

Web Services in .Net

This exercise introduces you to the development of simple Web Services in .NET.

For this exercise we you will need

- Windows 2008 Server, Windows 2008 Server R2, Windows Vista or Windows 7 (IIS must be installed)
- Internet Information Server 7
- Visual Studio 2008 or Visual Studio 2010
- WSCF.blue from thinktecture

You can either install the software on your own computer (faster, more responsive) or use Windows 2008 R2 server `sandstorm.cs.ut.ee`. Windows operating system and Visual Studio can be acquired through MSDN AA download site (more information at <http://www.math.ut.ee/199208>). We will also use Microsoft SQL Server in future practices. WSCF.blue can be downloaded from <http://wscfblue.codeplex.com/>.

Part A – Creating a Web Service Client to a Remote Web Service

- Use Visual Studio to create a console application (C# or VB.NET)
 - Add a service reference (web reference in VS2005) to <http://terraservice.net/TerraService2.asmx?WSDL>
 - In your main method, create an instance of the `TerraServiceSoapClient` proxy class (when using Visual Studio 2005, the name of the proxy class is `TerraService` instead).
 - Create a `Place` object called `NY` and set its properties to “New York”, “New York”, “United States”.
 - Invoke the `GetPlaceFacts` method of the service to retrieve a `PlaceFacts` object about New York.
 - Write out the type of place that it is and the longitude and latitude its centre.
 - Compile and test your application.
- Invoke the `GetTileMetaFromLonLatPt` method of the service, passing the centre of New York, a theme of 1 and a scale of 8 metres.
 - If the tile metadata indicates that the tile exists, invoke the `GetTile` method of the service to retrieve the tile `Id` contained in the metadata.
 - Convert the byte array into a `System.Drawing.Image` using (you may have to add reference to `System.Drawing`):


```
Image.FromStream(new System.IO.MemoryStream(tile))
```
 - Invoke the `Image`'s `save` method to write the image to a file.
 - Compile and test your application.
- Try out some other scales (1m – 512m), themes (1 or 2) and cities (North America only).

Part B – Creating a Web Service (using WSDL first approach) – WCF version with WSCF.blue

- Use Visual Studio to create a new WCF Service Application called TemperatureConverter.
 - Delete IService1.cs and Service1.svc from the project.
 - Right click on the project in solution manager and choose WSCF.blue -> Choose WSDL to Implement....
 - Point to the ITempConverter.wsdl file you downloaded from the practice page.
 - Choose to create **Service-side stub**.
 - Choose to **adjust casing, format SOAP actions** and **generate separate files**.
 - Choose to **generate .svc file**.
 - Set Destination File Name as TempConverter.
 - Copy contents of system.serviceModel from output.config to web.config.
 - In web.config delete endpoint address.
 - Implement the CtoF and FtoC methods in ITempConverter.cs:
 - $F = C * 9/5 + 32$
 - $C = (F - 32) * 5/9$
 - Compile and “test” your web service.
 - Do not create a client application – just test locally via web browser and wcfstestclient (run from Visual Studio command prompt)

You can then deploy the Web service on the Sandstorm server by dropping the entire directory containing your Web site into C:\Inetpub\ESI\2011**YourLogin** on Sandstorm, and then check it by pointing your browser to:

<http://sandstorm.cs.ut.ee:PortNumber/2011/YourLogin/TempConverter.svc>

Optional – Creating a Web Service Client for you TempConverter service

- See Part A for hints.

Optional – Creating a Web Service (using WSDL first approach) – WCF version without WSCF.blue

- Open a Visual Studio command prompt
 - Use the svcutil tool to generate a client interface for the WSDL file located at [ITempConverter.wsdl](#)
(svcutil ITempConverter.wsdl /language:C#)

- Use Visual Studio to create a new WCF Service Application called `TemperatureConverter`.
 - Delete `IService1.cs` from the project.
 - Add the `ITempConverterService.cs` file (generated by `svcutil`) to the project.
 - Delete everything except the `ITempConverter` interface from `ITempConverterService.cs`.
 - Remove
 - `GeneratedCodeAttribute` from the interface,
 - `return:` attributes from the methods and
 - `ReplyAction` property of `OperationContractAttribute` from the methods.
 - Rename `Service1.svc` to `TempConverterService.svc`.
 - Right click on `web.config` and open WCF Service configuration dialog.
 - Change endpoint contracts to `ITempConverter` from `IService1`.
 - Modify the implementation of the Service:
 - Remove the methods in `TempConverterService`.
 - Change `IService1` to `ITempConverter` in the list of classes inherited by the service class.
 - Let Visual Studio implement the interface.
 - Implement the `CtoF` and `FtoC` methods:
 - $F = C * 9/5 + 32$
 - $C = (F - 32) * 5/9$
 - Compile and “test” your web service.
 - Do not create a client application – just test locally via web browser and `wcftestclient` (run from Visual Studio command prompt)

You can then deploy the Web service on the Sandstorm server by dropping the entire directory containing your Web site into `\\Inetpub\ESI\2011\YourLogin` on Sandstorm, and then check it by pointing your browser to:

<http://sandstorm.cs.ut.ee:PortNumber/YourLogin/TemperatureConverter.asmx>

Optional – Creating a Web Service (using WSDL first approach) – version for VS.Net 2005

- Open a Visual Studio command prompt
 - Use the `wsdl` tool to generate a server interface for the WSDL file located at [ITempConverter.wsdl](#)
- Use Visual Studio to create a new Web Site containing an ASP.NET Web Service called `TemperatureConverter`.

- Add the `ITempConverterserviceInterfaces.cs` file to the `App_Code` folder of the Visual Studio project
- Modify the implementation of the Service:
 - Remove the constructor and `HelloWorld` method
 - Remove the attributes from the class.
 - Change the base class to `IITempConverterbinding`
 - Create `CtoF` and `FtoC` methods as defined in the interface.
- Implement the `CtoF` and `FtoC` methods:
 - $F = C * 9/5 + 32$
 - $C = (F - 32) * 5/9$
- Compile and “test” your web service.
 - Do not create a client application – just test locally via web browser

You can then deploy the Web service on the Sandstorm server by dropping the entire directory containing your Web site into `\\Inetpub\ESI\2011\YourLogin` on Sandstorm, and then check it by pointing your browser to:

<http://sandstorm.cs.ut.ee:PortNumber/2011/YourLogin/TemperatureConverter.asmx>

The “login” is your normal university login. The port number will be disclosed by the lab assistant during the tutorial.