

Excel数据处理利器：C#中三款热门开源类库推荐及实例代码解析

作者：微信公众号：【架构师老卢】

11-24 8:25

250



概述：Excel数据处理利器：C#中三款热门开源类库推荐及实例代码解析,这些类库都提供了丰富的功能，你可以根据项目的需求选择合适的类库。以上示例代码中的注释是为了说明基本功能，实际应用中你可能需要根据业务需求进行更复杂的操作。

在C#中有一些优秀的开源Excel处理类库，以下是其中一些热门的类库：

1. EPPlus:

- **功能：**EPPlus是一个用于处理Excel文件的功能强大且易于使用的开源库。它支持Excel 2007及以上版本（.xlsx格式）。
- **适用范围：**适用于创建、读取和编辑Excel文件，支持图表、样式、公式等高级功能。
- **下载地址：**[EPPlus GitHub](#)
- **实例代码：**

```
1 using OfficeOpenXml;
2
3 // 数据导出到Excel
4 using (var package = new ExcelPackage())
5 {
6     var worksheet = package.Workbook.Worksheets.Add("Sheet1");
7     worksheet.Cells["A1"].Value = "Hello";
8     worksheet.Cells["A2"].Value = "World";
9     package.SaveAs(new FileInfo("output.xlsx"));
10 }
11
12 // 从Excel导入数据
13 using (var package = new ExcelPackage(new FileInfo("input.xlsx")))
14 {
15     var worksheet = package.Workbook.Worksheets[0];
16     var value1 = worksheet.Cells["A1"].Value.ToString();
17     var value2 = worksheet.Cells["A2"].Value.ToString();
18     Console.WriteLine(value1 + " " + value2);
19 }
```

2. NPOI:

- **功能：**NPOI是一个.NET实现的开源类库，支持Excel、Word和PowerPoint文件的读写。
- **适用范围：**适用于处理多种Office文档格式，包括Excel。
- **下载地址：**[NPOI GitHub](#)
- **实例代码：**

```
1 using NPOI.SS.UserModel;
2 using NPOI.XSSF.UserModel;
3 using NPOI.HSSF.UserModel;
4
5 // 数据导出到Excel
6 IWorkbook workbook = new XSSFWorkbook(); // 或者 new HSSFWorkbook(), 取决于Excel版本
7 ISheet sheet = workbook.CreateSheet("Sheet1");
8 sheet.CreateRow(0).CreateCell(0).SetCellValue("Hello");
9 sheet.CreateRow(1).CreateCell(0).SetCellValue("World");
10 using (var fs = new FileStream("output.xlsx", FileMode.Create, FileAccess.Write))
11 {
12     workbook.Write(fs);
13 }
14
15 // 从Excel导入数据
16 using (var fs = new FileStream("input.xlsx", FileMode.Open, FileAccess.Read))
17 {
18     IWorkbook workbook = WorkbookFactory.Create(fs);
19     ISheet sheet = workbook.GetSheetAt(0);
20     var value1 = sheet.GetRow(0).GetCell(0).ToString();
21     var value2 = sheet.GetRow(1).GetCell(0).ToString();
22     Console.WriteLine(value1 + " " + value2);
23 }
```

3. ClosedXML:

- **功能：**ClosedXML是一个简单的开源库，用于在.NET应用程序中创建和处理Excel文件。
- **适用范围：**适用于创建、读取和编辑Excel文件，提供简洁的API。
- **下载地址：**[ClosedXML GitHub](#)
- **实例代码：**

```
1 using ClosedXML.Excel;
2
3 // 数据导出到Excel
4 using (var workbook = new XLWorkbook())
5 {
6     var worksheet = workbook.Worksheets.Add("Sheet1");
7     worksheet.Cell("A1").Value = "Hello";
8     worksheet.Cell("A2").Value = "World";
9     workbook.SaveAs("output.xlsx");
10 }
11
12 // 从Excel导入数据
13 using (var workbook = new XLWorkbook("input.xlsx"))
14 {
15     var worksheet = workbook.Worksheet(1);
16     var value1 = worksheet.Cell("A1").Value.ToString();
17     var value2 = worksheet.Cell("A2").Value.ToString();
18     Console.WriteLine(value1 + " " + value2);
19 }
```

这些类库都提供了丰富的功能，你可以根据项目的需求选择合适的类库。以上示例代码中的注释是为了说明基本功能，实际应用中你可能需要根据业务需求进行更复杂的操作。

