //打开连接
```



```

string strSql = "select * from news where id = " + Request.QueryString["id"] + ""; //查询数据
SqlDataAdapter da = new SqlDataAdapter(strSql, con);
DataSet ds = new DataSet(); //填充数据
int count = da.Fill(ds, "table");
if (count > 0)
{
    Label1.Text = ds.Tables["table"].Rows[0]["title"].ToString(); //填充控件
    Label2.Text = ds.Tables["table"].Rows[0]["content"].ToString(); //填充控件
    Label3.Text = ds.Tables["table"].Rows[0]["author"].ToString(); //填充控件
    Label4.Text = ds.Tables["table"].Rows[0]["time"].ToString(); //填充控件
}
else
{
    Response.Redirect("default.aspx"); //页面跳转
}
else
{
    Response.Redirect("default.aspx");
}
}

```

新闻显示页面的 HTML 代码使用了 Label 控件进行页面文字的呈现，当用户打开页面时，会通过传递过来的 id 参数进行数据查询。如果数据查询成功，就会填充到相应的控件中，如上述代码所示，填充后就会显示相应的文本给用户，用户就能够看到新闻。

25.5.6 静态模板编写

如果新闻模块不需要静态生成，那么就可以通过 id 参数进行 articles.aspx 页面的访问，例如可以通过 articles.aspx?id=1 这样规则的地址进行访问。但是动态的访问不仅数据访问量大，还不能很好的被搜索引擎搜录，如果选择新闻模块要生成静态，就需要编写相应的模板。由于新闻模块在添加新闻时就需要生成静态文件，其静态文件的模板读取的是 template.htm，所以编写 template.htm 的 HTML 代码就能够编写模板文件，template.htm 示例代码见光盘中源代码第 25 章\25-1\25-1\admin\template.htm。

上述代码编写了模板文件，其中[新闻标题]、[新闻作者]等都是模板标签。当生成静态文件时，系统会根据标签替换数据库中的数据，例如把[新闻标题]替换成“新闻标题”，然后生成到 html 文件夹中形成一个静态文件。

当用户单击新闻列表的时候，就可以跳转到相应的静态文件进行静态访问而无需动态读取文件，这样就能够快速的提高读取效率并降低服务器压力。

25.6 实例演示

使用新闻模块能够快速的发布新闻和显示新闻，如果新闻模块选择生成静态，那么就需要对模板进行编写和解析，管理员可以在后台快速的进行新闻发布，前台就能够显示相应的新闻。在发布新闻之前，首先需要进行登录，如图 25-13 所示。



图 25-13 登录验证

登录完成后, 就能够跳转到后台框架集, 后台框架集为管理员提供了遍历的操作, 如图 25-14 和图 25-15 所示, 其中侧边是导航栏, 右边大块的区域是工作区。



图 25-14 主框架集



图 25-15 添加新闻

框架集能够让管理员方便的进行导航和功能操作, 管理员可以在侧边栏中点击相应的菜单, 例如单击【添加新闻】和【添加分类】选项进行相应页面的跳转。如果管理员单击的是【添加新闻】选项, 则主工作区就能够呈现新闻添加页面; 如果管理员单击的是【添加分类】选项, 则主工作区就会呈现添加分类页面, 而对其他的页面丝毫不会造成影响。单击【添加分类】按钮, 首先要为系统添加一个分类, 如图 25-16 所示。

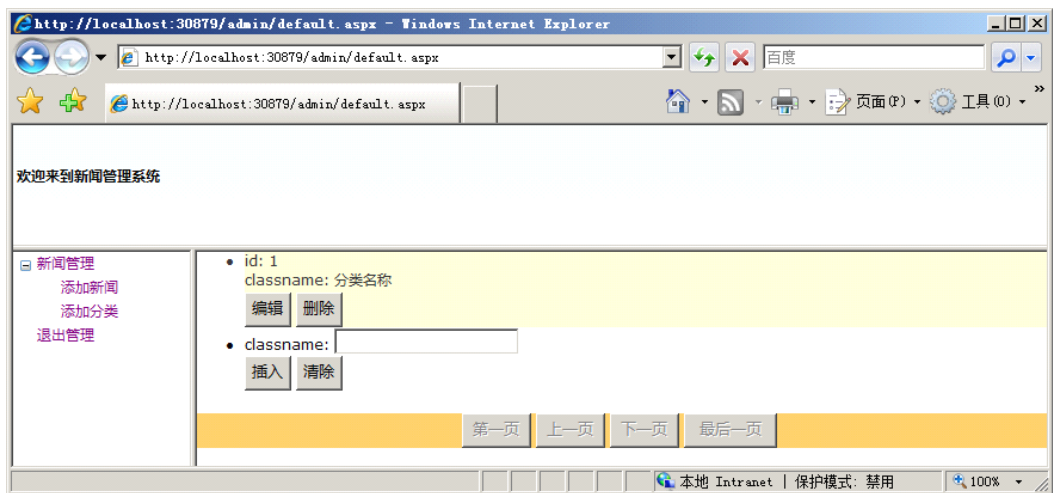


图 25-16 添加分类

分类添加完成后才能够添加新闻，如果在添加新闻之前已经有了相关分类，则没有必要再行添加，如果没有则必须添加分类。在添加分类后就能够添加新闻，这里使用了生成静态的方法生成了静态文件，虽然生成了静态文件，但是同样可以通过参数进行新闻数据的访问。添加新闻和管理新闻操作如图 25-17 和图 25-18 所示。

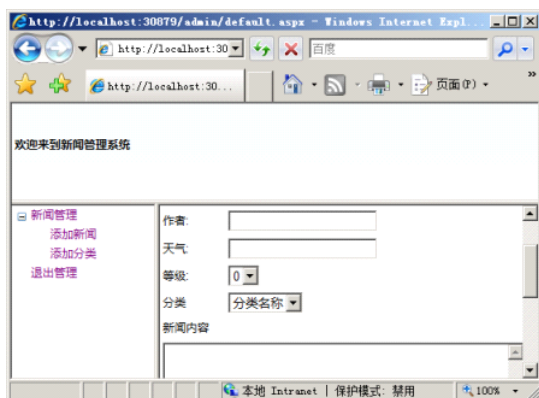


图 25-17 添加新闻

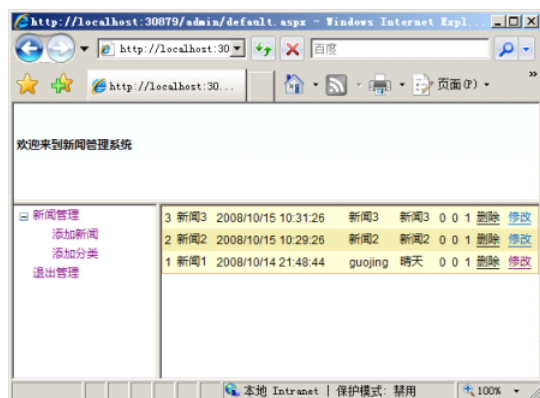


图 25-18 管理新闻

在添加完成后，就能够进行新闻的管理，新闻管理能够修改和删除新闻，修改和删除新闻都能够进行新闻模块中新闻数据的更改。当新闻被修改后，在前台所呈现的新闻样式也不同，当新闻被删除时，当用户访问该新闻则会跳转到首页。

新闻添加后，会在 html 文件夹中生成相应的静态文件，通过访问静态文件可以极大的减少服务器的压力，但是会增加服务器的 IO 读写负担，生成静态文件后同样可以通过参数的方式进行数据的访问。在 articles.aspx 文件中，通过传递的 id 参数进行数据查询和显示，虽然有些情况下需要进行数据库的操作，但是从另一方面来说减少了 IO 负担，如图 25-19 所示。

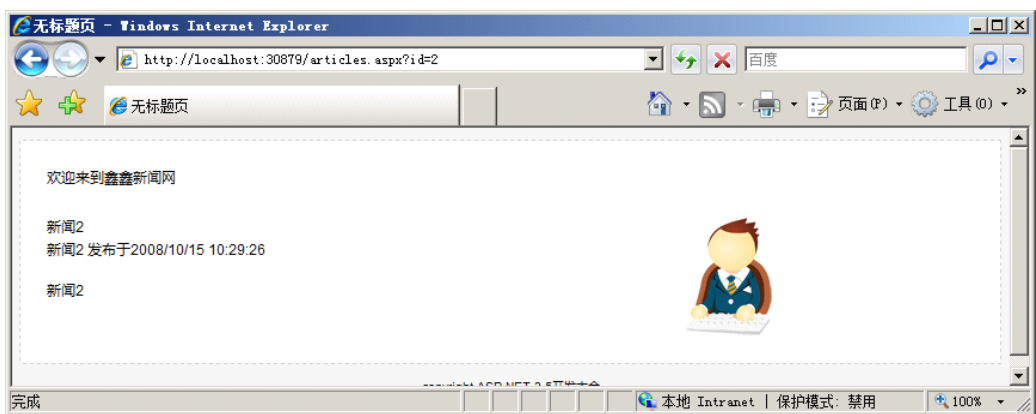


图 25-19 动态访问新闻

动态访问新闻要从数据库中读取数据，而静态访问不需要，新闻模块将静态数据文件存放在 html 文件夹中，访问 html 文件夹中的静态文件就能够访问相应的新闻，如图 25-20 所示。

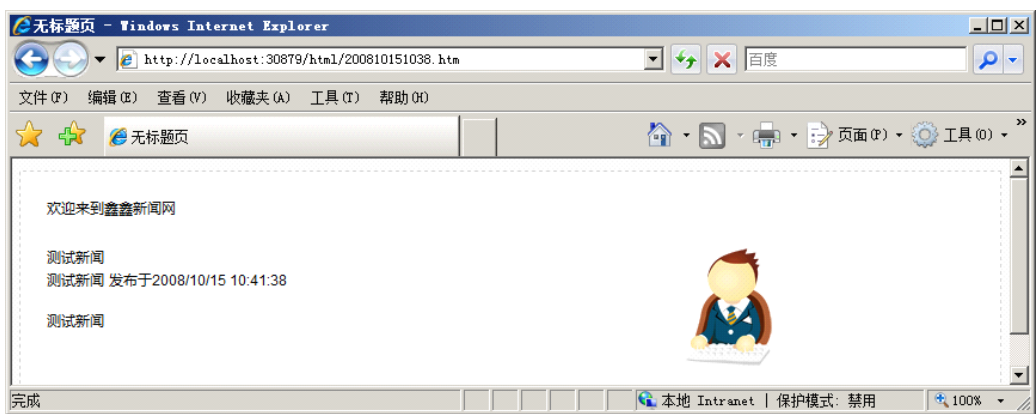


图 25-20 生成静态页面

正如图 25-20 中的地址栏所示，用户访问的是一个.htm 页面而不是.aspx 页面，这说明系统已经将数据进行替换并生成在 html 文件中，当用户访问相应的静态文件就能够访问相应的页面和数据而无需从数据中进行读取。

25.7 小结

本章通过讲解新闻模块的编写讲解了新闻模块的设计，动态读取和静态生成等功能。在网站中，新闻模块是必不可少的模块，对于大部分网站应用而言，新闻模块能够吸引用户，进行基础的用户信息交互。通过编写新闻模块巩固了：

- ☐ ASP.NET 的网页代码模型。
- ☐ Web 窗体基本控件。
- ☐ 数据库基础。
- ☐ ADO.NET 常用对象。
- ☐ Web 窗体数据控件。
- ☐ ASP.NET 内置对象。

□ 生成静态的概念

本章讲解了生成静态的概念，以及如何生成静态，还讲解了基本的模板解析技术，在大型的网站应用开发过程中，静态生成是非常重要的也是必不可少的。但是对于中小型的网站而言生成静态有可能会造成额外的负担，所以生成静态并不是万能的，开发人员需要针对不同的应用情况进行静态生成的选择。

本章讲解了静态生成的概念但是没有详细的介绍首页中如何调用新闻列表，调用新闻列表可以使用自定义控件进行调用，这一点可以参考广告模块中广告模块的自定义控件的显示，同样也能够使用静态化的方式进行新闻列表的生成，由于篇幅限制，这里不再进行过多的讲解，其基本原理和静态生成和模板解析相同。