



链滴

Alizhou.Office-.NetCore 导入导出 excel, 根据 word 模板生成 word

作者: [Alizhou](#)

原文链接: <https://ld246.com/article/1597852245269>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



1.简介

工作中，遇到需要导入导出excecl，导出word需求，还有各种各样的样式，于是有了word模板导出。类工作不涉及到核心业务，但又往往不可缺少

查询了大量资料，有以下框架/库可以用:

名称	是否收费	说明
NPOI Office,功能强大	开源免费	不需要服务器安
Docx Office,功能强大	开源免费	不需要服务器安
Aspose fice,功能很强大	收费	不需要服务器安装O
OpenXml 安装Office,微软提供的组件	开源免费	不需要服务
Spire e, 官方有提供免费版本，不过有些限制	收费	不需要服务器安装Offi

- 参考文档：<https://www.cnblogs.com/holdengong/p/10889838.html>

最后决定采用 **Docx**操作word，**EPPlus**操作excecl

由于工作需要，我封装了 **Alizhou.Office**，目的是最大化节省导入导出这种非核心功能开发时间，专于业务实现，并且业务端与底层基础组件完全解耦，即业务端完全不需要知道底层使用的是什基础，使得重构代价大大降低。

Alizhou.Office底层库目前使用Docx,因此是完全免费的

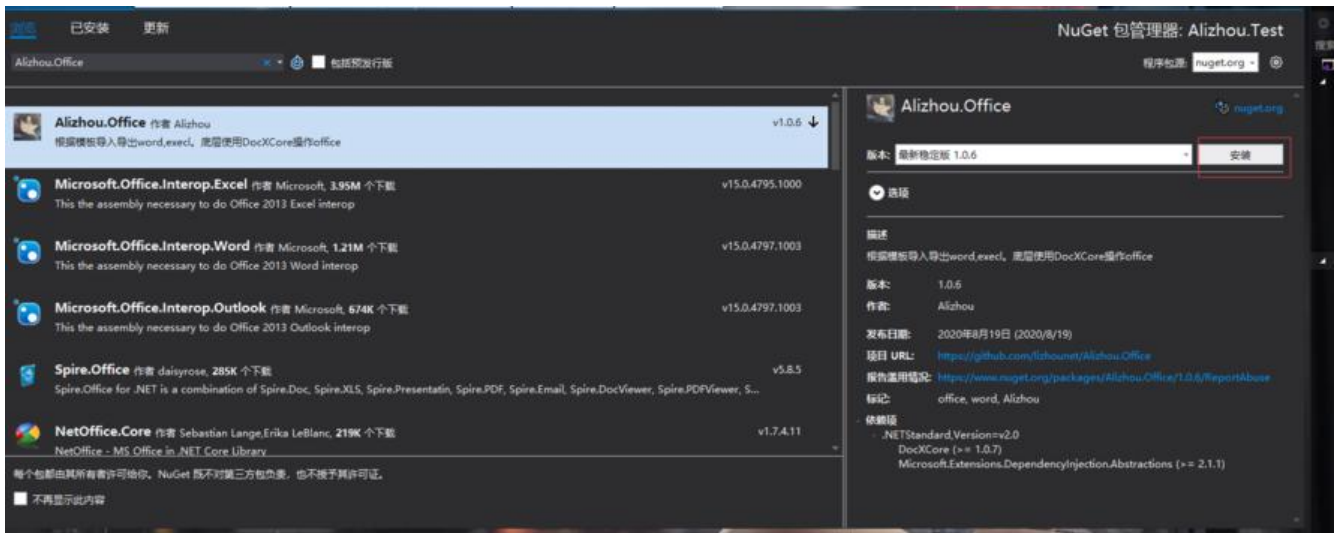
Alizhou.Office目前提供了

- Word根据模板生成：支持使用文本/图片/表格替换，占位符只需定义模板类，制作Word模板，一代码导出docx文档；
- 根据实体对excel导入导出；

2.如何使用

2.1安装Alizhou.Office

nuget 搜索并安装 [Alizhou.Office](#)



2.2依赖注入

```
// 注入Office基础服务
services.AddAlizhouOffice();
```

2.3 IWordExportService - Word通用导出服务

2.3.1定义模板类,模板类需实现IWordExportTemplate空接口

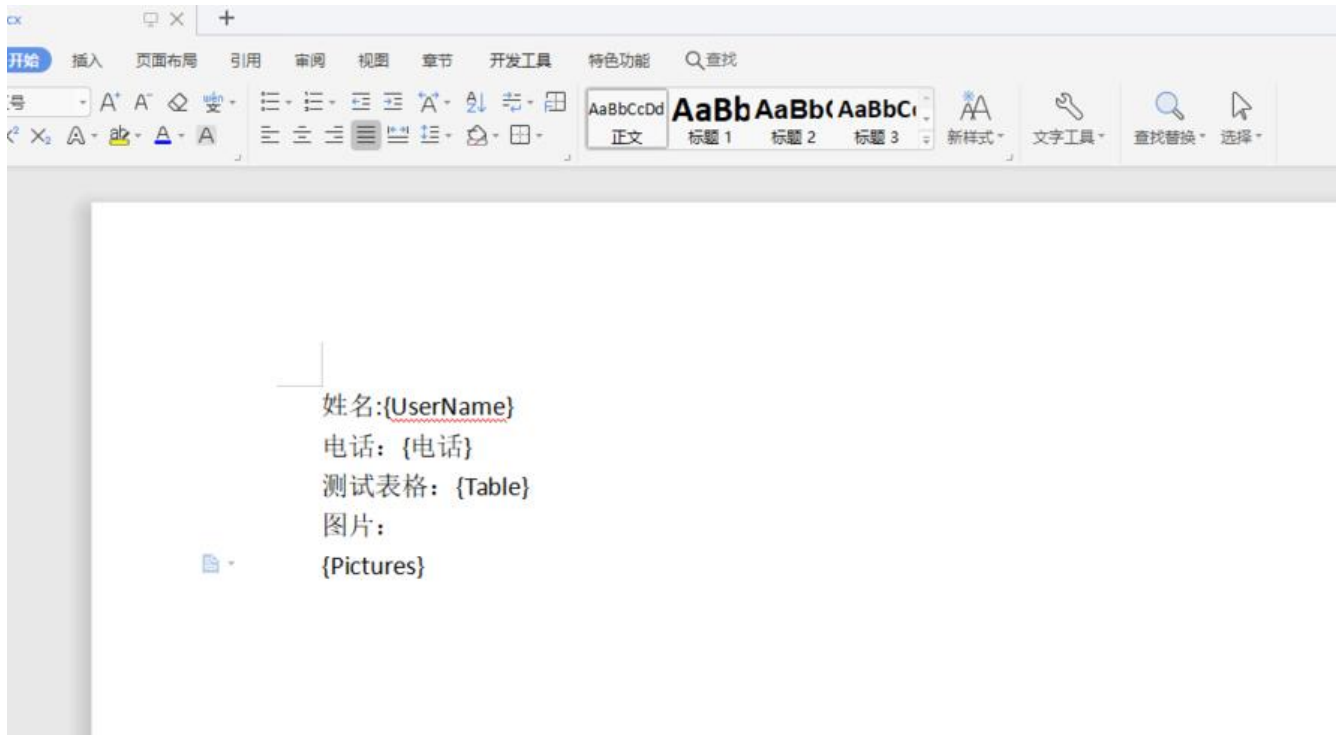
```
public class WordUserTemplate : IWordExportTemplate
{
    /// <summary>
    /// 默认占位符为{PropertyName}
    /// </summary>
    public string UserName { get; set; }
    /// <summary>
    /// 默认
    /// </summary>
    [Placeholder("{电话}")]
    public string Phone { set; get; }
    /// <summary>
    /// 表格
    /// </summary>
    public AlizhouTable Table { get; set; }
}
```

```

/// <summary>
/// 图片
/// </summary>
public IEnumerable<AlizhouPicture> Pictures { get; set; }
}

```

2.3.2制作模板



2.3.3根据模板生成word

```

string basePath = Environment.CurrentDirectory;
string templateUrl = @$"{basePath}/template/word/TemplateWrod.docx";
IWordExportService wordExportService = new WordExportService(new WordExportPr
vider());
WordUserTemplate userTemplate = new WordUserTemplate
{
    UserName = "小周黎",
    Phone = "175626565656",
    Pictures = new List<AlizhouPicture>() {
        new Office.Model.AlizhouPicture
        {
            PictureUrl = "D://图片1.png",
            Width=540,
            Height=405,
        },
        new Office.Model.AlizhouPicture
        {
            PictureUrl = "D://图片2.png",
            Width=540,
            Height=405,
        }
    }
}

```

```

    },
    Table = new Office.Model.AlizhouTable(3, 2)
    {
        Rows = new System.Collections.Generic.List<Office.Model.AlizhouTableRow>() {
            new Office.Model.AlizhouTableRow{
                Height=200,
                Cells=new System.Collections.Generic.List<Office.Model.AlizhouTableCell>()

                    new Office.Model.AlizhouTableCell{
                        Width=100,
                        Paragraphs=new System.Collections.Generic.List<Office.Model.Alizhou
aragraph>(){
                            new Office.Model.AlizhouParagraph{
                                Run=new Office.Model.AlizhouRun{
                                    Text="姓名",
                                    IsBold=true
                                }
                            }
                        }
                    },
                    new Office.Model.AlizhouTableCell{
                        Width=100,
                        Paragraphs=new System.Collections.Generic.List<Office.Model.Alizhou
aragraph>(){
                            new Office.Model.AlizhouParagraph{
                                Run=new Office.Model.AlizhouRun{
                                    Text="年龄"
                                }
                            }
                        }
                    }
                },
            new Office.Model.AlizhouTableRow{
                Height=200,
                Cells=new System.Collections.Generic.List<Office.Model.AlizhouTableCell>()

                    new Office.Model.AlizhouTableCell{
                        Width=100,
                        Paragraphs=new System.Collections.Generic.List<Office.Model.Alizhou
aragraph>(){
                            new Office.Model.AlizhouParagraph{
                                Run=new Office.Model.AlizhouRun{
                                    Text="周黎"
                                }
                            }
                        }
                    },
                    new Office.Model.AlizhouTableCell{
                        Width=100,
                        Paragraphs=new System.Collections.Generic.List<Office.Model.Alizhou

```

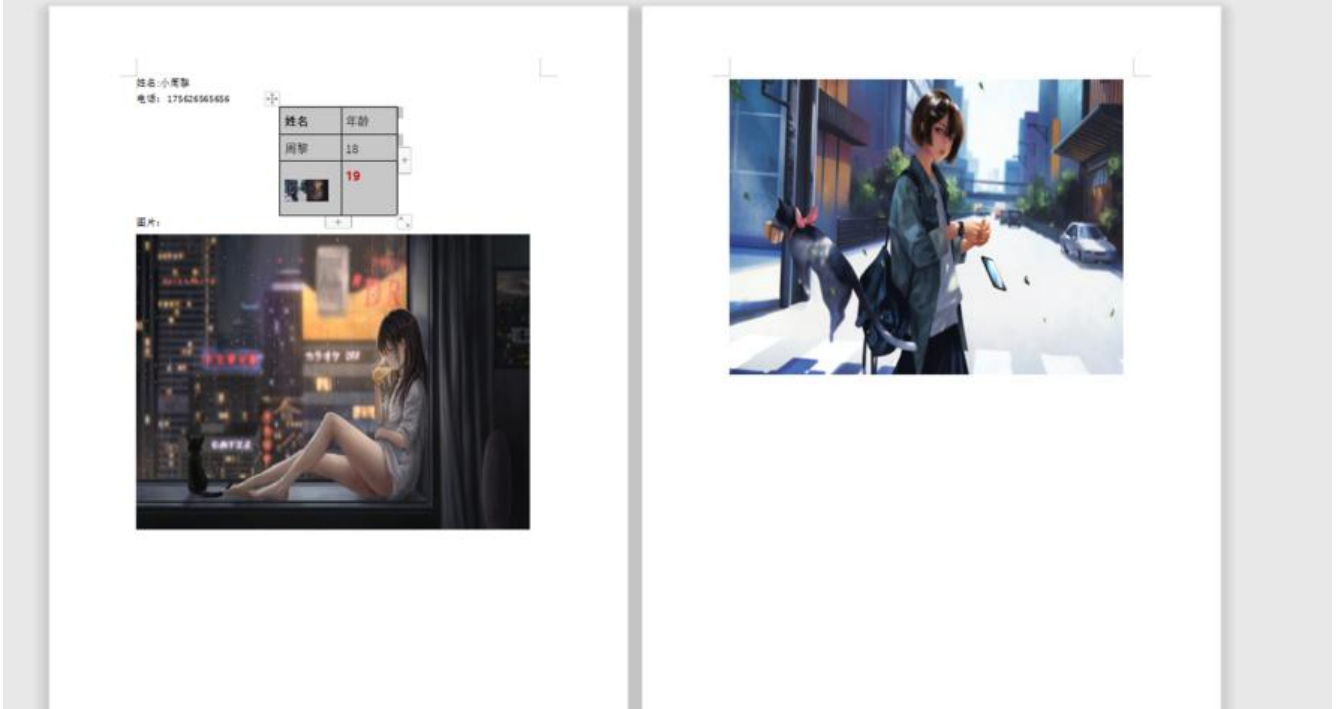


```

    }
};
var word = wordExportService.TemplateCreateWord(templateUrl, userTemplate);
File.WriteAllBytes($"{basePath}/{DateTime.Now.ToString("yyyyMMddHHmmss")}测试
成word.docx", word.WordBytes);
Console.ReadKey();

```

2.3.3生成效果



2.4 IExcelImportExportService- execl通用导入导出服务

2.4.1定义execl模板类

```

public class PersonDto
{
    [ExcelTableColumn("姓名")]
    [Required(ErrorMessage = "姓名必填")]

    public string Name { get; set; }
    [ExcelTableColumn("年龄")]
    [Range(1, 25, ErrorMessage = "年龄超过25岁的MM不要")]//范围判断
    public int Age { get; set; }

    [ExcelTableColumn(3)]//根据索引取
    public string Adress { get; set; }
    [ExcelTableColumn("电话")]
    [MaxLength(11, ErrorMessage = "手机号 长度不能超过11")]
    public string Phone { set; get; }
    public override string ToString()
    {
        return $"姓名: {Name}; 年龄: {Age}; 地址: {Adress}; 电话: {Phone}";
    }
}

```

```
}  
}
```

2.4.2 导入导出execl

```
string basePath = Environment.CurrentDirectory;  
var list = new List<PersonDto>();  
for (int i = 0; i < 100; i++)  
{  
    list.Add(new PersonDto  
    {  
        Name = $"张三{i}",  
        Age = 18,  
        Adress = $"地址{i}",  
        Phone = $"17783042962"  
    });  
}  
IExeclImportExportService execlImportExport = new ExeclImportExportService(new Ex  
clImportExportProvider());  
var alizhouExecl = execlImportExport.Export(list);  
string path = $"{basePath}..\..\..\OutPut\execl\ExeclImportExportTest.xlsx";  
File.WriteAllBytes(path, alizhouExecl.WordBytes);  
var data = execlImportExport.Import<PersonDto>(File.OpenRead(path));  
foreach (var item in data)  
{  
    Console.WriteLine(item.ToString());  
}
```

2.4.3 导出效果

ExecelImportExportTest

文件 开始 插入 页面布局 公式 数据 审阅 视图 团队 设计

Calibri 11 A⁺ A⁻ 自动换行 常规 条件格式 套用表格格式 检查单元格

姓名

姓名	年龄	Address	电话
张三0	18	地址0	17783042962
张三1	18	地址1	17783042962
张三2	18	地址2	17783042962
张三3	18	地址3	17783042962
张三4	18	地址4	17783042962
张三5	18	地址5	17783042962
张三6	18	地址6	17783042962
张三7	18	地址7	17783042962
张三8	18	地址8	17783042962
张三9	18	地址9	17783042962
张三10	18	地址10	17783042962
张三11	18	地址11	17783042962
张三12	18	地址12	17783042962
张三13	18	地址13	17783042962
张三14	18	地址14	17783042962
张三15	18	地址15	17783042962
张三16	18	地址16	17783042962
张三17	18	地址17	17783042962
张三18	18	地址18	17783042962
张三19	18	地址19	17783042962
张三20	18	地址20	17783042962
张三21	18	地址21	17783042962

2.4.3导入效果

```

C:\Program Files\dotnet\dotnet.exe
使用
1 安装
2 依赖
3 IWo
2.3
2.3
2.3
2.3
4 IExe
2.4
2.4
2.4
2.4
: 0; 年龄: 18; 地址: 地址0; 电话: 17783042962
: 1; 年龄: 18; 地址: 地址1; 电话: 17783042962
: 2; 年龄: 18; 地址: 地址2; 电话: 17783042962
: 3; 年龄: 18; 地址: 地址3; 电话: 17783042962
: 4; 年龄: 18; 地址: 地址4; 电话: 17783042962
: 5; 年龄: 18; 地址: 地址5; 电话: 17783042962
: 6; 年龄: 18; 地址: 地址6; 电话: 17783042962
: 7; 年龄: 18; 地址: 地址7; 电话: 17783042962
: 8; 年龄: 18; 地址: 地址8; 电话: 17783042962
: 9; 年龄: 18; 地址: 地址9; 电话: 17783042962
: 10; 年龄: 18; 地址: 地址10; 电话: 17783042962
: 11; 年龄: 18; 地址: 地址11; 电话: 17783042962
: 12; 年龄: 18; 地址: 地址12; 电话: 17783042962
: 13; 年龄: 18; 地址: 地址13; 电话: 17783042962
: 14; 年龄: 18; 地址: 地址14; 电话: 17783042962
: 15; 年龄: 18; 地址: 地址15; 电话: 17783042962
: 16; 年龄: 18; 地址: 地址16; 电话: 17783042962
: 17; 年龄: 18; 地址: 地址17; 电话: 17783042962
: 18; 年龄: 18; 地址: 地址18; 电话: 17783042962
: 19; 年龄: 18; 地址: 地址19; 电话: 17783042962
: 20; 年龄: 18; 地址: 地址20; 电话: 17783042962
: 21; 年龄: 18; 地址: 地址21; 电话: 17783042962
: 22; 年龄: 18; 地址: 地址22; 电话: 17783042962
: 23; 年龄: 18; 地址: 地址23; 电话: 17783042962
: 24; 年龄: 18; 地址: 地址24; 电话: 17783042962
: 25; 年龄: 18; 地址: 地址25; 电话: 17783042962
: 26; 年龄: 18; 地址: 地址26; 电话: 17783042962
: 27; 年龄: 18; 地址: 地址27; 电话: 17783042962
: 28; 年龄: 18; 地址: 地址28; 电话: 17783042962
: 29; 年龄: 18; 地址: 地址29; 电话: 17783042962

```

最后附上源码地址:<https://github.com/lizhounet/Alizhou.Office>