

# ASP.NET Web Services

## Beispiel 1: Fragen

1. Ein Web Service benötigt als Input Parameter ein Integer-Array. Welche der drei Protokolle HTTP-GET, HTTP-POST und SOAP eignen sich für die Übergabe eines Arrays?
2. Warum benötigt ein Web Service Client einen Proxy um mit einem Web Service zu kommunizieren?
3. Was ist WSDL und wofür wird es benötigt?

## Beispiel 2: Webservice mit SOAP

Schreiben Sie ein **Webservice** in einer Datei "encodedecode.asmx". Speichern Sie die Datei in einem Verzeichnis, auf das Ihr IIS Zugriff hat. Schreiben Sie ein Interface "IEncoder", welches eine Methode enthält, um einen String zu kodieren. Schreiben Sie ein Interface "IDecoder", welches eine Methode enthält, um einen kodierten String wieder zu dekodieren.

Schreiben Sie eine Klasse "**Reverser**", welche beide Interfaces implementiert (verwenden Sie einen entsprechenden einfachen Kodieralgorithmus, beispielsweise Buchstaben im String vertauschen). Leiten Sie die Klassen von System.MarshalByRefObject ab.

- a) Überprüfen Sie die WSDL, indem Sie Ihr Webservice ohne Parameter in einem Browser aufrufen.
- b) Überprüfen Sie die WSDL, indem Sie Ihr Webservice in einem Browser mit dem Parameter "?WSDL" ausführen.
- c) Generieren Sie einen Proxy mit "WSDL.EXE" und schreiben Sie einen einfachen Client der die Funktionalität des Webservices (über diese generierte Proxyklasse) verwendet.

## Beispiel 3: Web Services

Entwickeln Sie ein **Web Form** Horoskop.aspx mit dem ein Sternzeichen ausgewählt und mit einem Click auf den „Horoskop berechnen“ Button das zugehörige Horoskop angezeigt wird.

Die Berechnung des Horoskops erfolgt durch den **Web Service** Horoskop.asmx, der als Input einen String mit dem Namen des ausgewählten Sternzeichens erhält. Der Inputstring dient als Schlüssel zur Suche des Rückgabestrings aus der „**Horoskopdatenbank**“ Horoskop.xml. Diese ist als XML-Datei in folgendem Format zu erstellen (Beispiel):

```
<?xml version="1.0" ?>
<quotes>
  <Steinbock>
    Ihr Horoskop besagt, dass Sie zwischen 22. Dezember
    und 20. Jänner geboren sind.
  </Steinbock>
  <Wassermann>
    Ihr Horoskop besagt, dass Sie zwischen 21. Jänner
    und 18. Februar geboren sind.
  </Wassermann>
  <Fisch>
    ...
  </Fisch>
  ...
</quotes>
```

Erstellen Sie eine vollständige XML-Horoskop-Datei entsprechend dem Beispiel und verwenden sie die entsprechenden XML-Zugriffsklassen aus dem .NET Framework um effizient auf die Datei zuzugreifen.

Verwenden Sie für das Webservice die Outline Technik (d.h. Code in separatem File).

# ADO.NET

## Beispiel 4: ADO.NET

In diesem Beispiel soll der Managed Provider für SQL Server (System.Data.SQL) verwendet werden, um auf eine Datenbank zuzugreifen. Zur Erstellung der Test-Datenbank und der Stored Procedures sind folgende Schritte durchzuführen (Verwendung von ISQLW.EXE oder ähnliches DBMS Tools – Hinweise unten beachten!):

```
use master
go
create database demo
go
use demo
go
create table instructors (name varchar(32) primary key, age int, homeoffice
varchar(32))
insert instructors values('Don Smith', 26, 'London')
insert instructors values('Ted Pattison', 26, 'Paris')
insert instructors values('Tim Ewald', 22, 'Boston')
insert instructors values('John Miller', 25, 'Vienna')
insert instructors values('Steve Rodgers', 19, 'London')
select * from instructors
go
create procedure AgeInstructors
    @office varchar(32),
    @amount int,
    @totalAge int output
as
    update instructors set age = age + @amount where homeoffice = @office
    select @totalAge = sum(age) from instructors where homeoffice = @office
    select name from instructors where homeoffice = @office
```

Das Programm soll folgende Funktionalität aufweisen:

- Durchführung einer Abfrage (z.B. `select * from instructors`) und das Resultat ausgeben.
- Durchführung eines parametrisierten Abfrage (der Parameter wird als Command Line Argument übernommen): `select * from instructors where homeoffice = @home`
- Aufruf der Stored Procedure `AgeInstructors` (die beiden Parameter werden als Command Line Argumente übernommen). Das durch die Stored Procedure zurückgelieferte Resultat soll ausgegeben werden.

Beispiel für Aufruf:

```
adodemo.exe Boston 5
```

### Hinweise:

Falls nicht der SQL Server zur Verfügung steht, kann auch die MSDE verwendet werden. Sie muß allerdings zuerst fertig installiert werden:

C:\Program Files\Microsoft.NET\Framework SDK\Samples\StartSamples.htm

Verbindung zur MSDE herstellen (Beispiel; der User sa hat standardmäßig kein gesetztes Passwort):

```
osql -S MachineName\NETSDK -U sa
```