

创建服务端Web API

- 1. 描述
- 2. 创建服务端Web API



1. 描述

介绍如何创建服务端Web API。



2. 创建服务端Web API

操作步骤

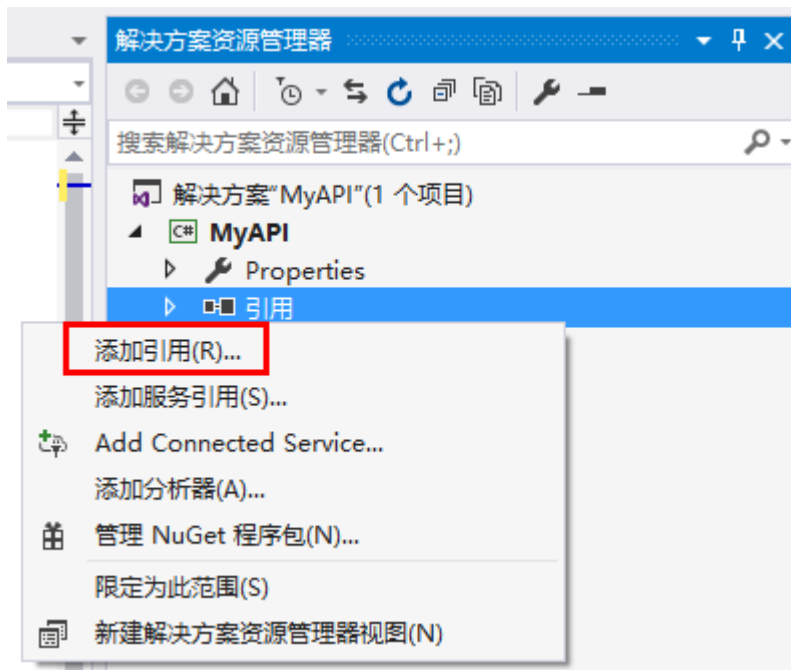
1

在Visual Studio中创建类库项目。设置项目的Framework框架为.NET Framework 4.7.2，如下图所示。



2

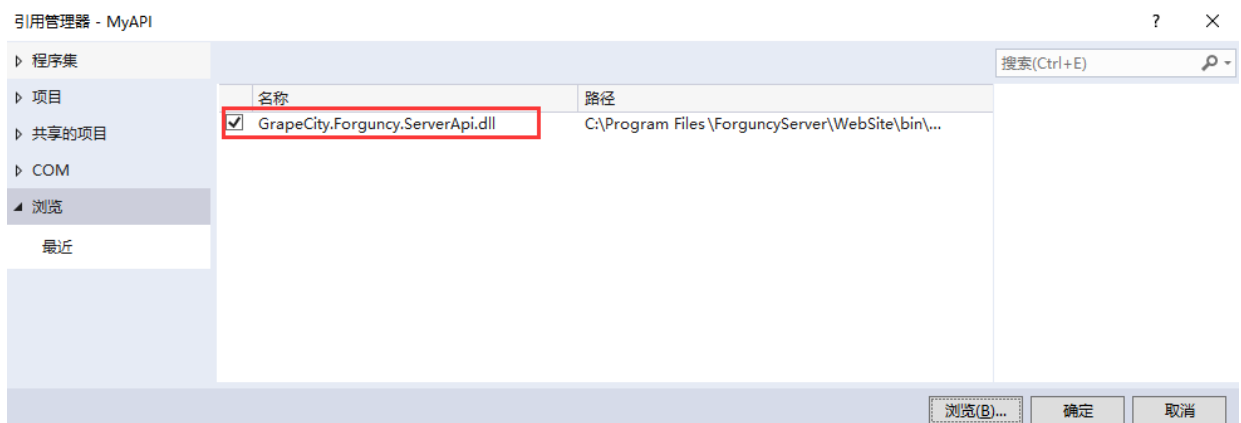
在解决方案资源管理器下，右击“引用”，选择“添加引用”。



3

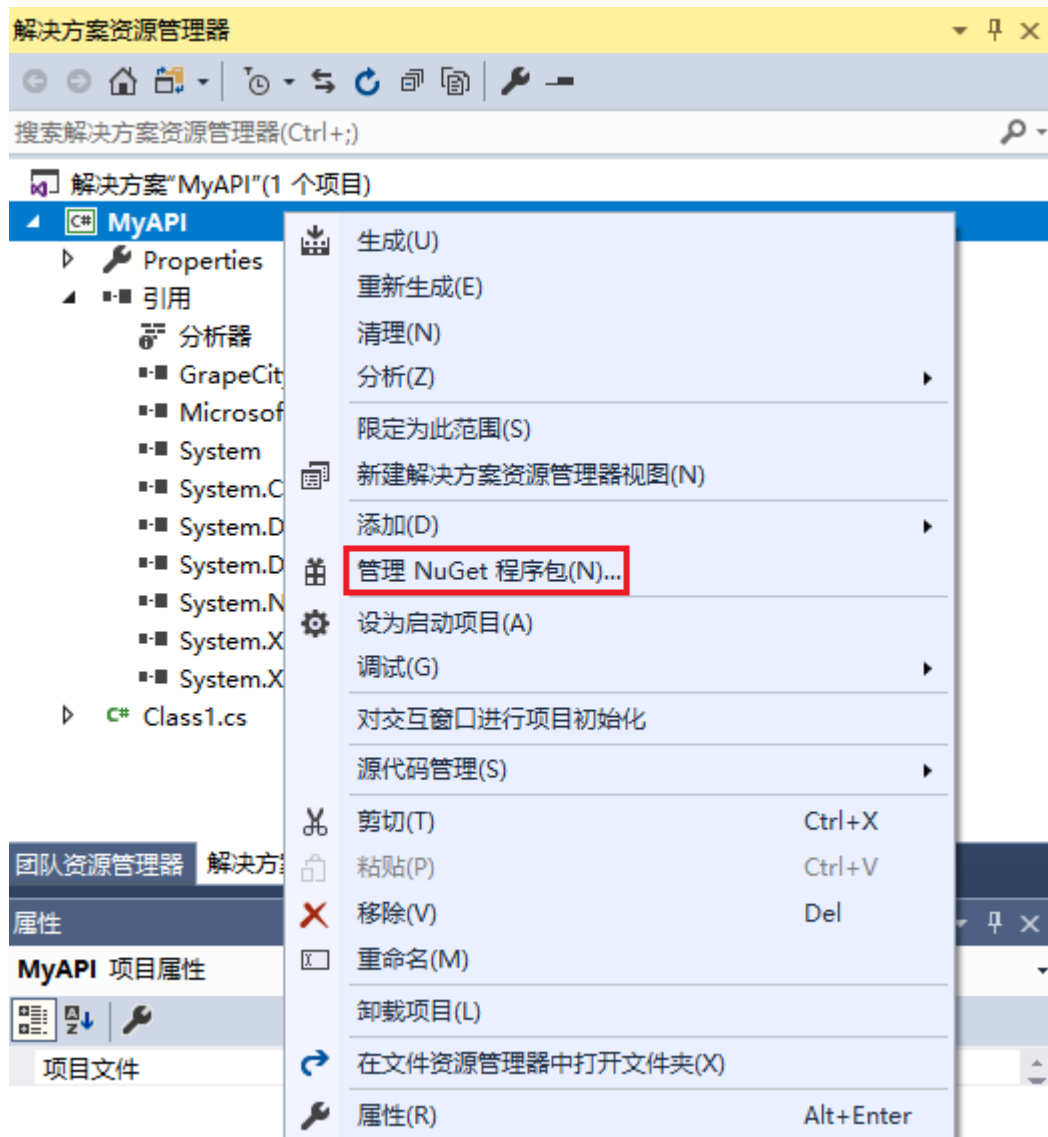
单击“浏览”，在活字格的安装目录下找到“GrapeCity.Forguncy.ServerApi.dll”文件并将其添加为该工程的引用。

- 如果安装活字格服务端时，安装目录为默认目录，则此文件的路径为：C:\Program Files\ForguncyServer\Website\bin
- 如果安装活字格服务端时，安装目录为自定义路径，则此文件的路径为“自定义路径\ForguncyServer\Website\bin”。

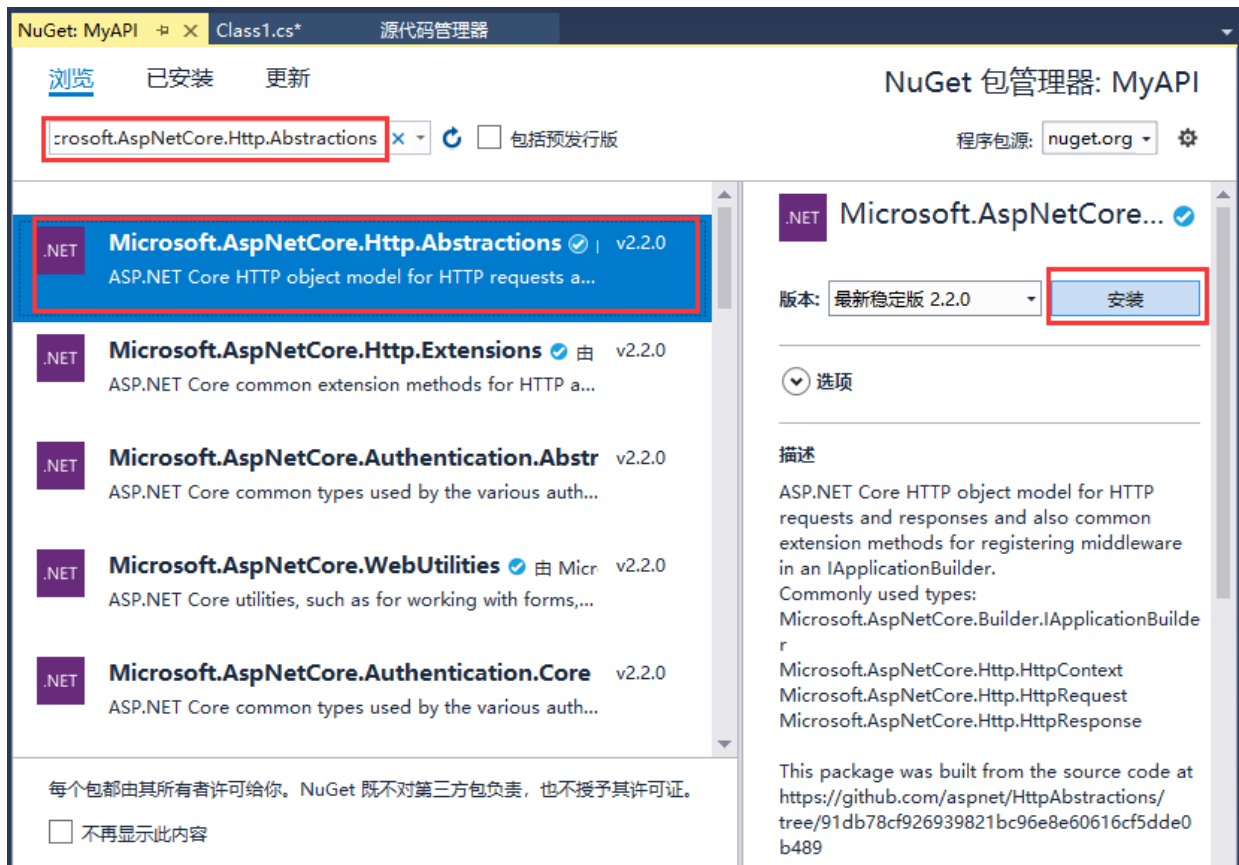


4

在解决方案资源管理器中，选中“MyAPI”，右击，选择“管理NuGet程序包”。



在NuGet管理器中，搜索并安装“Microsoft.AspNetCore.Http.Abstractions”。



5

创建一个public class，使其从GrapeCity.Forguncy.ServerApi命名空间的ForguncyApi类继承。

在类里加入public方法。通过给方法增加Get或Post特性可以把方法指定为可以通过HTTP协议调用的方法。

活字格服务端API

服务端API可以通过ForguncyApi类上的DataAccess属性来操作数据库。

获取HTTP请求信息

当通过HTTP的GET或POST调用方法时，可以通过ServerApi的Context属性获取请求的详细信息。Context属性是Asp.net Core的HttpContext类型，通过Request属性可以获取HTTP请求的全部信息。

API调用的URL

http://域名或主机名/应用程序名/类名/方法名

例如：http://computer1/活字格app1/customapi/sampleapi/addsampledata

下面的C#代码中，自定义Web API类 “MyAPI” ，该类包含一个post方法 “TestPostAPI”：

```

public class MyAPI : ForguncyApi
{
    [Post]
    public async Task TestPostAPI()
    {
        //post
        var form = await Context.Request.ReadFormAsync();
        var name = form["name"][0];
        var department = form["department"][0];
        await this.Context.Response.WriteAsync(name +
department);
        //AddTableData
        this.DataAccess.AddTableData("", new Dictionary<string,
object> { { "", name }, { "", department } });
    }
}

```

如果要使用WriteAsync方法，请添加“using Microsoft.AspNetCore.Http;”。

在前端使用以下JavaScript代码调用TestPostAPI方法，具体关于post方法请参见post方法。

```

//
var page = Forguncy.Page;
//
var cell1 = page.getCell("name");
var cell2 = page.getCell("department");
//
var data = {
    name: cell1.getValue(),
    department: cell2.getValue()
};
//
Forguncy.Helper.post("customapi/myapi/testpostapi", data, function ()
{
    alert("");
});

```



说明

post方法的url参数“customapi/myapi/testpostapi”中，myapi为自定义的类名，testpostapi为方法名。

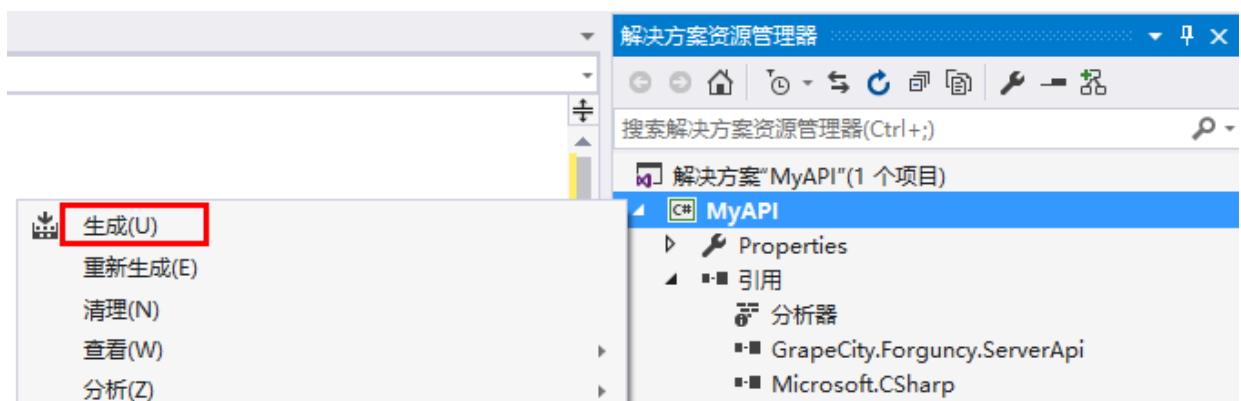
在.cs文件中输入代码。

```
Class1.cs* x
MyAPI
TestPostAPI()

7  using System.Threading.Tasks;
8
9  namespace MyAPI
10 {
11      0 个引用
12      public class MyAPI : ForguncyApi
13      {
14          [Post]
15          0 个引用
16          public async Task TestPostAPI()
17          {
18              //获取post请求的数据
19              var form = await Context.Request.ReadFormAsync();
20              var name = form["name"][0];
21              var department = form["department"][0];
22              await this.Context.Response.WriteAsync(name + department);
23              //使用AddTableData方法向员工表中添加数据
24              this.DataAccess.AddTableData("员工表", new Dictionary<string, object> { { "姓名", name }, { "部门", department } });
25          }
26      }
27  }
```

6

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”。



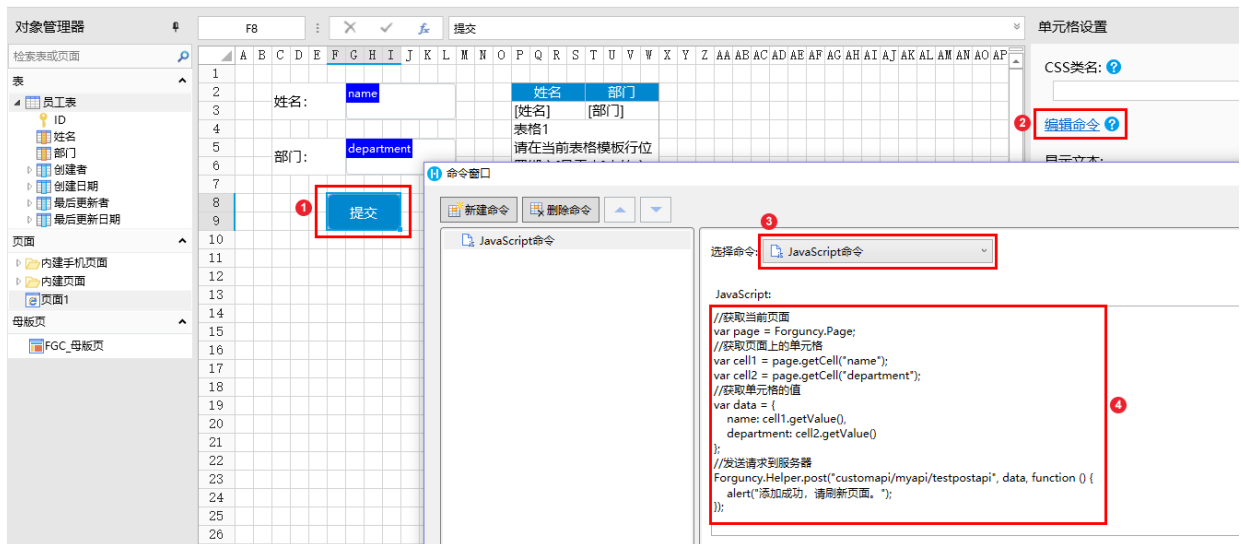
7

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。



8

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。

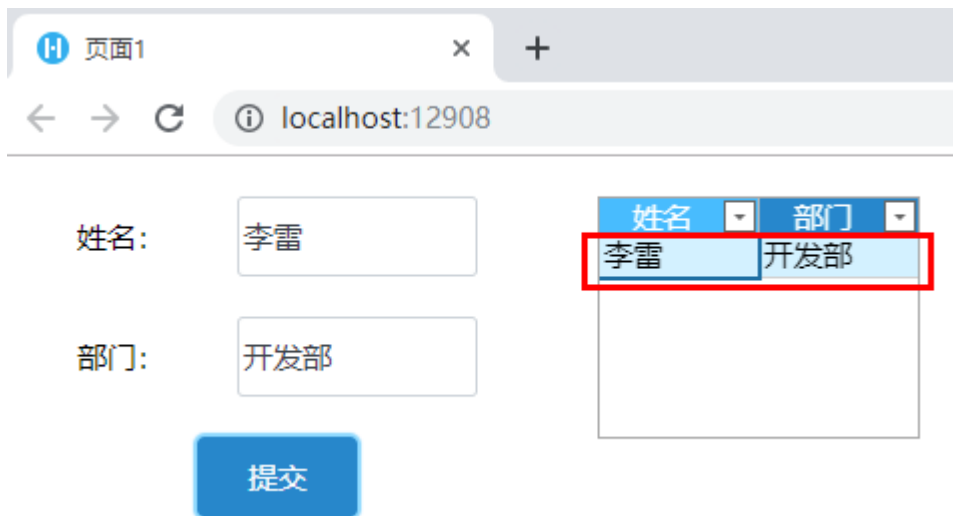


9

运行页面，单击提交按钮，就会弹出警告框。



关闭警告框，表格刷新后，就可以看到员工表中新增的数据。



结束

[回到顶部](#)