

Web Services

Ingo Köster

Diplom Informatiker (FH)

Motivation

- › Viele Anwendungen, gerade Unternehmensanwendungen, sind datengetriebene Anwendungen
- › D.h. neben einer Geschäftslogik spielen die Unternehmensdaten eine große Rolle
- › Diese Daten sind oft in einer Datenbank gespeichert
 - › Andere Quellen wie JSON- oder XML-Dateien sind ebenfalls möglich

Weitere Daten

- › Neben den Unternehmensdaten wie Kunden, Produkten, Lieferanten, etc. gibt es weitere Daten, welche für eine Anwendung relevant sein können
- › Für den Versand von Produkten oder Rechnungen, Bestellung von Rohstoffen, etc. sind z.B. Standortdaten sehr wichtig
 - › Z.B. PLZ zur Straße
- › Da sich Straßennamen und Postleitzahlen nur selten ändern, könnten diese ebenfalls mit den Unternehmensdaten gespeichert werden

Dynamische Daten

- › Es gibt Daten, welche nicht sehr sinnvoll in Dateien oder Datenbanken zwischengespeichert werden können oder sollten
- › Gewisse Daten ändern sich sehr oft
- › Aktuelle Daten sind für die weitere Verarbeitung oft notwendig
- › Beispiele
 - › Wechselkurse von Währungen
 - › Wetterdaten
 - › Fahrzeiten von Zügen oder dem ÖPNV



Dynamische Daten aus dem Internet

- › Bedeutet, dass eher dynamische Daten in dem Moment abgerufen werden sollten, in dem sie benötigt werden
- › Viele dieser Daten könnten z.B. von Web-Seiten bezogen werden, müssten dann aber mühselig aus den HTML-Daten geparkt werden



`<div data-name="US-Dollar">US-Dollar entspricht</div><div class="dDoNo ikb4Bb gsert gzfeS">0,85 <div class="tF2Cxc"><div class="yuRUbf">https://www.xe.com/de/currencyconverter/convert/?Amount=1&From=USD&To=EUR</div></div></div>`

Services

- › Um den Abruf dieser Daten zu standardisieren, werden Dienste (Services) definiert
- › Ein Service oder auch Web Service liefert bei einer Anfrage (z.B. per HTTP) die gewünschten Informationen als JSON oder XML
- › Beispiel
 - › In einem HTTP Request wird Quellwährung, Zielwährung und der Betrag übermittelt
 - › Der Service empfängt diese Daten und ermittelt anhand der ihm vorliegenden Wechselkurse den Betrag in der Fremdwährung
 - › Der Service verpackt die Antwortdaten in JSON und sendet diese per HTTP Response an den Client zurück

Services

- › Sind Schnittstellen, über die zwei Maschinen (oder Anwendungen) miteinander kommunizieren können
 - › Maschine zu Maschine Kommunikation
- › Sind plattformunabhängig, d.h. Client und Server müssen nicht die gleiche Hardware und Software verwenden
 - › Client in .NET unter Windows kann mit Service in Java unter Linux kommunizieren
- › Über das Internet greifen unterschiedliche Clients auf Dienste zu

Services - Protokolle

- › Web Services verwenden oft das HTTP Protokoll für Anfragen und Antworten
 - › Request und Response
- › Web Services antworten auf eine Anfrage (Request) in einem definierten Format wie z.B. JSON oder XML
- › Diese Daten können innerhalb einer Anwendung einfach deserialisiert werden

Services



Webservice Client mit C#

Klasse für das HTTP-Protokoll

- › Um aus einer .NET Anwendung einen HTTP-Request an einen Service zu senden, wird die Klasse `HttpClient` verwendet

```
using System.Net.Http;
```

```
...
```

```
HttpClient client = new HttpClient();
```

Request an den Service senden

- › Mittels der Methode GetStringAsync wird der URL für den Request übergeben
- › Der Rückgabewert ist ein String mit den Antwortdaten des Servers
- › Diese enthalten nur den Nachrichtinhalt (z.B. JSON), nicht die gesamte HTTP-Response

```
string message = await  
client.GetStringAsync("https://api.predic8.de/shop/products/  
62");
```

Request konfigurieren

- › Um festzulegen ob die Antwortdaten im JSON- oder XML-Format vorliegen sollen, können oft Header gesetzt werden

```
HttpClient client ...  
client.DefaultRequestHeaders.Accept.Add(  
    new MediaTypeWithQualityHeaderValue("application/json"));
```

- › Außerdem ist es möglich dem Service z.B. Informationen zum Client zuzusenden

```
client.DefaultRequestHeaders.Add("User-Agent", "Mein C# Programm");
```

System.AggregateException: "One or more errors occurred. (The remote server returned an error: (403) Forbidden. Please comply with the User-Agent policy: https://meta.wikimedia.org/wiki/User-Agent_policy.)"

Antwort deserialisieren

- › Der String kann anschließend in ein Objekt deserialisiert werden
 - › Oder in ein Array oder eine Collection

```
string message = await client.GetStringAsync("https://...");
```

```
Article article =  
JsonSerializer.Deserialize<Article>(message);
```

Konfigurieren der Deserialisierung

- › Bezeichner in den JSON Dokumenten eines Service werden oft komplett klein geschrieben
 - › Es können auch Bindestriche oder Leerzeichen vorkommen
- › Um Properties in gewohnter C# Schreibweise verwenden zu können, werden diese mittels eines Attributes gemappt

- › Beispiel

```
[JsonPropertyName("name")]  
public string Name { get; set; }
```

```
{  
  "name": "Mango fresh",  
  "price": 5.55,  
  "photo_url": "/shop/products/62/photo",  
  "category_url": "/shop/categories/Fresh",  
  "vendor_url": "/shop/vendors/672"  
}
```

Antwort deserialisieren - Alternative

- › Die Klasse `HttpClient` hat eine Methode, welche JSON herunterlädt und in ein Objekt deserialisiert
- ›

```
Article article = await  
client.GetFromJsonAsync<Article>("https://...");
```
- › Diese Methode ignoriert die Groß und Kleinschreibung beim Deserialisieren