# 宏书签替换库说明文档

类型结构图

## 类介绍

宏书签替换器工厂类

|  |  |
| --- | --- |
| **类图** |  |
| **功能** | Handle函数提供模板输出功能总入口 |
| **说明** | 上下文类型(ReplacerContext<T>)派生类所在的类库即为处理器(IWordBookmarkHandler<T>)查找程序集,确保处理器实现和上下文类型在一个程序集类库. |

书签模板替换器

|  |  |
| --- | --- |
| **类图** |  |
| **功能** | 该类控制模板替换逻辑/规则,被工厂类引用,一般不需要关注 |
| **说明** | Handle函数提供对外逻辑/规则控制入口,构造函数接受模板书签替换器 |

上下文基类

|  |  |
| --- | --- |
| **类图** |  |
| **功能** | 该类控制模板替换流程中的上下文,需要重点关注 |
| **说明** | 属性说明请参考帮助文档,重点关注的属性和方法如下:   * BookMarkConfigPath: 配置文件路径属性,该属性指示配置文件路径,派生类可重写,默认是程序集类库所在目录下BookMarkConf.xml文件. * PreprocessText:预处理文本函数,执行对配置文件中的文本进行宏替换或者其他预处理操作. * PreprocessSql: 预处理sql函数,同上 * PreprocessPicUri: 预处理图片函数,同上 * CreateCommand:sql执行命令对象 |

**附件:BookMarkConf.xml格式示意图**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8" ?>  <BookMarkConfs>  <BookMarkConf name="a">  <BookMark name="dz\_time" type="TEXT" text="hjhj" sql="" url=""></BookMark>  <BookMark name="districtName2" type="TEXT" text="@SYS.TIME:'yyyyMMdd'" sql="" url=""></BookMark>  <BookMark name="tu1" type="PICTURE" text="" sql="" url="a.png"></BookMark>  <BookMark name="d" type="" text="" sql="" url=""></BookMark>  </BookMarkConf>  </BookMarkConfs> |
|  |

书签配置文件对象

|  |  |
| --- | --- |
| **类图** |  |
| **功能** | 该类为模板配置文件实体类 |
| **说明** | 从左到右依次包含子项 |

模板处理器

|  |  |
| --- | --- |
| **类图** |  |
| **功能** | 该类为模板处理器,用于处理配置文件不能处理的情况,派生自IWordBookmarkHandler<T>接口的子类(一般从WordBookmarkRepositoryHandler<T>派生)将被自动识别进行匹配,优先匹配代码,框架将会查找函数签名满足下列条件的函数作为书签处理函数,函数名称对应书签名称(忽略大小写):   1. 参数个数为三个 2. 第一个参数是下列类型之一: DocumentFormat.OpenXml.Wordprocessing.Text/    1. DocumentFormat.OpenXml.Packaging. ImagePart/DocumentFormat.OpenXml.Wordprocessing.Table 3. 第二个参数是ReplacerContext<T>类型的 4. 第三个参数是Action<string>类型的 |
| **说明** | 查找的处理器范围和派生的上下文类在同一程序集中 |

实例代码:

|  |
| --- |
| [TestClass]      public class UnitTest1      {          [TestMethod]          public void TestMethod1()          {                string filepath = "F:\\test\\src\\a.docx";              string targetDir = "F:\\test";              new WordBookmarkReplacerFac<string>().Handle(new MyReplacerContext(filepath, targetDir)                  , (sender, args) =>              {                  Debug.WriteLine($"当前进度:{args.ProgressPercentage}%::{args.UserState}");              },                  s => { Debug.WriteLine(s); }, s => { });          }      }        public class MyReplacerContext : ReplacerContext<string>      {          public MyReplacerContext(string filepath, string targetDir)              : base(filepath, targetDir)          {          }            protected override string BookMarkConfigPath          {              get              {                  return Path.Combine("D:\\git\\10LabTest\\MapGIS.GM.Word2Pdf\\",                base.BookMarkConfigPath);              }          }            public override string PreprocessPicUri(string url)          {              return Path.Combine("F:\\test\\src\\",                  base.PreprocessPicUri(url));          }            public override IDbCommand CreateCommand()          {              throw new **NotImplementedException**();          }      } |