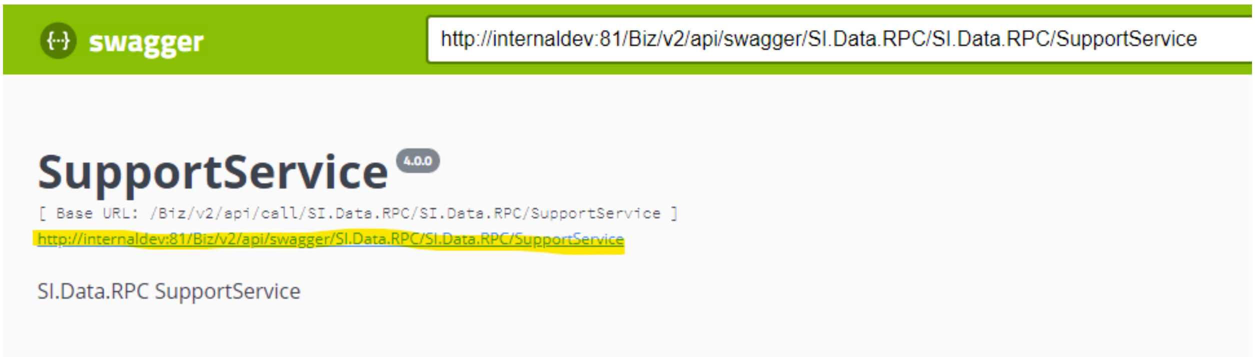
Auto generate code for SIF RPC in VS

Contents

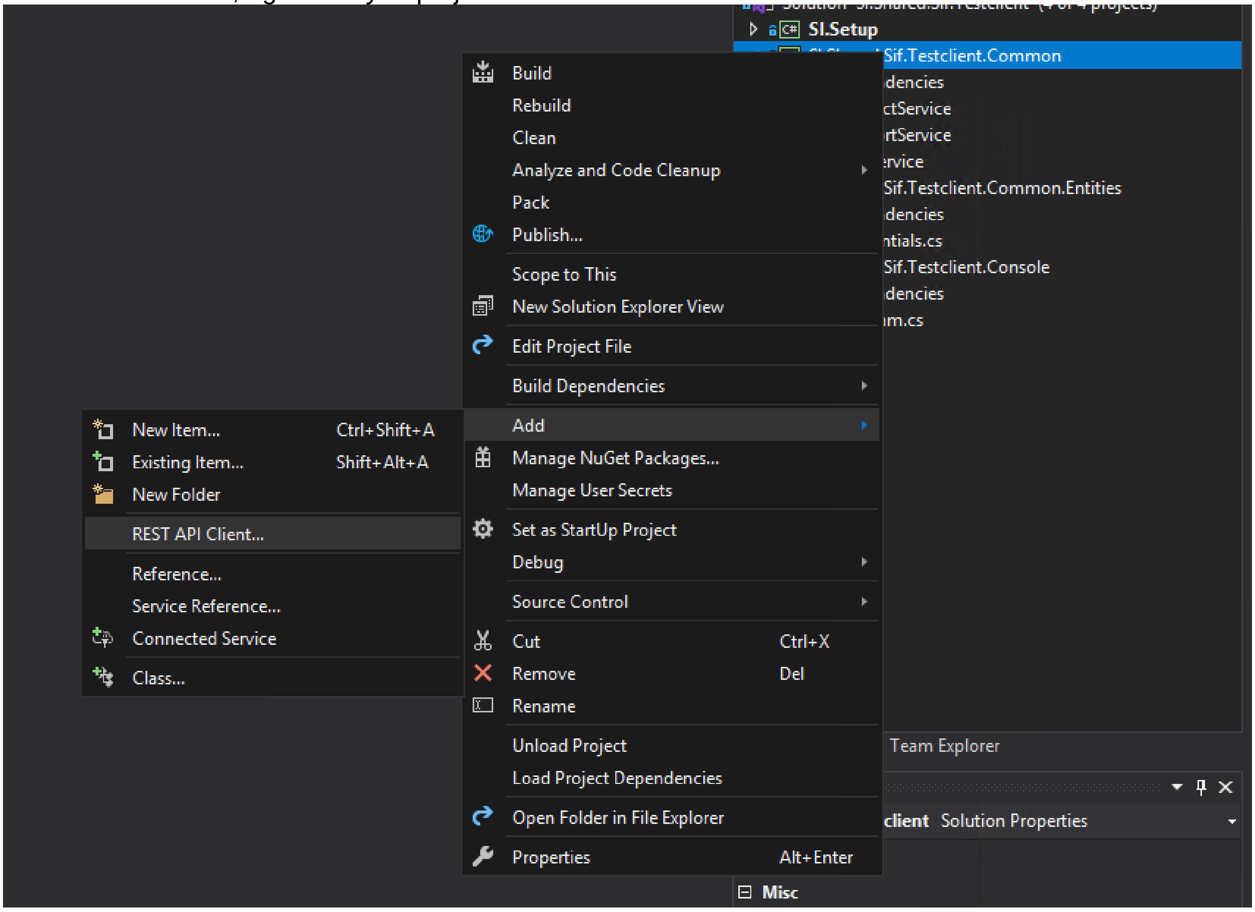
1 Guide

1. Go to swagger and find this link. It points to the API schema in Json format. Copy that URL.

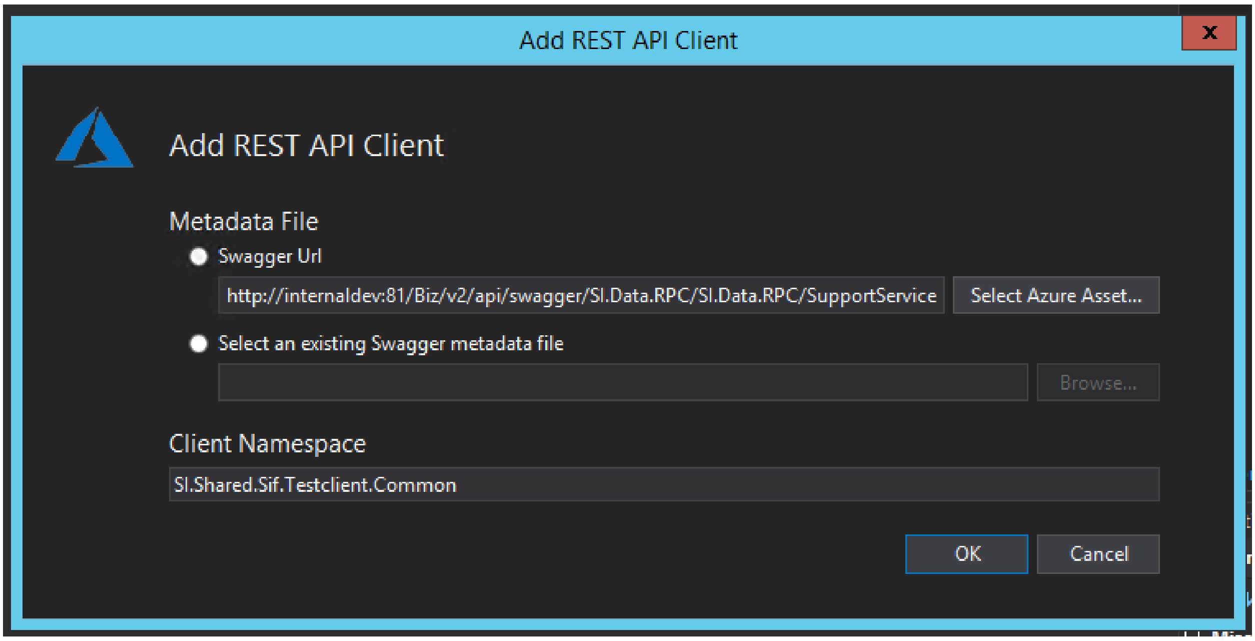


swagger
http://internaldev:81/Biz/v2/api/swagger/SI.Data.RPC/SI.Data.RPC/SupportService
SupportService
4.0.0
[Base URL: /Biz/v2/api/call/SI.Data.RPC/SI.Data.RPC/SupportService]
http://internaldev:81/Biz/v2/api/swagger/SI.Data.RPC/SI.Data.RPC/SupportService
SI.Data.RPC SupportService

2. Go to Visual Studio, right click you project and chose Add -> REST API Client



3. Past URL and click OK.

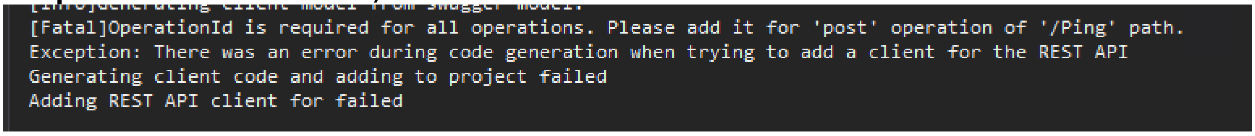


http://internaldev:81/Biz/v2/api/swagger/SI.Data.RPC/SI.Data.RPC/SupportService

SI.Shared.Sif.Testclient.Common

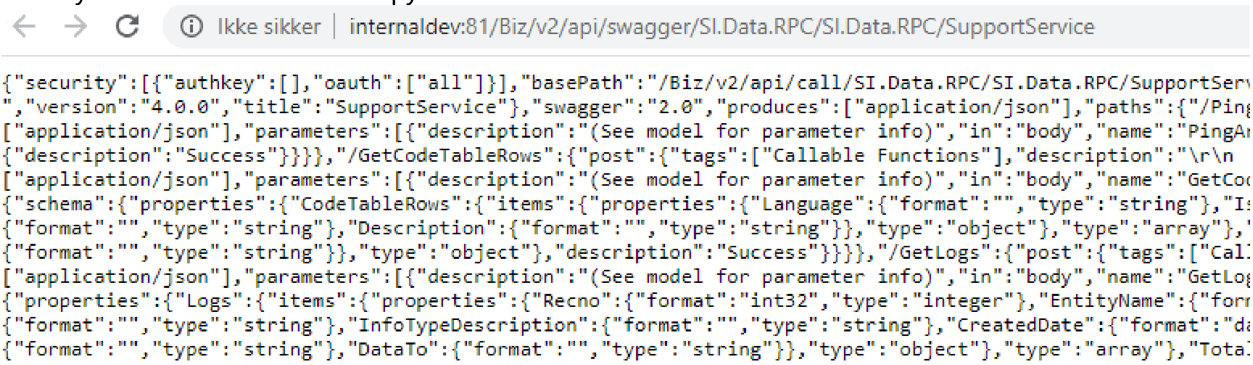
4. You should now se a folder in your project with the same name as your service. Se repo cie-shared-sif-testclient, for a sample on how to use the autogenerated code.

2 OperationID fatal error, walkaround



```
[Info]Generating client model from Swagger model.
[Fatal]OperationId is required for all operations. Please add it for 'post' operation of '/Ping' path.
Exception: There was an error during code generation when trying to add a client for the REST API
Generating client code and adding to project failed
Adding REST API client for failed
```

1. Go to your link in browser and copy the Json.

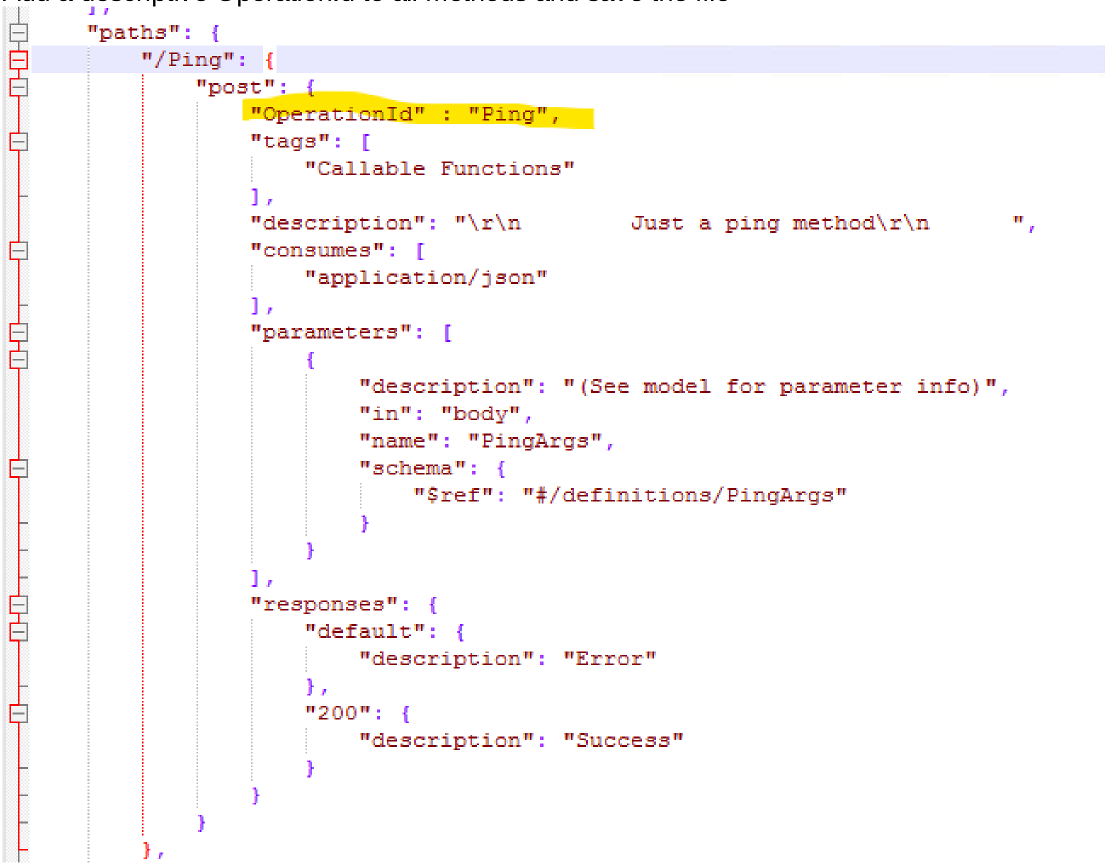


ⓘ Ikke sikker | internaldev:81/Biz/v2/api/swagger/SI.Data.RPC/SI.Data.RPC/SupportService

```
{"security":[{"authkey":[],"oauth":["all"]}],"basePath":"/Biz/v2/api/call/SI.Data.RPC/SI.Data.RPC/SupportSer
","version":"4.0.0","title":"SupportService"},"swagger":"2.0","produces":["application/json"],"paths":{"/Pin
["application/json"],"parameters":[{"description":"(See model for parameter info)","in":"body","name":"PingA
{"description":"Success"}}}},"/GetCodeTableRows":{"post":{"tags":["Callable Functions"],"description":"\r\n
["application/json"],"parameters":[{"description":"(See model for parameter info)","in":"body","name":"GetCo
{"schema":{"properties":{"CodeTableRows":{"items":{"properties":{"Language":{"format":"","type":"string"},"I
{"format":"","type":"string"},"Description":{"format":"","type":"string"}},"type":"object"},"type":"array"},
{"format":"","type":"string"}},"type":"object"},"description":"Success"}}}},"/GetLogs":{"post":{"tags":["Cal
["application/json"],"parameters":[{"description":"(See model for parameter info)","in":"body","name":"GetLo
{"properties":{"Logs":{"items":{"properties":{"Recno":{"format":"int32","type":"integer"},"EntityName":{"for
{"format":"","type":"string"},"InfoTypeDescription":{"format":"","type":"string"},"CreatedDate":{"format":"d
{"format":"","type":"string"},"DataTo":{"format":"","type":"string"}},"type":"object"},"type":"array"},"Tota
```

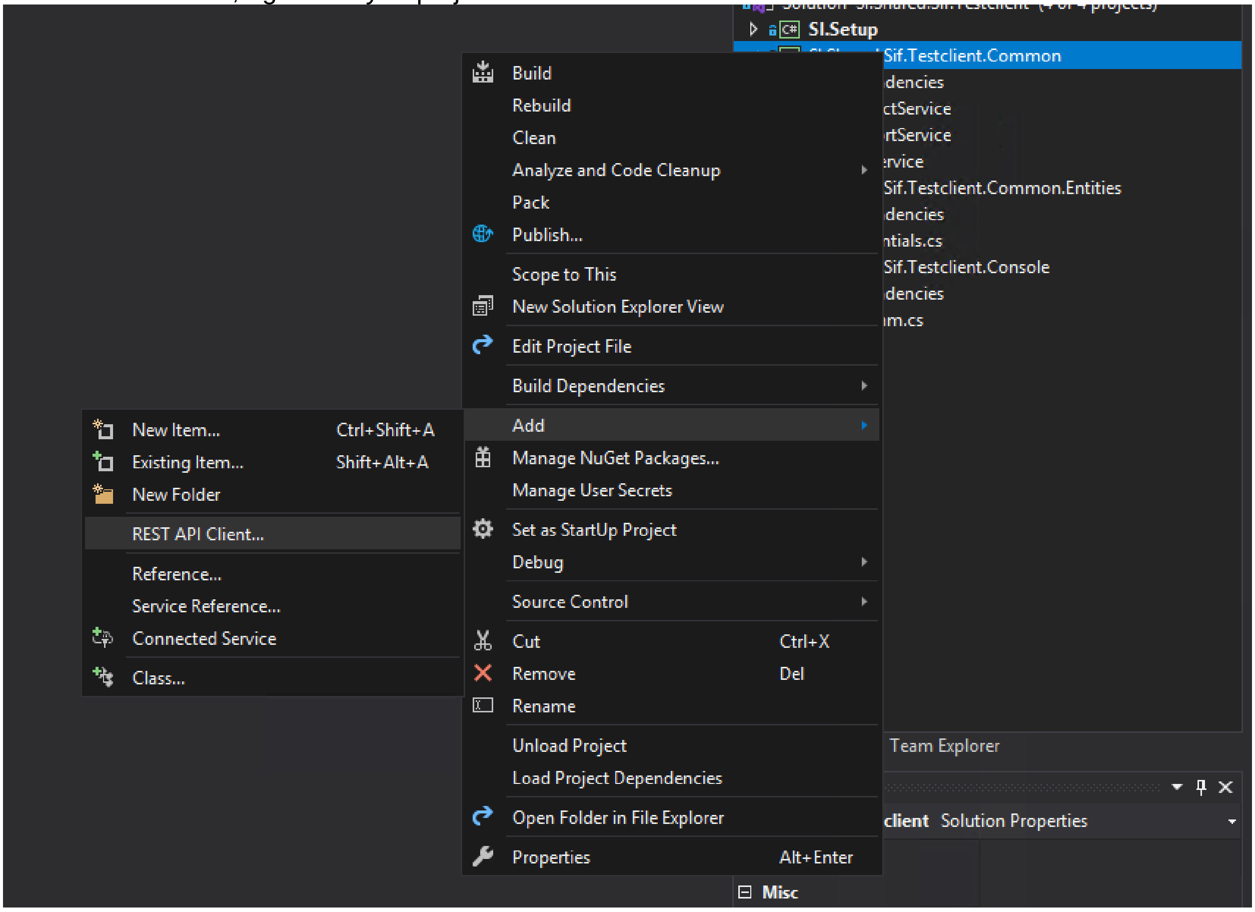
2. Past the code in an editor. Notepad++ with JSON Viewer plugin works great.

3. Add a descriptive OperationId to all methods and save the file

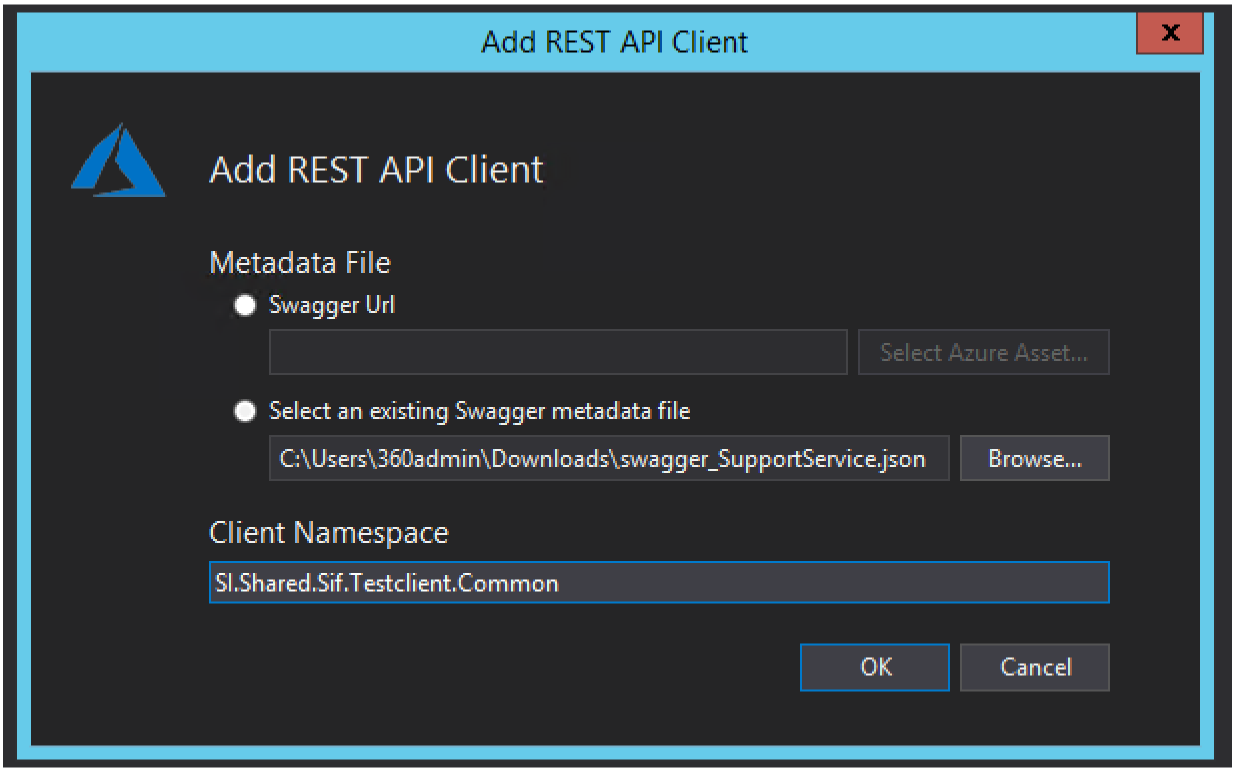


```
    ],
    "paths": {
        "/Ping": {
            "post": {
                "OperationId" : "Ping",
                "tags": [
                    "Callable Functions"
                ],
                "description": "\r\n        Just a ping method\r\n      ",
                "consumes": [
                    "application/json"
                ],
                "parameters": [
                    {
                        "description": "(See model for parameter info)",
                        "in": "body",
                        "name": "PingArgs",
                        "schema": {
                            "$ref": "#/definitions/PingArgs"
                        }
                    }
                ],
                "responses": {
                    "default": {
                        "description": "Error"
                    },
                    "200": {
                        "description": "Success"
                    }
                }
            }
        },
```

4. Go to Visual Studio, right click you project and chose Add -> REST API Client



5. Brows to the file and Click OK



Add REST API Client

Metadata File

○ Swagger Url

○ Select an existing Swagger metadata file

C:\Users\360admin\Downloads\swagger_SupportService.json

Client Namespace

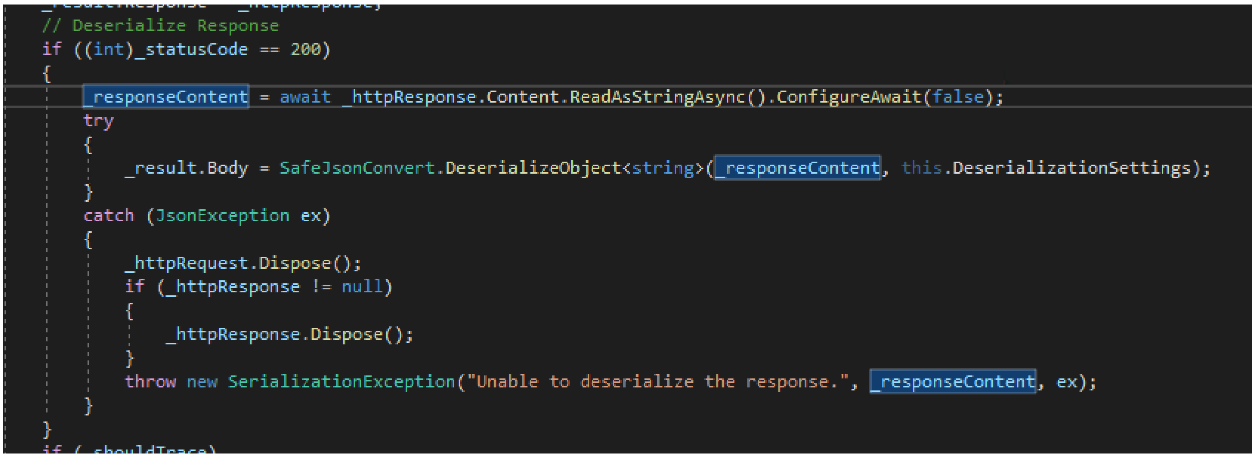
SI.Shared.Sif.Testclient.Common

6. You should now se a folder in your project with the same name as your service. Se repo https://github.com/Public360/SIF_RPC_TestClient, for a sample on how to use the autogenerated code.

3 How to skip Json deserialization

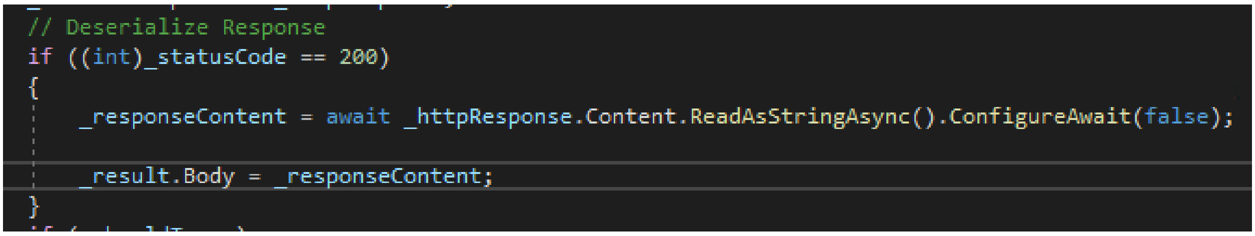
Find the code for your method and remove the serialization.

Before:



```
_result.Response = _httpResponse;
// Deserialize Response
if ((int)_statusCode == 200)
{
    _responseContent = await _httpResponse.Content.ReadAsStringAsync().ConfigureAwait(false);
    try
    {
        _result.Body = SafeJsonConvert.DeserializeObject<string>(_responseContent, this.DeserializationSettings);
    }
    catch (JsonException ex)
    {
        _httpRequest.Dispose();
        if (_httpResponse != null)
        {
            _httpResponse.Dispose();
        }
        throw new SerializationException("Unable to deserialize the response.", _responseContent, ex);
    }
}
if (_shouldTrace)
```

After:



```
// Deserialize Response
if ((int)_statusCode == 200)
{
    _responseContent = await _httpResponse.Content.ReadAsStringAsync().ConfigureAwait(false);

    _result.Body = _responseContent;
}
```