ModifyTablesData 方法

方法

ModifyTablesData(modifyData)

描述

通过指定条件(主键列名和值),操作(插入、更新、删除)多张指定表的多条记录。

参数说明

参数	类型	说明
modifyData	Dictionary <string, modifydata=""></string,>	包含所需更新的数据表名、插入的记录、更新的记录或删除的记录的信息。

示例

下面的示例代码中,通过ModifyTablesData方法,操作(插入、更新、删除)多张指定数据表的多条记录。

```
public class SampleApi : ForguncyApi
        [Post]
        public void ModifyDataSample()
            var data = new Dictionary<string, ModifyData>();
            var modifyData = new ModifyData();
            //
            modifyData.AddRows.Add(new Dictionary<string, object>
        {
            {"", "" },
            {"", "" }
        });
            modifyData.DeleteRows.Add(new ColumnValuePair()
                ColumnName = "ID",
                Value = 2
            });
            //
            modifyData.EditRows.Add(new EditData()
                PrimaryKey = new ColumnValuePair()
                    ColumnName = "ID",
                    Value = 3
                },
                Values = new Dictionary<string, object>
                {"", "" },
                {"", "" }
            });
            data.Add("", modifyData);
            //ModifyTablesData
            this.DataAccess.ModifyTablesData(data);
        }
    }
```

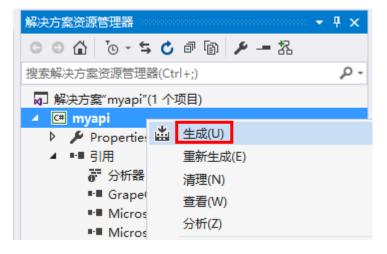
操作步骤

创建服务端Web API应用程序集,具体请参见创建服务端Web API应用程序集。

在.cs文件中输入代码。

```
myapi - Microsoft Visual Studio
文件(E) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 团队(M) 工具(T) 体系结构(C) 测试(S) 分析(N) 窗口(W) 帮助(H)
💪 - 🗇 📸 - 當 💾 🥍 🥠 - 🦿 - Debug - Any CPU
                                                      工具箱
   源代码管理器
                   Class1.cs ≠ X
   c# myapi
                                                                   🕶 🔩 myapi.SampleApi
             □using GrapeCity.Forguncy.ServerApi;
        2
              using System. Collections. Generic;
        3
        4
             ⊟namespace myapi
        5
        6
                  public class SampleApi : ForguncyApi
        8
                      [Post]
                      0 个引用
                      public void ModifyDataSample()
        9
        10
        11
                         var data = new Dictionary(string, ModifyData)();
                         var modifyData = new ModifyData();
        12
        13
                         //添加数据
                         modifyData.AddRows.Add(new Dictionary<string, object>
        14
        15
        16
                          {"姓名","小王"},
{"部门","开发部"}
        17
        18
                      }):
        19
                         //删除数据
       20
                          modifyData.DeleteRows.Add(new ColumnValuePair()
       21
        22
                             ColumnName = "ID",
       23
                             Value = 2
        24
                         1):
        25
                          //修改数据
                          modifyData.EditRows.Add(new EditData()
       26
       27
       28
                             PrimaryKey = new ColumnValuePair()
       29
                                 ColumnName = "ID",
        30
        31
                                 Value = 3
        32
                             }.
       33
                             Values = new Dictionary string, object>
        34
                             {"姓名","小李"},
{"部门","市场部"}
       35
       36
        37
       38
                          1):
                          data.Add("员工表", modifyData);
       39
       40
                          //使用ModifyTablesData更新多条记录
                         this.DataAccess.ModifyTablesData(data);
       41 😨
        42
                  }
        43
```

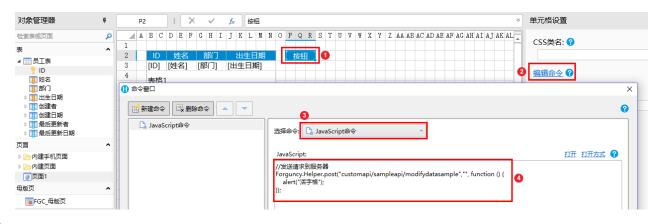
在解决方案资源管理器中,右击解决方案,在右键菜单中选择"生成"。



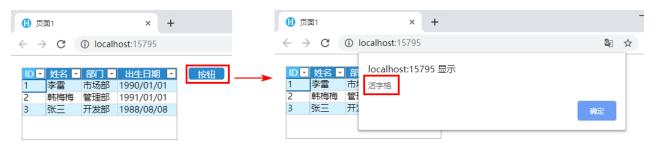
】 在设计器中,选择"文件->设置->自定义Web Api",单击"上传Web Api Assembly",上传生成的dll文件。



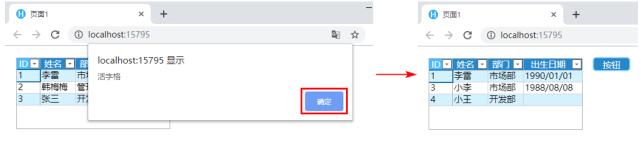
5 在页面中,选择一个单元格区域,将其单元格类型设置为按钮,设置其命令为JavaScript命令,并输入JavaScript代码。



运行页面,单击按钮,就会弹出警告框。



关闭警告框,页面上的表格刷新后,就可以看到员工表新增了ID为4,姓名为小王的记录;删除了ID为2的记录,更新了ID为3的姓名为小李,部门更新为市场部。



- 结束 -----