

前端框架 Bootstrap

5.1 Bootstrap 简介

Bootstrap 是美国 Twitter 公司推出的基于 HTML、CSS、JavaScript 开发的前端开发框架。Bootstrap 用于快速开发 Web 应用程序和网站,是由 Twitter 的 Mark Otto 和 Jacob Thornton 开发的。Bootstrap 使用简洁灵活且完全开源,是目前广受欢迎的前端框架。

5.1.1 安装 Bootstrap

1. Bootstrap 的下载

可以通过 https://v3.bootcss.com/下载本节所使用的 Bootstrap 的 v3.4.1 版本, Bootstrap 中文文档页面如图 5-1 所示。注意: Visual Studio 2019 已经内嵌了 Bootstrap v3.4.1 版本,无须再下载安装。



图 5-1 Bootstrap v3.4.1 中文文档首页

单击"下载 Bootstrap"按钮,将会看到如图 5-2 所示的三种下载方式。

为了更好地理解和更方便地使用,本节中使用 Bootstrap 的预编译版本,即使用第一种 方式进行 Bootstrap 环境的安装。 126

下载		
Bootstrap (当前版本 v3.4.1) 技能等级的开发者和不同的使 吧。)提供以下几种方式帮你快速上档 用场景。继续阅读下面的内容,	手,每一种方式针对具有不同 看看哪种方式适合你的需求
用于生产环境的 Bootstrap 编译并压缩后的 CSS、JavaScript 和字体 文件。不包含文档和源码文件。 下载 Bootstrap	Bootstrap 源码 Less、JavaScript 和 李体文件的源码,并 且带有文档。需要 Less 编译器和一些设 置工作。 下载源码	Sass 这是 Bootstrap 从 Less 到 Sass 的源码移 植项目,用于快速地在 Rails、Compass 或 只针对 Sass 的项目中引入。 下载 Sass 项目

图 5-2 Bootstrap 的下载方式

2. 预编译版

Bootstrap 提供了两种形式的压缩包,在下载的压缩包内可以看到以下目录和文件,这些文件按照类别放到了不同的目录内,并且提供了压缩与未压缩两种版本。

下载压缩包之后,将其解压缩到任意目录即可看到如图 5-3 所示的目录结构。

bootstrap/	
├── css/	
	//预定义的CSS文件
	//CSS与LESS源码对应文件
	//压缩
bootstrap.min.css.map	
	//主题文件
bootstrap-theme.css.map	
bootstrap-theme.min.css	
└── bootstrap-theme.min.css.map	
├── js/	
	//js
L bootstrap.min.js	
L fonts/	
├── glyphicons-halflings-regular.eot	//字体 (字体图标)
├── glyphicons-halflings-regular.svg	
├── glyphicons-halflings-regular.ttf	
├── glyphicons-halflings-regular.woff	
ull glyphicons-halflings-regular.woff2	

图 5-3 解压缩后的目录结构

图 5-3 展示的是 Bootstrap 的基本文件结构:可以看到已编译好的 CSS 和 JS (bootstrap.*)文件,还有经过压缩的 CSS 和 JS(bootstrap.min.*)文件。同时还提供了 CSS 源码映射表(bootstrap.*.map),可以在某些浏览器的开发工具中使用。还有来自 Glyphicons 的图标字体,在附带的 Bootstrap 主题中使用到了这些图标。

预编译文件可以直接应用到任何 Web 项目中。

3. Bootstrap 源码

Bootstrap 源码包含预先编译的 CSS、JS 和图标字体文件,并且还有 LESS、JavaScript 和文档的源码。

less\js\和 fonts\目录分别包含 CSS、JS 和字体图标的源码。dist\目录包含上面所说的 预编译 Bootstrap 包内的所有文件。docs\包含所有文档的源码文件, examples\目录是 Bootstrap 官方提供的实例工程。除了这些,其他文件还包含 Bootstrap 安装包的定义文件、



许可证文件和编译脚本等。

5.1.2 Bootstrap 的主要特性

1. HTML 5 文档类型

Bootstrap 使用到的某些 HTML 元素和 CSS 属性需要将页面设置为 HTML 5 文档类型。在项目中的每个页面都要参照下面的格式进行设置。

```
<!doctype html>
<html lang="en">
...
</html>
```

2. 移动设备优先

移动设备优先是 Bootstrap v.3.4.1 的最显著的变化。在之前的 Bootstrap 版本中(直到 2.x),需要手动引用另一个 CSS,才能让整个项目支持移动设备。而 Bootstrap v.3.4.1 默认的 CSS 本身就对移动设备优先支持,然后才是桌面设备(PC 显示器)。

为了让 Bootstrap 开发的网站对移动设备友好,确保适当的绘制和触屏缩放,需要在 <head>之中添加视口(viewport)元数据标签,代码如下。

<meta name="viewport" content="width=device-width, initial-scale=1">

说明:

(1) width 属性控制设备的宽度。假设开发的网站将面向不同屏幕分辨率的设备浏览, 那么将它设置为 device-width 可以确保它能正确呈现在不同设备上。

(2) initial-scale=1确保网页加载时,以1:1的比例呈现,不会有任何的缩放。

(3) 在移动设备浏览器上,通过为视口设置 meta 属性为 user-scalable=no 可以禁用其 缩放(zooming)功能。禁用缩放功能后,用户只能滚动屏幕,使网站看上去更像原生 App。

```
<meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=no">
```

3. 排版与链接

Bootstrap 的排版链接样式均设置了基本的全局样式,分别描述如下。

(1)为 body 元素设置背景色为 # fff。

(2) 使用@font-family-base、@font-size-base 和@line-height-base 变量作为排版的基本参数。

(3)为所有链接设置了基本颜色@link-color,并且当链接处于悬停状态时才添加下 画线。

这些样式都能在 scaffolding.less 文件中找到对应的源码。

4. Normalize.css

为了增强跨浏览器渲染的一致性,Bootstrap使用了 Normalize.css,这是由 Nicolas Gallagher 和 Jonathan Neal 维护的一个 CSS 重置样式库,在 HTML 元素的默认样式中提供了更好的跨浏览器一致性。

5. 布局容器

Bootstrap 需要为页面内容和网格系统包裹一个 container 容器。Bootstrap 提供了两

个类。

128

(1) container 类用于固定宽度并支持响应式布局的容器。

```
<div class="container">
...
</div>
```

(2) container-fluid 类用于 100% 宽度,占据全部视口的容器。

```
<div class="container-fluid">
...
</div>
```

注意,由于 padding 等属性的原因,这两种容器类不能互相嵌套。

5.1.3 Bootstrap 在 Visual Studio 2019 中的应用

Bootstrap 可用任何 Web 开发环境进行编写,本节实例使用 Visual Studio 2019 进行编写。下面给出第一个例子的实现步骤。

【例 5-1】 第一个实例 Hello BootStrap。

(1) 启动 Visual Studio 2019,新建一"ASP.NET Web 应用程序(.NET Framework)", 单击"下一步"按钮,如图 5-4 所示。

创建新项目	搜索模板(Alt+S)(<u>S</u>)		<i>р</i> -	全部清除(<u>C</u>)
最近使用的项目模板(<u>R</u>)	C#	- Windows	- Web	-
『J ASP.NET Web 应用程序(.NET Framework) C#	E^C NUnit 测试项目			^
N ASP.NET Core 空 C#	▲」 包含 NUnit 测词 行。	的项目,可在 Windows、Linux i	和 MacOS 中的 .NET Cor	re 上运
	C# Linux	macOS Windows 桌面	测试 Web	
	C# Windo	立用程序(.NET Framework) ET 应用程序的项目模板。你可以f B程序,并可以在 ASP.NET 中添加 ws 云 Web	创建 ASP.NET Web Form 3许多其他功能。	is, MVC
	「這用于边缘的 W 包含测试单元的 测试(使用 Micro	eb 驱动程序测试(.NET Core) 页目,测试单元可自动在 Microsc ssoft WebDriver)。	oft Edge 浏览器内执行网	站的 UI
	C# Windo 适用于边缘的 W 包含测试单元的 测试(使用 Micro	ws Web 測试 eb 驱动程序测试(.NET Framewo 项目,测试单元可自动在 Microsc osoft WebDriver)。	rk) oft Edge 浏览器内执行网:	站的 UI
	C# Windo	ws Web 测试		
	F@ D光庄			
				下一步(N)

图 5-4 "创建新项目"窗口

(2) 在"配置新项目"对话框中输入项目名称并选择保存位置后单击"创建"按钮,如图 5-5 所示。

(3) 在"创建新的 ASP.NET Web 应用程序"对话框中选择项目类型——Web Forms 选项,然后单击"创建"按钮,完成项目创建,如图 5-6 所示。



]	×
配置新项目			
ASP.NET Web 应用程序(.NET Framework) C# Windows 云 Web			
项目名称(1)			
тувоотstrap			
位置(L)			
D:\数材改版\ -	2**		
解决方案(<u>S</u>)			
创建新解决方案			
解决方案名称(<u>M</u>)()			
myBootstrap			
✓ 将解决方案和项目放在同一目录中(D)			
框架(E)			
.NET Framework 4.7.2 *			
	上一步(<u>B</u>) 创建(<u>C</u>)		

图 5-5 "配置新项目"对话框

⊕ _4	空 用于创建 ASP.NET 应用程序的空项目模板。此模板中没有任何内容。	身份验证 不进行 身份验 证 更改
5	Web Forms 用于创建 ASP.NET Web Forms 应用程序的项目模板。ASP.NET Web Forms让你使用熟悉的拖拢操作、事件驱动模型构建 动态网站。使用设计图面和数百个控件和组件,可以快速生成允许数据访问的复杂且功能强大的 UI 驱动站点。	添加文件夹和核心引用
(0)-	MVC 用于创建 ASP.NET MVC 应用程序的项目模板。ASP.NET MVC 允许你使用"模型-视图-控制器"体系结构构建应用程序。 ASP.NET MVC 为创建使用最新标准的应用程序提供了许多功能,这些功能实现了测试驱动的快速开发。	 ✓ Web 窗体(E) MVC(M) Web API(<u>W</u>)
{e}	Web API 用于创建可以访问范围广泛的客户端(包括浏览器和移动设备)的 RESTful HTTP 服务的项目模板。	高级 ✔ 为 HTTPS 配置(<u>C</u>)
(0)_4	单页应用程序 用于使用 ASP.NET Web API 创建群客户端 JavaScript 驱动的 HTML5 应用程序的项目模板。单页应用程序提供丰富的用户 体验,其中包括使用 HTML5、CSS3 和 JavaScript 进行客户端交互。	 Docker 支持 (需要 Docker Desktop) 此外创建一个单元测试项目(<u>U</u> myBootstrap.Tests

图 5-6 "创建新的 ASP.NET Web 应用程序"对话框

(4) 在"解决方案资源管理器"中可以看到,VS2019 中已添加了 Bootstrap 核心文件,如 图 5-7 所示。

(5) 右击项目,在弹出的右键菜单中依次选择"添加"→"Web 窗体"命令,在弹出的"指

基于 C#的管理信息系统开发(第3版)

130



图 5-7 Bootstrap 核心文件

定项名称"对话框中输入项名称,如图 5-8 所示。

指定项名和	家	×
项名称:	firstBootsPage	确定 取消

图 5-8 指定项名称

(6) 在文件中增加两个 meta 标签,代码如下。

```
<!-- 用以声明当前页面用最新的 IE 版本模式来渲染-->
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<!-- 让 Bootstrap 开发的网站对移动设备友好,确保适当的绘制和触屏缩放-->
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

(7) 导入 Bootstrap 主样式,代码如下。

<!-- 导人 bootstrap 的主样式--> <link rel="stylesheet" href="Content/bootstrap.min.css" type="text/css" /> **注意**:如自定义文件夹与本节代码不同,请将路径修改为自己命名的文件夹。 在文件的
body>标签中添加代码如下。

<h1>Hello,Bootstrap!</h1>

即完成第一个例子,如图 5-9 所示。

N 3 25117	登录 A, − □ × Ligi tive Share R
Image: Control Reading Reading Reading Reading Reading Reading Lines (Second Reading Reading Reading Lines (Second Reading Reading Reading Lines (Second Reading Reading Lines (Second Readi	BX A - × In the lase If the same If the same Image: Second Seco
	 2 2015(2004) 年前(Cocogle Chromo)(B) Ctrl+Shift+W 4 2015(2004) 年前(Cocogle Chromo)(B) Ctrl+Shift+W 4 2016(2014) 年前(C)(D) 4 2016(2014) 14(10)
	 部議解決力室的習慣器((型)(N) 以項目中非時(J)
	□ 重約4(M) F2 □ 算動完整路径(U)
133% - 6.林政府総合機 (

图 5-9 第一个例子

右击 firstBootsPage,可直接在浏览器中查看页面运行效果,如图 5-10 所示。



后续例子均可在该项目下新建 HTML 文件运行。

5.2 Bootstrap 样式

本节将讲述 Bootstrap 底层结构的关键部分,包括网格系统、排版、表格、表单及按钮等。

5.2.1 网格系统

1. 简介

Bootstrap 提供了一套响应式、移动设备优先的流式网格系统,随着屏幕或视口尺寸的 增加,系统会自动分为最多 12 列。它包含易于使用的预定义类,还有强大的 mixin 用于生 成更具语义的布局。

2. 工作原理

132

网格(也称栅格)系统用于通过一系列的行(row)与列(column)的组合来创建页面布局,而网页的内容就可以放入这些创建好的布局中。以下介绍 Bootstrap 网格系统的工作原理。

(1) "行"必须包含在固定宽度的容器(container)或 container-fluid(100% 宽度)中,以 便为其赋予合适的对齐(alignment)和内边距(padding)。

(2)通过"行"在水平方向创建一组"列"。内容应当放置于"列"内,并且,只有"列"可以 作为"行"的直接子元素。

(3) 类似 row 和 col-xs-4 这种预定义的类,可以用来快速创建网格布局。Bootstrap 源 码中定义的 mixin 也可以用来创建语义化的布局。

(4) 通过为"列"设置内边距属性,从而创建列与列之间的间隔(gutter)。通过为"行"元素设置负值边距(margin)从而抵消掉为 container 元素设置的内边距,也就间接为"行"所包含的"列"抵消掉了内边距。

(5)负值的边距可以使得页面元素向外突出。在网格列中的内容排成一行。

(6) 网格系统中的列是通过指定 1~12 的值来表示其跨越的范围。例如,三个等宽的 列可以使用三个 col-**-4 来创建。

(7)如果一"行"中包含的"列"大于 12,多余的"列"所在的元素将被作为一个整体另起 一行排列。

(8) 网格类适用于屏幕宽度大于或等于分界点大小的设备,并且针对小屏幕设备覆盖 网格类。因此,在元素上应用任何 col-md-* 网格类适用于屏幕宽度大于或等于分界点大 小的设备,并且针对小屏幕设备覆盖网格类。

3. 媒体类型查询

在网格系统中,使用媒体类型查询来创建关键的分界点阈值,代码如下。

```
/* 超小屏幕(手机,小于 768px) */
/* 没有任何相关的代码,因为这在 Bootstrap 中是默认的(Bootstrap 是移动设备优先) */
/* 小屏幕(平板,大于或等于 768px) */
@media (min-width: @screen-sm-min) { ... }
/* 中等屏幕(桌面显示器,大于或等于 992px) */
@media (min-width: @screen-md-min) { ... }
/* 大屏幕(大桌面显示器,大于或等于 1200px) */
@media (min-width: @screen-lg-min) { ... }
```

我们偶尔也会在媒体类型查询代码中包含 max-width 从而将 CSS 的影响限制在更小范围的屏幕大小之内,如下。

```
@media (max-width: @screen-xs-max) { ··· }
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ··· }
@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ··· }
@media (min-width: @screen-lq-min) { ··· }
```

4. 网格参数

Bootstrap的网格系统在多种屏幕设备上工作的方式,如表 5-1 所示。

屏幕类型属性	网格系统行为	.container 最大宽度	类前缀	最大列宽
超小屏幕(手机)(<768px)	总是水平排列	None(自动)	col-xs-	自动
小屏幕(平板)(≥768px)		750px	col-sm-	$\sim 62 \mathrm{px}$
中等屏幕(桌面显示器) (≥992px)	开始是堆叠在一起的, 当大于这些阈值时将变	970px	col-md-	$\sim 81 \mathrm{px}$
大屏幕(大桌面显示器) (≥1200px)	为水平排列	1170px	col-lg-	$\sim 97 \mathrm{px}$

表 5-1 网格系统在各种屏幕上工作的方式

另外,对于各种类型的屏幕,其最大列数均为12,间隔(gutter)宽均为30px,可嵌套、有 偏移且按列排序。

注意:后续文字描述中均以超小屏幕指代手机屏幕,小屏幕指代平板屏幕,中等屏幕指 代桌面显示器,大屏幕指代大桌面显示器。

5. 基本的网格结构

基本的网格结构代码如下。

```
<div class="container">
    <div class="row">
        <div class="col- * - * "></div>
        <div class="col- * - * "></div>
        </div>
        <div class="row">...</div>
        <div class="row">...</div>
    </div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div>
```

即为每一个列元素指定类。例 5-2 给出一个基本网格系统的例子。

说明:由于 Bootstrap 代码量较大,本节例题在正文中均只给出关键代码(省略部分由 "…"开头的斜体文字说明),以读者能理解其结构及作用为主,完整代码请参考配套资料中 的例题。

【例 5-2】 网格系统从堆叠到水平排列。

使用单一的一组 col-md-* 网格类,就可以创建一个基本的网格系统,在超小屏幕和小 屏幕设备上是堆叠在一起的,在中等屏幕设备上就会变成水平排列的。所有列必须放在 行内。

```
<div class="container">
    <div class="row">
        <div class="row">
        <div class="col-md-1">内容 1</div>
        <div class="col-md-1">内容 2</div>
        ···省略 3~11列部分代码
        <div class="col-md-1">内容 12</div>
        </div class="col-md-1">内容 12</div>
        </div class="col-md-4">内容 12</div>
        <div class="col-md-4">内容 12</div>
        <div class="col-md-4">内容 12</div>
        </div class="col-md-4">内容 3</div>
        </div class="col-md-4">内容 3</div>
        </div class="col-md-4">内容 3</div>
        </div class="row">
```



基于 C#的管理信息系统开发(第3版)

134

```
 <div class="col-md-6">内容 1</div>
 <div class="col-md-6">内容 2</div>
 </div>
 <div class="row">
      <div class="col-md-12">内容 1</div>
 </div>
 </div>
```

保存代码并运行,在超小屏幕及小屏幕上显示效果如图 5-11 所示。

内容1
内容2
内容3
内容4
内容6
内容6
内容7
内容8
内容9
内容10
内容11
内容12
内容1
内容2

图 5-11 例 5-2 超小屏幕及小屏幕显示效果

在中等屏幕上显示效果如图 5-12 所示。

内容1	内容2	内容3	内容4	内容5 内容6 内容7 内容8 内容9 内容10						内容11	内容12
内容1 内容					容2			内	容3		
内容1						内	容2				
内容1											

图 5-12 例 5-2 中等屏幕显示效果

说明:

(1)为了呈现更好的效果方便观看,为网格系统(除例 5-8 嵌套列)的例子均设置了如下样式。