## 一、系统参数配置

1、通过配置，可以提高开发效率及降低维护成本。

### 2、使用说明

-- 一些会变化的常量名字配置在AppSettingKeys，数值配置在DB中

例如获取发票开票额限制:

String const InVoiceMaxValue=”InVoiceMaxValue”;

Needs.AppSettings.SettingManager.AppSettings[AppSettingKeys.InvoiceMaxValue].Value

注：

Needs.AppSettings.SettingManager.AppSettings[key].Items//获取配置项集合

使用说明：如果配置单值，使用Needs.Settings.SettingManager.AppSettings[key].Value获取，将会抛出异常。

### 3、数据结构设计

#### 3.1 AppSettings 配置项目主表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 是否可控 | 备注 |
| ID | nvarchar(150) | 否 | 主键 |
| Name | nvarchar(150) | 否 | 配置项名称 |
| Enabled | Bit | 否 | 是否启用 |
| IsMultiValue | Bit | 否 | 是否多值 |
| IsComplex | Bit | 否 | 是否复合值 |
| Summary | nvarchar(150) | 是 | 描述 |

#### **3.2 AppSettingItems** 配置项目子表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 是否可控 | 备注 |
| ID | nvarchar(150) | 否 | 主键 |
| AppSettingID | nvarchar(150) | 否 |  |
| Key | nvarchar(150) | 否 | 名称 |
| Value | nvarchar(200) | 否 | 值 |
| Enabled | Bit | 否 | 是否启用 |
| SortNo | Int | 否 | 排序序号 |
| Summary | nvarchar(150) | 是 | 描述 |

## 2、文件服务器

数据文件存储在独立的服务器，独立备份。解决报关部手动备份文件的问题

域名（创新恒远）：https://files.wl.net.cn

文件访问：https://files.wl.net.cn/orders/2019/08/30/8901009991.pdf

域名（芯达通）：https://files.for-ic.net

文件访问：https://files.for-ic.net/orders/2019/08/30/8901009991.pdf

## 二、webConfig 配置

调试，发布Asp.net程序的时候，开发环境和发布环境的Web.Config往往不同，比如connectionstring等。如果常常有调试，发布的需求，就需要常常修改web.config文件，这往往是一件非常麻烦的事情。

　　Web.Config Transformation能够在不同的发布环境下，产生不同的web.config文件，非常方便和实用。

**1、Web.Debug.config 和Web.Release.config 文件配置**

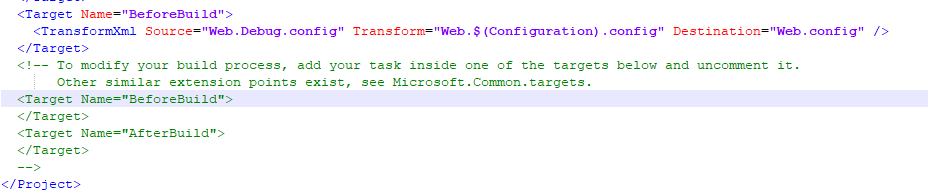
方法如下：

- 打开”xxx.csproj” 工程文件，在最底下加上如下 语句

<Target Name="BeforeBuild">

<TransformXml Source="Web.Debug.config" Transform="Web.$(Configuration).config" Destination="Web.config" />

</Target>



- 展开web.config 会看到两个文件Web.Config.config

和Web.Release.config



- 在这两个文件中,分别配置各自的配置，比如连接字符串

- >Web.Debug.config 中的连接字符串为：



->Web.Release.config 中的连接字符串为：



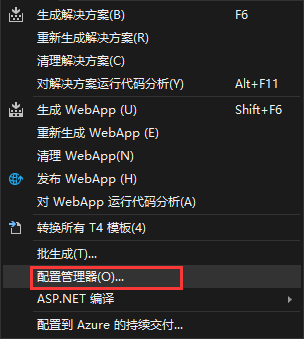
在该项目Debug,重新生成的时候,web.config 字符串会被Web.Debug.config中的替换

该项目Release,重新生成的时候,web.config字符串会被Web.Release.config中的替换

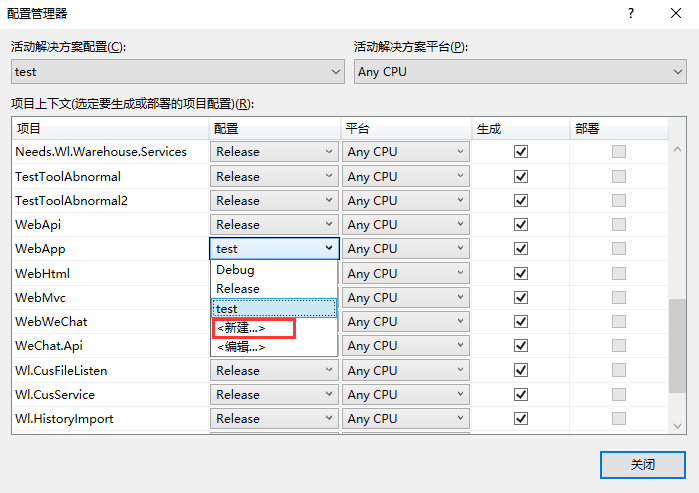
**2、增加Tesst.Debug.config**

自带只支持两个分支config，debug对应测试环境，release对应生产环境。实际情况下我们还有开发环境，也需要一个config分支，添加方法如下

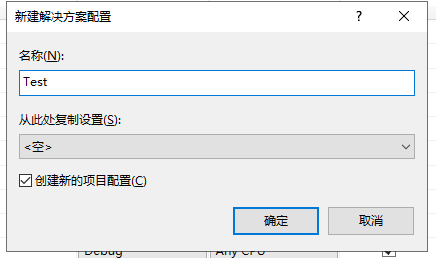
- VS 打开配置管理器



- 配置管理器中，选择相应的项目，在配置一栏，选择“新建”

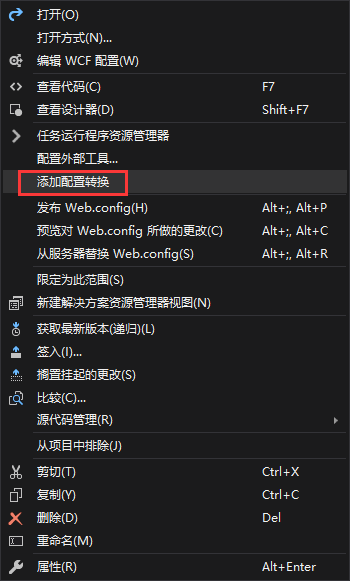


- 设置名称

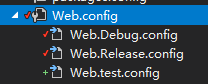


- 添加配置转换

找到相应项目对应的web.config文件，右击，选择添加配置转换



- Web.test.config 文件添加完成



**3、发布时的webconfig配置**

在每个连接字符串后面加上xdt:Transform="SetAttributes" xdt:Locator="Match(name)"

