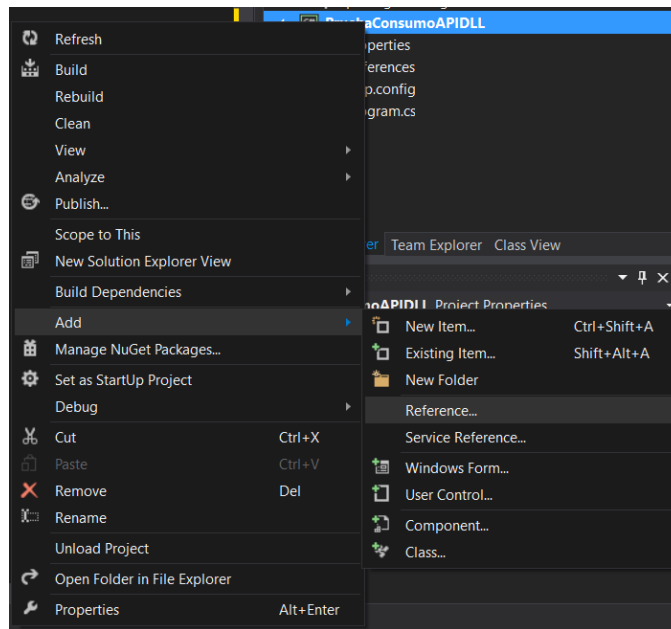
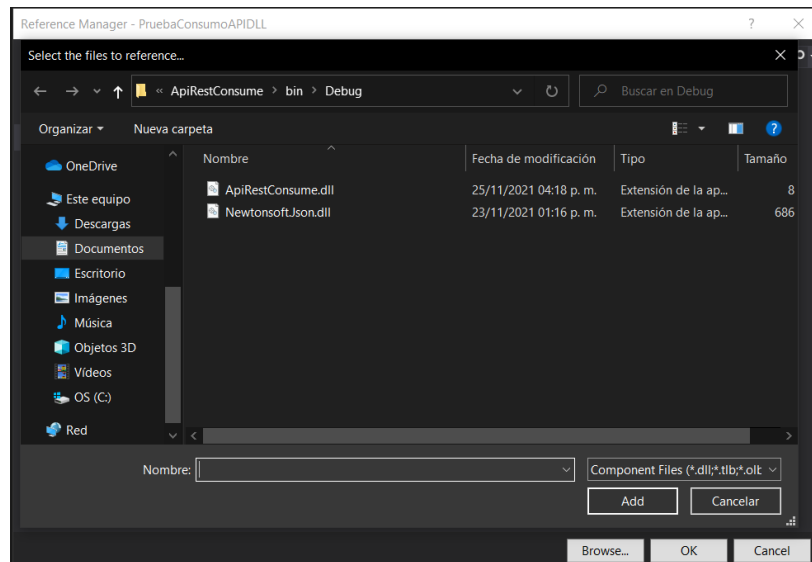


Implementación

Para realizar una petición a través de la librería ApiRestConsume primero es necesario agregar la librería dentro de las referencias del proyecto.



Y buscamos el archivo ApiRestConsume.dll



Una vez hecho esto, tenemos que agregarlo a los namespaces dentro del archivo en el que haremos la implementación.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ApiRestConsume;
using System.Data;
```

Ahora ya podemos empezar a utilizar la clase Consume, la cual se encarga de manejar las peticiones y retornar las respuestas. Empezamos generando una nueva instancia de la clase.

```
Consume consume = new Consume();
```

Nota: la clase Consume tiene una propiedad llamada sslVerification, con la que se le indica al objeto si debe usar o no la verificación del certificado SSL del EndPoint al que se va a conectar, si se requiere saltar la verificación debe colocarse en false, en el caso contrario se puede dejar el valor por defecto (true es el valor por defecto) o indicar explícitamente la propiedad como true.

```
consume.sslVerification = false;
```

Para realizar la petición utilizamos el método conectar, el cual regresa la respuesta en forma de un String con un JSON. Para el usar el método GET se llama al método de la siguiente forma:

```
string url = "https://url.prueba/";  
String mensaje = consume.conectar(url, "get");
```

Si queremos usar el método POST, tenemos dos opciones para enviar parámetros mediante el cuerpo de la petición, usar un JSON y usar un Dictionary.

En el caso de querer usar un JSON:

```
string data = "{\"parametro\":5}";  
string url = " https://url.prueba/";  
Consume consume = new Consume();  
String mensaje = consume.conectar(url, "post", data);
```

En el caso de querer usar un Dictionary, primero hay que crear el objeto de tipo Dictionary<String, String>, con los datos que queramos enviar.

```
Dictionary<String, String> dataDictionary = new Dictionary<string, string>  
{  
    {"variable1", "52"},  
    {"variable2", "52"}  
};
```

Y luego hacemos la implementación, mandando la variable dataDictionary como parámetro:

```
string url = "https://url.prueba/";  
Consume consume = new Consume();  
String mensaje = consume.conectar(url, "post", dataDictionary);
```