How to: use WCF service

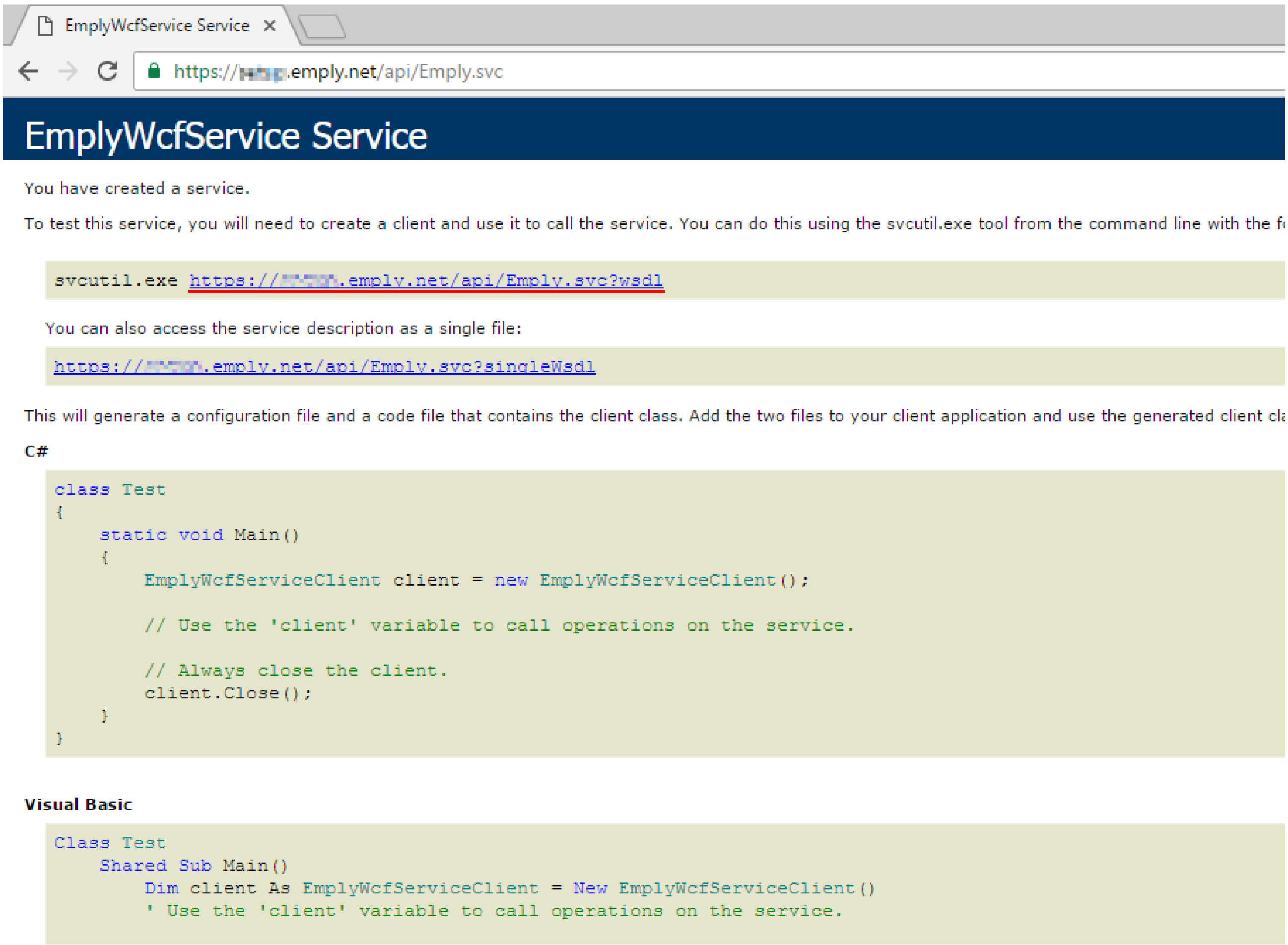
As additional feature Emply Hire system offers to its customers an ability to use WCF service. This can be helpful in case if there is a need to manipulate with department(s) or user(s) from custom .NET application.

In general WCF service allows to:

- Get departments *
- Add new department
- Modify existing department
- Remove existing department
- Get users *
- Get user
- Add new user
- Modify existing user
- Remove existing users
- Get candidate
- Get vacancy
- Get vacancies *
- Get applications from vacancy
- Get workflows

* *request can take a lot of time depending on amount of existing data*

Next example requires link to WSDL of WCF service. It can be found if open the service page **<site_address>/api/Emply.svc** in browser. For example if site address is https://customer.emply.net/ than full link to service page will be https://customer.emply.net/api/Emply.svc. On service page present developer information about service and direct link to service WSDL - for https://customer.emply.net/ it will be https://customer.emply.net/api/Emply.svc?wsdl:



EmplyWcfService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the f

```
svcutil.exe https://[redacted].emply.net/api/Emply.svc?wsdl
```

You can also access the service description as a single file:

```
https://[redacted].emply.net/api/Emply.svc?singleWsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client cla

C#

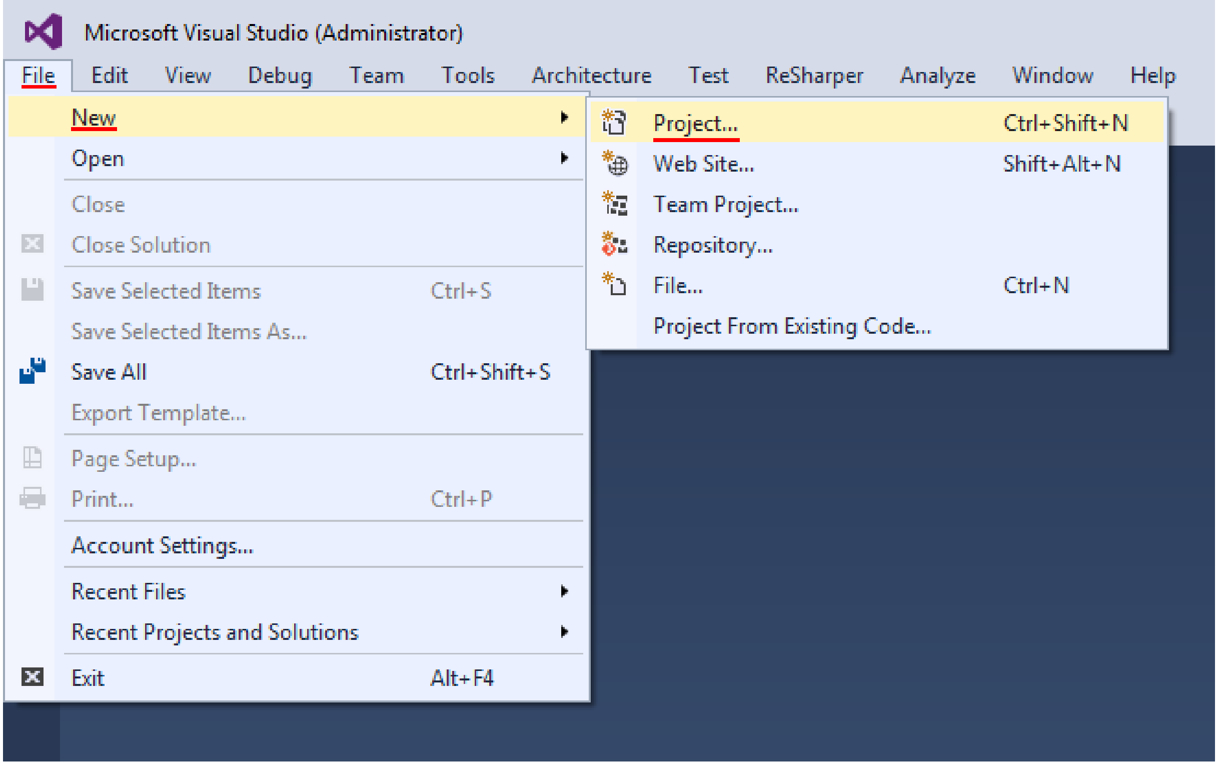
```
class Test
{
    static void Main()
    {
        EmplyWcfServiceClient client = new EmplyWcfServiceClient();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```

Visual Basic

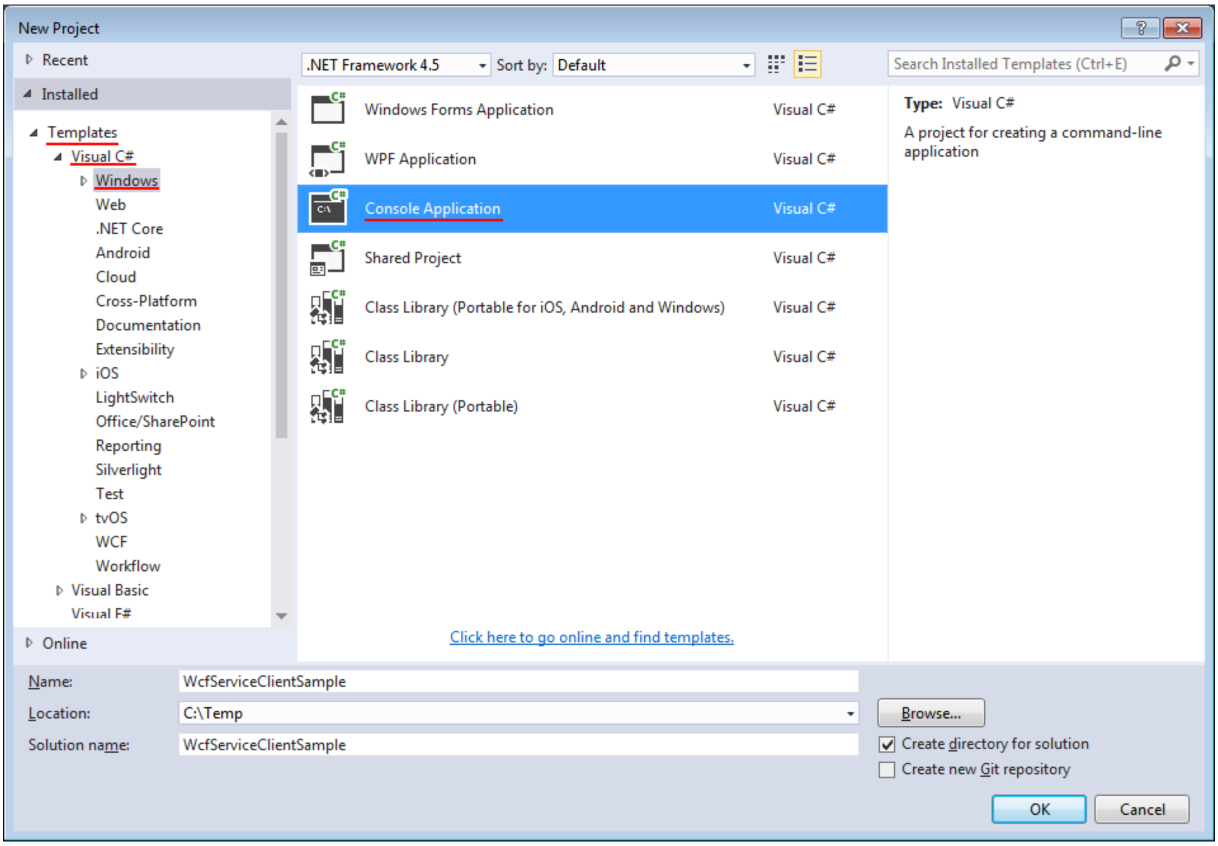
```
Class Test
    Shared Sub Main()
        Dim client As EmplyWcfServiceClient = New EmplyWcfServiceClient()
        ' Use the 'client' variable to call operations on the service.
```

Below is sample of how to configure simple console application for interaction with WCF service:

1. Open Visual Studio and create new project for console application **File** → **New** → **Project**:



2. In **New Project** window choose preferable language and platform among accessible templates - for example select **C#** language and **Console Application** under **Windows** templates section:



Type: Visual C#

A project for creating a command-line application

Name: WcfServiceClientSample

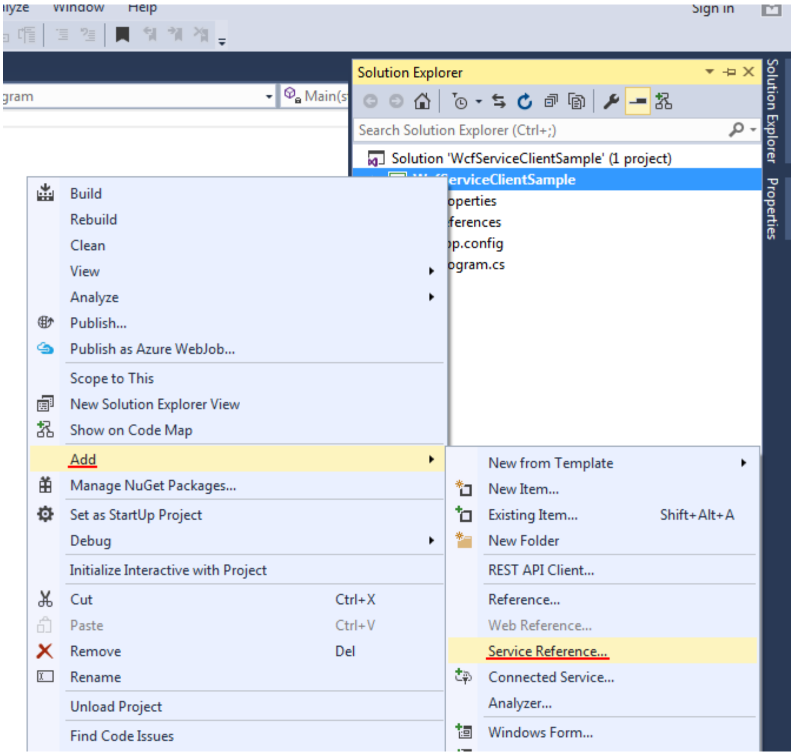
Location: C:\Temp

Solution name: WcfServiceClientSample

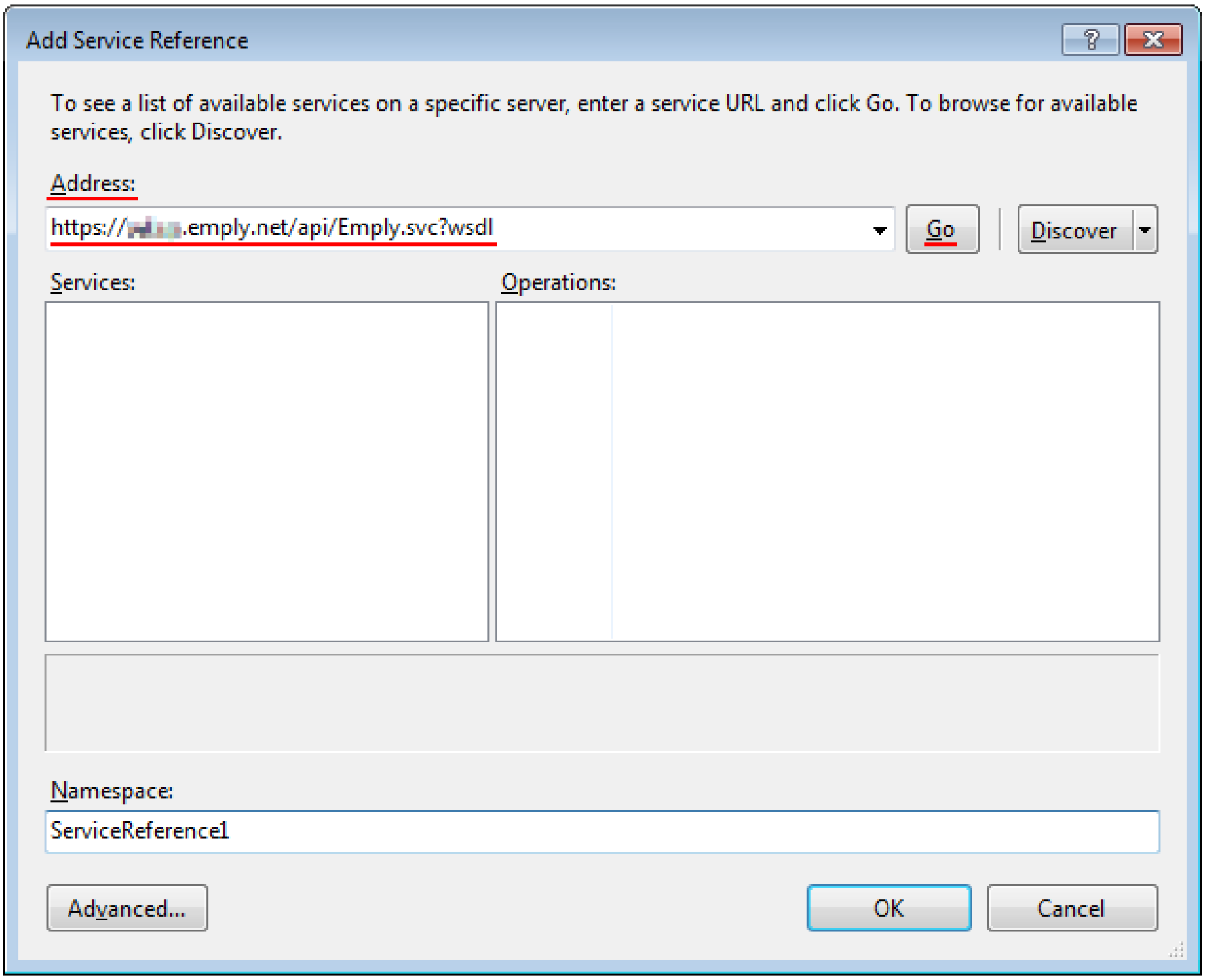
☑ Create directory for solution

☐ Create new Git repository

3. Expand **Solution Explorer**, right-click on project and select from menu **Add** → **Service reference…**:



4. In appeared **Add Service Reference** window paste copied link to WSDL of WCF service from service page in **Address** field and click **Go**:


Add Service Reference

To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.

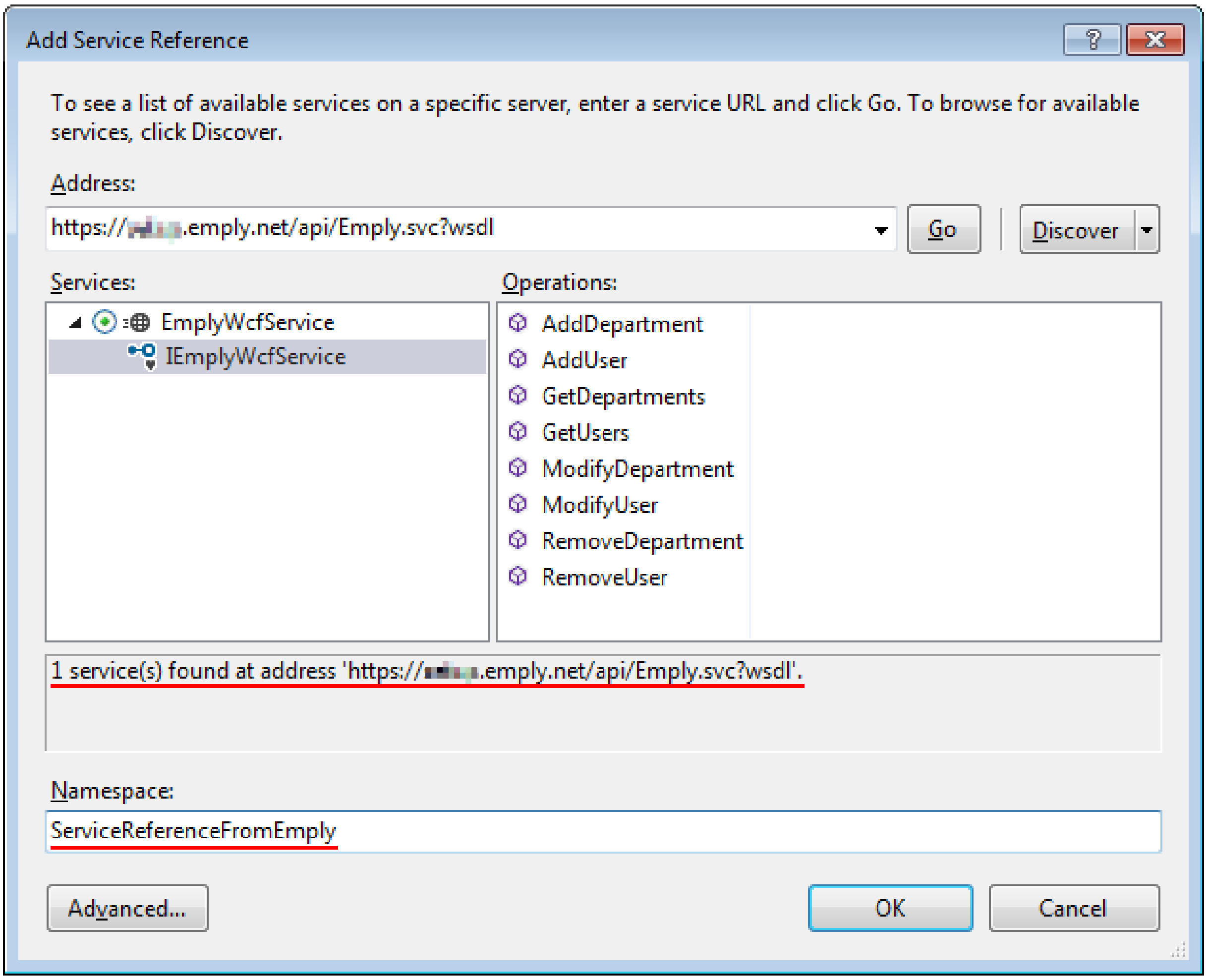
Address:

https://[redacted].emply.net/api/Emply.svc?wsdl

Namespace:

ServiceReference1

5. After previous step **Add Service Reference** window will be updated. **Namespace** field can be changed in a more meaningful value:



Add Service Reference

To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.

Address: https://[redacted].emply.net/api/Emply.svc?wsdl

Services:
- EmplyWcfService
  - IEmplyWcfService

Operations:
- AddDepartment
- AddUser
- GetDepartments
- GetUsers
- ModifyDepartment
- ModifyUser
- RemoveDepartment
- RemoveUser

1 service(s) found at address 'https://[redacted].emply.net/api/Emply.svc?wsdl'.

Namespace: ServiceReferenceFromEmply

Using of WCF service assuming that developer has knowledge about **API key**, provided to customer by Emply Hire system.

Next code sample shows hot to retrieve departments:

C#

```
using (EmplyWcfServiceClient client = new EmplyWcfServiceClient())
{
    DepartmentsRequest request = new DepartmentsRequest
    {
        ApiKey = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
    };
    DepartmentsResponse response = client.GetDepartments(request);

    foreach (DepartmentData department in response.Departments)
    {
        Console.WriteLine(department.Name);
    }
}
```

Also recommended to increase **maxBufferPoolSize**, **maxBufferSize** and **maxReceivedMessageSize** for service connection. For this purpose changes **app.config** like below:

```
<configuration>
  …
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
      <binding maxBufferPoolSize="2147483647"
            maxBufferSize="2147483647"
            maxReceivedMessageSize="2147483647"
  …
```

Detailed description for all accessible features of WCF service can be found in developer documentation which you can find in attachments to this article.