

WCF del 2: Service

I del 1 byggde ni en telefonbokapplikation och använde er då av en lokal lagringsklass för att spara och hantera kontakter. Uppgiften i denna del är att du ska bygga en tjänst som ska tillhandahålla samma funktionalitet som denna lagringsklass, tjänsten ska även lagra kontakter beständigt genom att lagra kontakterna i en XML fil. Tjänsten ska vara en SOAP baserad WCF Service. Du ska låta din webbapplikation använda sig av (vara klient till) denna tjänst för lagring och hantering av kontakter istället för att använda den statiska klassen Lagring.

Genom klienten ska man kunna använda all den funktionalitet som specificeras nedan för tjänsten vilket är nästan samma funktionalitet som i del 1.

Tjänsten (WCF Service)

Telefonboken ska bestå av ett antal lagrade kontakter samt ett antal funktioner. En kontakt har minst följande information:

- Förnamn
- Efternamn
- Telefonnummer

Tjänsten ska stödja följande funktionalitet:

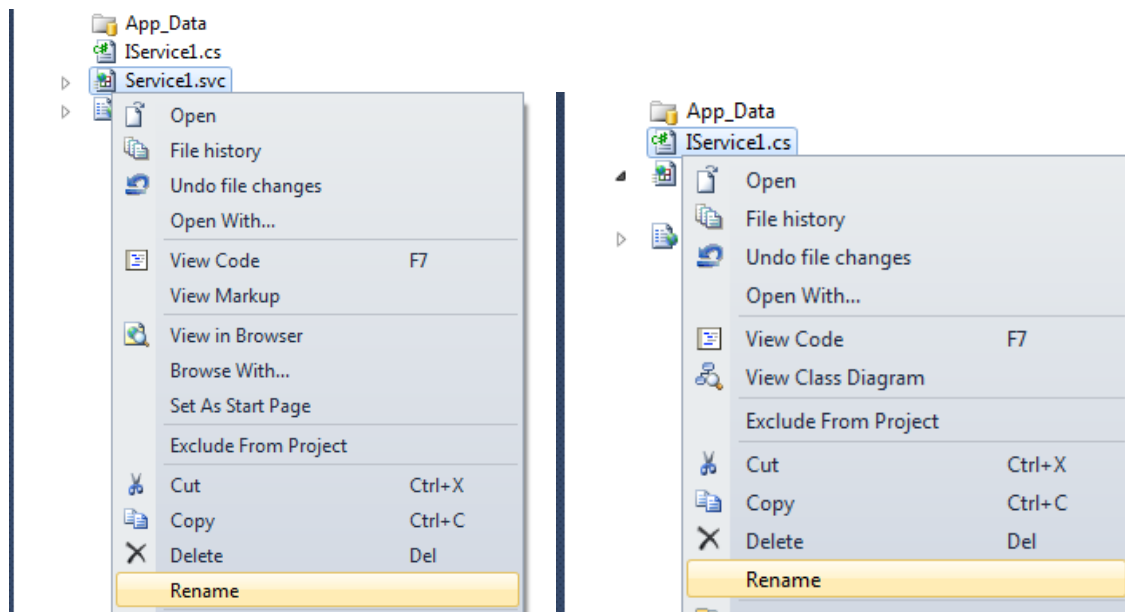
- Lägg till kontakt
- Editera en kontakt
- Ta bort en kontakt
- Sök efter kontakter på, förnamn, efternamn, telefonnummer

Tjänsten ska inte lagra kontakterna i minnet, då vi inte vill förlora alla kontakter när servern eller tjänsten går ned. Därför ska de lagras i XML-format hos tjänsten.

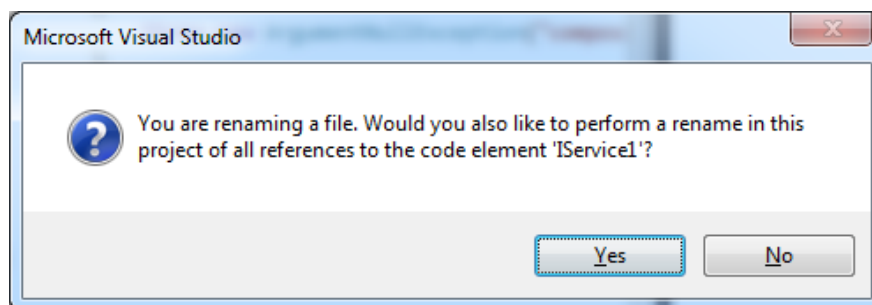
Komma igång

Starta först upp projektet som du började med i del 1. Den applikationen ska få agera klient till den service som du nu ska bygga. Nu ska du lägga till en WCF Service till samma solution. Detta gör du genom att trycka File > Add > New Project och välja WCF Service Application. Kom ihåg att ge dina projekt meningsfulla namn redan när du skapar dem.

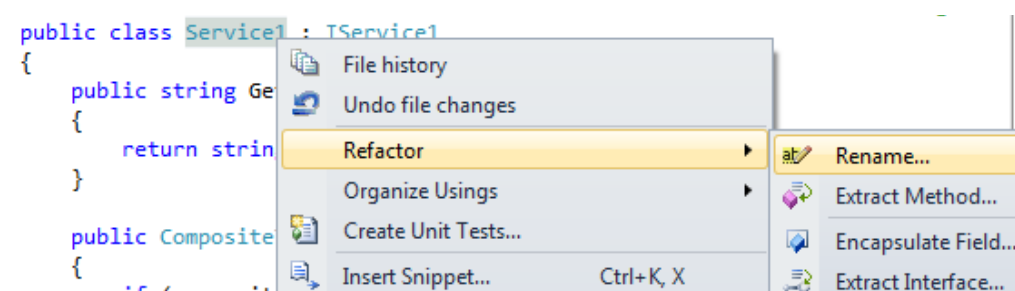
I ditt WCF Service projekt kommer du redan ha ett par filer, Service1 är tjänsten som kommer vara synlig från de klienter som använder tjänsten. Du ska alltså inte behålla det autogenererade namnet utan ge din tjänst ett meningsfullt namn. Gör detta för både IService1 samt IService.



När interfacet döps om frågar även Visual Studio om du vill uppdatera referenser i koden till det nya namnet, klicka på Yes så slipper du manuellt ändra även det.



Du behöver även ändra namnet i koden för klassen Service1 till samma namn som du gav till svc-filen. Högerklicka -> Refactor -> Rename.



Det finns även viss autogenererad kod i klasserna som du kan titta på och testköra om du vill men den behövs inte så du kan ta bort den för du ska göra din egen implementation.

Instruktioner

Interfacet med "ServiceContract" som attribut definierar vilka operationer som ska vara synliga utåt från denna service. MetodsSignaturerna deklareras som vanligt i interfacet men behöver även dekoreras med "OperationContract"-attributet.

Deklarera nu de metoder som tjänsten ska göra synliga genom att deklarerat följande metodsignaturer i interfacet: (för XElement behöver du using.XML.Linq)

```
[ServiceContract]
public interface IKontaktService {

    [OperationContract]
    XElement HämtaAllaKontakter();

    [OperationContract]
    void LäggTillKontakt(XElement kontakt);

    [OperationContract]
    void TaBortKontakt(int id);

    [OperationContract]
    void ÄndraKontakt(XElement ändradKontakt);

    [OperationContract]
    XElement SökKontakter(XElement kriterier);
}
```

SOAP och XML

Kommunikationen mellan klienten (MVC-applikationen) och tjänsten (WCF-servicen) görs med SOAP. SOAP möjliggör att två system kan kommunicera genom att komma överens om hur de ska anropa varandra, detta görs genom ett kontrakt. I vårt fall (som vi kan utläsa från det ovan definierade interfacet) så ingår klienten och tjänsten ett avtal att kommunicera mha typen XElement, dvs XML.

SOAP-meddelanden är definierade i XML samt att vi enligt vårt kontrakt (ServiceContract) kommer överens att skicka XML-objekt (XElement) som input och output till metoderna. Exempelvis för LäggTillKontakt så tar den metoden en XML som argument och själva anropet från klienten till tjänsten görs genom ett SOAP-meddelande som också är definierat i XML.

Lägg till service referensen

För att kommunicera mellan klienten och tjänsten behöver ni lägga till en så kallad "service reference". Hur ni går tillväga för att göra detta beskrivs i dokumentet "Introduktion_MVC_WCFServices".

Kontakt-klassen (klienten)

I klienten (MVC-applikationen) så används klassen Kontakt för att representera en kontakt med namn, telefonnummer osv. För att kommunicera med tjänsten används klasser som skapades i och med service referensen (ovan) men metoderna tar XML som input och output. Vi behöver alltså ett sätt att smidigt konvertera mellan ett Kontakt objekt och XML som representerar samma kontakt. Gör detta genom ta bort kommentarerna på metoderna ToXml och FromXml i Kontakt klassen:

```
public XElement ToXml()
public static Kontakt FromXml(XElement xml)
```

Implementera service-metoderna

Nu återstår att implementera metoderna i service-klassen (.svc-filen) i samband med att ni byter ut användningen av Lagring-klassen mot att ni istället nyttjar tjänsten.

Tips

Tips 1 Filväg och XML

För att få filvägen till App_Data-foldern hos tjänsten går det bra att använda

```
string appDataFolder = HostingEnvironment.MapPath("/App_Data/");
```

Antagligen behöver du lägga in ett using-statement också:

```
using System.Web.Hosting;
```

```
string appdata = HostingEnvironment.MapPath("/App_Data/");
XElement KontaktXml;
0 references | 0 authors | 0 changes
public KontaktService()
{
    KontaktXml = XElement.Load(appdata + "kontaktlista.xml");
}
```

Tips 2 Hantera datalagring mot XML

Det finns många sätt att göra detta, nedan finns några tips för ett sätt.

Lägg till en kontakt – Använd den XML som ges till metoden, ladda in hela XML-trädet, lägg till noden och spara sedan ned XML-trädet.

Editera en kontakt – Ladda in XML-trädet, leta upp rätt nod, byt ut dess innehåll mot det nya innehållet (ReplaceContent) och spara sedan ned XML-trädet.

Ta bort en kontakt – Ladda in XML-trädet, leta upp rätt nod, ta bort noden (Remove) och spara sedan ned XML-trädet.

Sök – Ladda in alla kontakter från XML-trädet och använd sedan LINQ för filtreringen.

Examination

Examination sker genom inlämning via mail till robert.karlsson@zocom.se. Du får svar när ditt projekt har testkörts om det är godkänt eller behövs kompletteras.