

WCF Services mal anders

22.03.2011

Melanie Eibl

E-Mail: melanie@eibl-winterscheid.de

Website: <http://www.eibl-winterscheid.de>

Hostingmöglichkeiten

- Konsolenanwendung
- Windows-Forms-Anwendung
- Windows-Dienst
- COM+
- IIS/ASP.NET
- Windows ActivationService (WAS)

Self-Hosting

- Der Host-Prozess wird im Code gemanaged
- Instanz eines *ServiceHost*-Objekts erzeugen und Öffnen
- *ServiceHost* implementiert den Host von WCF-Services
- Listener, der Messages empfängt
- Endpunkte definieren (imperativ oder deklarativ in Konfigurationsdatei wie *App.config*)

Endpunkte

- Endpunkte ermöglichen Clients den Zugriff auf die Funktionalität, die ein WCF-Dienst anbietet
- Nachrichten werden zwischen Client-und Server-Endpunkten ausgetauscht
- Endpunkt-ABC: Address–Binding – Contract(Mindestbestandteile eines Endpunktes)
- Ein Service kann über mehrere Endpunkte erreichbar sein

HTTP

- Auf Request-/Response-Paradigma basierendes Protokoll
- Im OSI-Referenzmodell der Anwendungsschicht (7) zugeordnet (oberhalb TCP/IP)
- Neben den Information werden Metadaten übertragen

HTTP Anforderungsmethoden

- GET
- PUT
- POST
- DELETE
- HEAD
- Außerdem gibt es noch: PATCH, COPY, MOVE, LINK, UNLINK, TRACE, WRAPPED

SOAP-basierte Services

- Selbstbeschreibend durch WSDL
- Service Referenz mit Proxy und Entities
- Kein net.Tcp möglich
- Aufgebläht durch Envelope
- Nachricht wird über HTTP mit POST versendet

RESTful Services

- HTTP basiert
- Außer POST werden noch andere Anforderungsmethoden wie GET, PUT und DELETE genutzt
- Daten werden in XML oder JSON übertragen
- Keine Service Referenz
- Google, Amazon, Digg, Flickr, Twitter

Konsumieren eines RESTful Services

- *WebClient* und *HttpWebRequest* aus *System.Net* Namespace
- *WebClient* ist der einfache Wrapper von *HttpWebRequest*
- *HttpWebRequest* wird bei tiefergehenden Erfordernissen genutzt (z. B. Callbacks über Worker-Thread anstatt UI-Thread)
- Asynchroner Aufruf
- API muss verstanden sein 😊

Bereitstellen eines RESTful Services

- *WebGetAttribute* (nur GET)
- *WebInvokeAttribute* (alle Anforderungsmethoden)
- Attribute: URI-Template, Parameter, Nachrichtenformat, Anforderungsmethode usw.

Cross-Domain Calls and Policies

- Zugriff auf Services, die nicht in der selben Domäne liegen, wie die SL-Anwendung liegen
- Silverlight: *clientaccesspolicy.xml*
- Flash: *crossdomain.xml*
- Wohin damit, wenn es kein Root gibt? 😊

Demo