

SharpSite 低代码 CMS 系统

说明书

V2.0.1

目录

一、 产品简介	3
1. 产品简介	3
2. 企业痛点	3
3. 产品优点	3
4. 适用人群	3
二、 系统安装	4
三、 网站模板安装	5
四、 网站模板语言介绍	5
五、 ASP.NET 模板语言	6
1. 包含文件:	6
2. 条件判断:	6
3. 循环:	6
4. 普通变量输出:	6
5. 读取后台数据:	6
1) 方式一, 直接使用原生 SQL 读取:	6
2) 方式二, 使用 json 格式读取:	7
3) 方式三, 读取单条记录:	8
4) 页面数据输出	9
5) 提交数据到后台	10
6) 数据格式化函数	11
7) 包含文件的有效利用	11

一、产品简介

1. 产品简介

SharpSite 低代码 CMS 系统：是一款专注于 B/S 架构的内容管理软件，可用于做管理系统和企业建站。如企业官网、企业多语言网站、多门户网站、新闻资讯网站、招聘网站、企业产品宣传网站、视频网站、图片网站、小型 CRM 系统、文档管理、会员管理、订单管理、产品管理、资产管理等各种 WEB 管理系统。

2. 企业痛点

当前所有企业信息化痛点：

- (1) 企业网站风格、功能各不相同，尤其企业官网需要个性化网页设计和特色模块需求。
- (2) 找网站建设公司定制开发，周期长，费用高，后期模块变动维护困难，费用高。
- (3) 自助软件都是通用版功能，网页模板也限制太多，无法满足企业个性化需求。
- (4) 随着企业的快速发展，想自己新增模块、新增字段无法解决。
- (5) 不能自己部署到企业私有服务器，即使是定制开发的网站也需要专业的 IT 人员才能维护。

定制开发软件痛点：

- (1) 价格昂贵，开发周期较长，且需要很长的测试周期来修复各种 Bug。
- (2) 无法满足企业的快速发展，想自己新增模块、新增字段无法解决，必须二次开发，价格高，开发周期长。
- (3) 软件后期维护费用高。

3. 产品优点

- (1) 无需编程，普通人都可在 1 天内即可完成网站模块搭建。
- (2) 自定义软件 Logo 及界面。
- (3) 自定义表名、字段名、批量添加字段、表单布局操作简单。
- (4) 自由设置后台管理菜单，没有任何多余菜单。
- (5) 自定义模块，不限模块数量。
- (6) 自定义字段，不限字段数量。
- (7) 自定义表单：拖拽字段布局，采用响应式布局，手机 PC 可同时使用。
- (8) 一套系统支持多套网站皮肤自由切换；也可同时新建多个多语言网站，同时共存。
- (9) 支持任意前端 HTML 皮肤，100%支持定制皮肤，只需要写模板语言就可以打通后台数据。
- (10) 支持在线开发 Vue3 组件及预览，无需安装 Vue 开发环境，可任意扩展企业个性需求。
- (11) 支持网页前端二次开发，如实现在线产品报价、产品真伪查询等，企业使用现有技术和架构即可完成，无需学习新技术。
- (12) 软件可租用，也可自己下载，私有化部署，支持普通云主机和虚拟主机。
- (13) 提供 ASP.NET、PHP、JAVA 三种语言版本下载，可自由选择安装版本。
- (14) 数据库支持主流数据库：Sqlserver、Mysql、Oracle、PostgreSQL、国产数据库人大金仓、达梦等。
- (15) 服务器支持主流 windows、linux、CentOS、RadHat 等，所有支持 PHP 和 JAVA 的服务器都可以。

4. 适用人群

- (1) 学生

无需编程基础，通过 SharpSite 可独立完成创建表、字段、表单、权限设置，搭建各种管理系统，如学员管理、档案管理、积分管理、图书管理、新闻资讯管理、产品管理、资产管理等。

(2) 销售人员

根据老板或者客户提出的开发需求，直接在后台创建表、字段，搭建表单，所见及所得，有任何需求修改，都能立马调整，立即查看效果。

(3) 产品经理

后台产品 UE 设计，无需使用 Axure 等原型图设计工具，直接通过后台拖拉就能创建所需要的模块、字段、表单、产品列表等。无论是做网站还是管理系统都适用。

(4) IT 人员

无需学习 JAVA、PHP、ASP.NET 开发语言，只要会 SQL，就能独立使用 SharpSite 搭建专业级定制网站，后期页面变动、新增模块、新增字段等，直接后台就可操作，效率高效。

(5) 程序员

懂 SQL 和后台开发语言（JAVA、PHP、ASP.NET 任意一种），中小型网站，可直接使用 SharpSite 进行表、字段在线设计，可自定义表名和字段名，表单、列表页、搜索栏立马生成，对于复杂的网站需求，还可以做二次开发，效率提高数倍。

(6) UI 设计师

使用 AI 软件生成网页 html 代码后，通过 SharpSite 搭建模块和字段后，使用最基础的 SQL 查询语句，就能完成所有网站的开发。

(7) 网站前端

和 IT 人员一样，无需后台开发配合，只要会 SQL，就能独立使用 SharpSite 搭建专业级定制网站。还能使用 Vue3 创建专用组件，后期页面调整+字段变动，更能得心应手。

(8) 企业老板

如何使用最低的预算，最快的速度上线系统，后期修改维护又需要简单方便，使用 SharpSite 可以节约大部分成本，减少沟通时间，加速产品快速落地。

(9) 网站建设公司

经常需要为客户搭建各种网站，每次都要重新开发后台，时间紧、项目多、人手不够，并且客户对 JAVA、PHP、ASP.NET 要求都不一样，程序员水平参差不齐，代码效率千差万别，离职后，其他程序员接手维护成本高。使用 SharpCRM 开箱即用，无需重复造车轮，销售、产品经理、前端、程序员每个人都可独立完成前后台搭建，大大缩短每个项目的上线周期。

二、系统安装

1) 将相应的 ASP.NET、PHP、JAVA 版本文件用 FTP 工具上传至虚拟主机、或云主机空间。

ASP.NET 使用 4.8+版本，PHP 使用 PHP8.2+版本，JAVA 使用 java26+版本。

2) 打开数据库配置页面：<http://域名/admin/DBConfig.html>

▶ 系统初始化配置

数据库类型 *	SqlServer	
数据库IP *	localhost	
数据库端口号		
数据库名称 *	DB_SharpSite	
数据库用户名 *	sa	
数据库密码 *	SharpSite@SharpCRM.cn	
后台管理员账号 *	SharpSite	
后台管理员密码 *	SharpSite	密码长度必须在8到30位

✔ 确定

填写数据库相关信息，点击确定，自动安装数据库，安装完成后，自动跳入登录页面：</admin/SysLogin.html>

三、网站模板安装

- 1) 如果您的网站只做管理系统，到此已经结束，直接登录后台操作即可，以下内容可以不用看。
- 2) 如果您的网站需要前端皮肤，或者做多语言、多皮肤网站，请将网站模板使用英文文件夹命名，如：CN 或 EN，放入/Skin/目录下，多个网站模板都必须放入此文件夹下。
- 3) 网站模板请使用标准格式，如：



四、网站模板语言介绍

如果您用来建网站，那么只需要了解前端模板语言即可，无需后台编程。ASP.NET 版使用 **NVelocity** 模板引擎、PHP 版使用 **Smarpty** 模板引擎、JAVA 版使用 **Velocity** 模板引擎。以上三种语言的基础语法可百度搜索。

五、ASP.NET 模板语言

1. 包含文件:

```
#include("header.html")
```

2. 条件判断:

```
#if  
#elseif  
#end
```

3. 循环:

```
#foreach($item in $list)  
    <li>$item</li>  
#end
```

4. 普通变量输出:

\$Param: 输出变量, 如果变量为空则显示 null。

#{Param}: 输出变量, 如果变量为空则显示 null。此方法用于隔离变量和页面中的其他英文字符。

!\$Param: 输出变量, 如果变量为空则不显示。

!#{Param}: 输出变量, 如果变量为空则不显示。此方法用于隔离变量和页面中的其他英文字符。

5. 读取后台数据:

1) 方式一, 直接使用原生 SQL 读取:

```
<!--  
[SharpSite]  
{  
    "SetParam": "list1",          /* 将查询结果赋值给变量list1, 一个页面里面必须唯一, 变量名不能重复 */  
    "Sql": "Select * From C_PortalMenu" /* 原生SQL查询, 可使用RequestInt("page"), RequestStr("page")获取Url参数 */  
}  
[/SharpSite]  
-->
```

参数说明:

1、必须使用格式: <!--[SharpSite]{JSON 参数}..... [SharpSite] -->

2、最少 2 个参数, 即:

```
<!--  
[SharpSite]  
{  
    "SetParam": "listNews",  
    "Sql": "Select * From C_News",
```

```
}
[/SharpSite]
-->
```

参数名称	示例	参数说明
SetParam	listNews	将 sql 查询结果赋值给页面变量 listNews，页面中输出数据需要循环此变量
Sql	Select * From C_News	原生 sql，懂 SQL 的人可以使用此方法，方便快捷。 Sql 语句中可使用 RequestInt("page")，或者 url 整数类型的参数值。 Sql 语句中可使用 RequestStr("page")，或者 url 字符类型的参数值，值的二边自动加英文单引号。

2) 方式二，使用 json 格式读取：

```
<!--
[SharpSite]
{
  "SetParam": "list2",          /* 必填，将查询结果赋值给变量list2，一个页面里面必须唯一，且不能重复 */
  "Select": "*",               /* 选填，查询字段，星号代表查询全部字段 */
  "From": "C_PortalMenu",      /* 必填，查询表 */
  "Where": "",                 /* 选填，数据过滤条件 */
  "OrderBy": "",               /* 选填，排序字段 */
  "PageSize": 10,              /* 选填，每页显示记录数，默认等于1；当PageSize大于1时，需循环输出记录；当PageSize等于1时，无需循环，直接输出单条记录 */
  "NextPageUrlCount": 5,       /* 选填，默认等于5，分页时使用，显示前后翻页数量 */
  "ViewCount": "ClickCount",   /* 选填，详细页浏览字段，详细页浏览器次数统计 */
}
[/SharpSite]
-->
```

完整写法：

```
<!--
[SharpSite]
{
  "SetParam": "list",
  "Select": "*",
  "From": "T_News",
  "Where": "",
  "OrderBy": "",
  "Page": RequestInt("page"),
  "PageSize": 3,
  "NextPageUrlCount": 5,
}
[/SharpSite]
-->
```

简化后如下，最少 2 个参数：

```
<!--
[SharpSite]
{
  "SetParam": "listNews",
  "From": "C_News",
```

```
}
[/SharpSite]
-->
```

参数说明:

参数名称	示例	是否必填	参数说明
SetParam	listNews	是	将查询结果赋值给页面变量 listNews，页面中输出数据需要循环此变量
Select	*		需要查询的字段：选填，默认为全部字段
From	C_News	是	需要查询的表：必填
Where			过滤条件：选填
OrderBy			排序字段：选填
Page	1		从第几页开始读取数据：选填，可使用动态变量，RequestInt("page")
PageSize	5		每页提取记录数：默认等于 5，分页时使用，显示前后翻页数量
NextPageUrlCount	5		显示前后翻页页码数量，选填，默认等于 5

以上写法支持分页读取数据及翻页，以下为 listNews 变量自动生成的的翻页参数：

参数名称	参数说明
listNews_PrevPage	上一页之前的页码数组，需循环输出
listNews_NextPage	下一页之后的页码数组，需循环输出
listNews_TotalCount	总记录数
listNews_TotalPage	总页数
listNews_CurrentPage	当前页码
listNews_HasPrevPage	是否存在上一页
listNews_HasNextPage	是否存在下一页

3) 方式三，读取单条记录：

```
[SharpSite]
{
    "SetParam": "objNews",          /* 必填，将查询结果赋值给变量listNews，一个页面里面必须唯一，且不能重复 */
    "From": "T_News",              /* 必填，查询表 */
    "Where": "ListID= RequestInt('id') ", /* 必填，查询条件 */
    "ViewCount": "FViewCount",     /* 选填，详细页浏览字段，详细页浏览器次数统计 */
}
[/SharpSite]
```

完整写法：

```
<!--
[SharpSite]
{
    "SetParam": "objNews",
    "From": "T_News",
    "Where": "ListID=RequestInt('id') ",
    "ViewCount": "FviewCount",
}
[/SharpSite]
-->
```

新闻标题: \$Util.Get(\$objNews, "FTitle")
 新闻内容: \$Util.Get(\$objNews, "FContent")
 新闻日期: \$Util.FormatDate(\$Util.Get(\$objNews, "FAddTime"))
 浏览次数: \${Util.Get(\$objNews, "FViewCount")}

参数说明:

参数名称	示例	是否必填	参数说明
SetParam	objNews	是	将查询结果赋值给页面变量 listNews, 页面中输出数据需要循环此变量
From	C_News	是	需要查询的表: 必填
Where	ListID=RequestInt("id")	是	过滤条件: 必填, 要读取的具体记录, 一般从 url 获取
ViewCount	FviewCount		浏览次数字段, 选填, 详细页浏览器次数统计, 自动加 1

4) 页面数据输出

A、循环输出变量: 如列表页, 是多条记录, 需循环输出

```
#foreach($data in $listNews)
    ${data.FTitle} /* 显示新闻标题 */
#end
```

B、输出单条记录: 如果是单条记录的模块, 无需循环, 可直接输出, 如新闻详情页

```
${Util.Get($objNews, "FTitle")} /* 显示新闻标题 */
```

C、输出当前页码:

```
$listNews_CurrentPage
```

D、输出当前页之前的页码:

```
#foreach($page in $listNews_PrevPage)
    <a href="?page=$page">$page</a>
#end
```

E、输出当前页之后的页码:

```
#foreach($page in $listNews_NextPage)
    <a href="?page=$page">$page</a>
#end
```

F、[上一页]判断: 如果存在则可以点击, 如果不存在则不能点击

```
#if($listNews_HasPrevPage)
    #set($PrevPage = $listNews_CurrentPage - 1)
    <a class=" pagination " href="?page=$PrevPage">上一页</a>
```

```
#else
    <span class=" pagination disabled">上一页</span>
#end
```

```
G、[下一页] 判断：如果存在则可以点击，如果不存在则不能点击
#if($listNews_HasNextPage)
    #set($NextPage = $listNews_CurrentPage + 1)
    <a class=" pagination " href="news.html?page=$NextPage">下一页</a>
#else
    <span class=" pagination disabled">下一页</span>
#end
```

注：以上 html 标签和样式请根据自己的网站模板调整。

5) 提交数据到后台

如前端有留言模块，需要向后台提交数据，第一步必须先开启后台模块的 OpenAPI 参数：

编辑 — □ ×

* 模块名称	产品		
排序	100		
数据库表名	T_ Product	T_开头，只能为“数字字母下划线”，为空则自动生成	
自动编号前缀		每条记录都会自动生成唯一记录编号，前缀可为空	
模块类型	多条记录 ▼		
开放API提交	是 ▼		

✓ 确认

🔒 关闭

第二步，引用/admin/static/OpenAPI.js 文件，调用

`OpenAPI.post(TBName, JsonData, SuccessMsg)` 函数提交数据，如果需要验证字段等必填参数，需要自行实现。

根据需求，也可自行修改 `OpenAPI.post` 函数，增加额外功能。

参数说明：

参数名称	示例	是否必填	参数说明
------	----	------	------

TBName	"C_Word"	是	将数据提交到后台模块表名
JsonData	{ "FTitle": "留言标题", "Fcontent": "留言内容"} }	是	Json 格式的数据, 字段名称+字段值
SuccessMsg	"提交成功"	是	提交成功后的提示信息: 必填

6) 数据格式化函数

系统内置\$Util 对象, 提供常用参数格式化函数, 可在页面中直接调用:

函数名称	函数说明
\$Util.Get(\$list, \$FieldName)	输出单条记录, 如输出\$list 变量的 FTitle 字段值 \$Util.Get(\$list, "Ftitle")
\$Util.CheckMenuActive("PageName", \$showCss)	判断导航菜单选中, 如网页使用 header.html 包含文件, 需要判断新闻页选中菜单可使用: \$Util.CheckMenuActive("news", "actived")。 多个页面名称用英文逗号隔开, 如判断首页, 可使用: \$Util.CheckMenuActive("index,", "actived")。 如判断新闻页和新闻详细页都选中, 可使用: \$Util.CheckMenuActive("news,detail", "actived")。
\$Util.RequestInt("ParamName")	获取 Get 或者 Post 整型参数, 如获取当前页码: \$Util.RequestInt("page")
\$Util.RequestStr("ParamName")	获取 Get 或者 Post 字符型参数, 如获取 url 搜索参数: \$Util.RequestInt("SearchKey")
\$Util.Length(\$obj)	获取当前变量的字符长度
\$Util.IndexOf(\$s1, \$s2)	查找变量 s2 在 s1 中出现的位置, 从 0 开始, 未找到返回-1
\$Util.CutString(\$obj, \$length)	截取\$obj 指定长度的字符串值
\$Util.StartsWith(\$obj, \$s1)	判断变量\$obj 是否以\$s1 开头
\$Util.ToHTML(\$obj)	将字符转义成安全的 html 字符输出
\$Util.RemoveHTML(\$obj)	移除变量中的所有 html 字符
\$Util.RemoveJS(\$obj)	移除变量中的所有 JS 标记
\$Util.FormatPrice(\$price)	将变量格式化为 2 位小数, 如: 1,000,000.00
\$Util.FormatPrice(\$price, \$digitalCount)	将变量格式化为指定小数位, 如: 1,000,000.000
\$Util.FormatDate(\$date)	格式化日期变量, 默认格式: yyyy-MM-dd
\$Util.FormatDate(\$date, \$formatStr)	自定义格式化日期变量, 完整格式: yyyy-MM-dd HH:mm:ss 如只需要年月: \$Util.FormatDate(\$date, "yyyy 年 MM 月")
\$Util.ToLower(\$obj)	将变量中的 26 个英文字母全部转化成小写

7) 包含文件的有效利用

包含文件除了可以引用相同页面, 提高效率外, 还可以用来读取整站公共变量, 如在 footer.html 页面中, 有“联系电话”是动态变量, 每个页面都会包含, 那么我们可以直接在 footer.html 页面中加入模板代码:

```
<!--  
[SharpSite]  
{  
    "SetParam": "Company",  
    "From": "T_AboutUS",  
}  
[/SharpSite]  
-->  
输出联系电话: $Util.Get($Company, "FPhoneNumber")
```