

```
1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Linq;
5 using System.Runtime.Serialization;
6 using System.ServiceModel;
7 using System.ServiceModel.Web;
8 using System.Text;
9
10 // REMARQUE : vous pouvez utiliser la commande Renommer du menu
    Refactoriser pour changer le nom d'interface "IService" à la fois dans
    le code et le fichier de configuration.
11
12 // Client Test pour WCF : %ProgramFiles%\Microsoft SDKs\Windows\v----
    \Bin\SvcUtil.exe
13
14
15 [ServiceContract]
16 public interface IService
17 {
18
19     [OperationContract]
20     string GetData(int value);
21
22     [OperationContract]
23     string Meteo_GetTemperature();
24
25     [OperationContract]
26     string Meteo_GetTemperatureByDate(string sDate, out string
        sDateReleve);
27
28     [OperationContract]
29     DataSet Meteo_GetTemperatureByDay(string sDate);
30     //[OperationContract]
31     //CompositeType GetDataUsingDataContract(CompositeType composite);
32
33     // TODO: ajoutez vos opérations de service ici
34 }
35
36 // Utilisez un contrat de données comme indiqué dans l'exemple ci-après
    pour ajouter les types composites aux opérations de service.
37 //[DataContract]
38 //public class CompositeType
39 //{
40 //    bool boolValue = true;
41 //    string stringValue = "Hello ";
42
43 //    [DataMember]
44 //    public bool BoolValue
45 //    {
46 //        get { return boolValue; }
47 //        set { boolValue = value; }
48 //    }
```

```
49
50 // [DataMember]
51 // public string StringValue
52 // {
53 //     get { return stringValue; }
54 //     set { stringValue = value; }
55 // }
56 //}
57
```