


XML-Webservices

Frank Ruthardt




Agenda

1. Einführung
 2. XML
 3. Webservice-Technik
 4. Microsoft .NET
 5. Anwendungsbeispiel
- 

Agenda

1. Einführung
 2. XML
 3. Webservice-Technik
 4. Microsoft .NET
 5. Anwendungsbeispiel
- 

Motivation und Einführung

- ▶ Serviceorientierte Architektur (SOA)
 - ▶ Enterprise Application Integration (EAI)
 - ▶ Plattformübergreifende Kommunikation
 - ▶ Offene Standards
 - ▶ Geringes Datenaufkommen
- 

Agenda

1. Einführung
2. **XML**
3. Webservice-Technik
4. Microsoft .NET
5. Anwendungsbeispiel

XML

```
<Spiel Startdatum="01.12.2007">  
  <Spieler Id="1">Müller</Spieler>  
  <Spieler Id="2">Mayer</Spieler>  
</Spiel>
```

Dokumenttypdefinition (DTD)


- ▶ XML-Dokumente syntaktisch einzugrenzen
- ▶ Einfache syntaktische Einschränkungen
- ▶ Nur 2 Datentypen
 - CDATA
 - PCDATA
- ▶ Ermöglicht Schemavalidierung

DTD

```
<!DOCTYPE Spiel[
  <!ELEMENT Spiel (Spieler+)>
  <!ATTLIST Spiel StartZeit CDATA #REQUIRED>
  <!ELEMENT Spieler (#PCDATA)>
  <!ATTLIST Spieler Vorname CDATA #REQUIRED>
  <!ATTLIST Spieler Name CDATA #REQUIRED>
]>

<Spiel StartZeit="01.12.2007">
  <Spieler Vorname="Hans" Name="Mayer" />
  <Spieler Vorname="Martin" Name="Müller" />
</Spiel>
```

XML Schema (XSD)

- ▶ Viele Datentypen
 - String, Boolean, Date usw.
 - ▶ Genauere syntaktische Angabe
 - MinOccurs
 - MaxOccurs
 - ▶ Referenzen
- 

XSD

```
<xs:schema elementFormDefault="qualified" targetNamespac
  <!-- Root-Element -->
  <xs:element name="Spiel" type="SpielType" />
  <!-- Type-Definitions -->
  <xs:complexType name="SpielerType">
    <xs:attribute name="Vorname" type="xs:string" />
    <xs:attribute name="Name" type="xs:string" />
  </xs:complexType>
  <xs:complexType name="SpielType">
    <xs:sequence minOccurs="1" maxOccurs="unbounded">
      <xs:element type="SpielerType" name="Spieler" />
    </xs:sequence>
    <xs:attribute name="StartZeit" type="xs:dateTime" />
  </xs:complexType>
</xs:schema>
```

XML–Namensräume

- ▶ Namensraum–Prefix
 - `<raum:Elementname />`
- ▶ Definition des Schemas in beliebigem Element
 - `<game:Spiel xmlns:game="XMLSchemaEx.xsd" />`
- ▶ Vorteile:
 - Klare Strukturierung
 - Mehrere XSDs

XML-Serialisierung

- ▶ Objekte werden in sequenzielle Darstellung serialisiert
- ▶ Können beliebig wieder deserialisiert werden.
- ▶ Objekte werden dadurch immer kopiert!

Agenda

1. Einführung
2. XML
3. **Webservice-Technik**
4. Microsoft .NET
5. Anwendungsbeispiel

Simple Object Access Protokoll

- ▶ Einheitliches Interface zum Übertragen von Anfragen
- ▶ Anfragen und Antworten durch XML-Dokumente
- ▶ Nachrichtenstruktur definiert durch ein XSD
- ▶ Kann über verschiedene Protokolle versendet werden
 - Gängig: „SOAP over HTTP“ oder „SOAP over TCP“

Webservice Definition Language

- ▶ Manifest des Webservice
- ▶ Enthält
 - URL
 - Protokolle
 - Funktionen
 - SOAP-Nachrichtentypen
- ▶ Dient zum Erstellen der Proxy-Objekte
- ▶ Plattformunabhängig
- ▶ Definiert durch XSD

Proxy-Objekte

- ▶ Können automatisch aus WSDL generiert werden
- ▶ Enthalten allen Kommunikationscode
- ▶ Transparent für den Programmierer

- ▶ Programmierer spricht nur diese Proxy-Objekte an.

Universal Description, Discovery, and Integration (UDDI)

- ▶ Nur für öffentliche Webservices sinnvoll
 - Börsenkurse
 - Wetter
- ▶ Dient zum Auffinden aktueller Dienste
- ▶ Einheitliche DB für vorhandene WS
- ▶ Definiert durch XSD

Agenda

1. Einführung
2. XML
3. Webservice-Technik
4. **Microsoft .NET**
5. Anwendungsbeispiel

XML-Dateien im Visual Studio

- ▶ Bearbeiten von XML-Dateien und Schemata

Implementierung in Microsoft .NET

- ▶ Erstellen eines Webservices
- ▶ Basiert auf ASP.net
- ▶ Es wird XML statt HTML geliefert

Webserver in .NET

- ▶ Microsoft IIS
 - Bestandteil von Windows (Pro-Versionen)
 - Beliebiger Pfad wird dauerhaft per http erreichbar
- ▶ Integrierter Webserver von VS
 - Bestandteil des VS
 - Beliebiger Pfad wird temporär per http erreichbar

Debuggen eines Webservice

- ▶ Kann Lokal wie jede Anwendung debuggt werden
- ▶ Remote-Debuggen möglich

Deployment

- ▶ X-Copy
- ▶ MSI

Agenda

1. Einführung
 2. XML
 3. Webservice-Technik
 4. Microsoft .NET
 5. **Anwendungsbeispiel**
- 

Anwendungsbeispiel

- ▶ Integrierter Webserver
- ▶ WSDL
- ▶ SOAP-Nachrichten

- ▶ Test-Application

Einsatzszenario

- ▶ Beliebige .NET-Anwendung
 - z.B. ClickOnce-Deployment oder XBAP
- ▶ Keine lokalen Zugriffe
- ▶ Alle Daten sind servergespeichert
 - (auch Fenstergrößen oder ähnliches)
- ▶ Anwendung kann damit auf dem lokalen PC in Partial Trust Umgebung ausgeführt werden. (Sandbox)

Fragen?!

