

SNEK – SQL Server und .NET Entwicklerkonferenz

OData

SQL für das Web



Rainer Stropek
software architects gmbh

Web: <http://www.timecockpit.com>
Mail: rainer@timecockpit.com
Twitter: @rstropek



time cockpit
Saves the day.

Einleitung

Was ist OData?

Warum nicht TDS?

- ▶ **Tabular Data Stream Protokoll**
Wurde nicht für das Web gebaut
- ▶ **HTTP de facto Standard im Web**
TDS nutzt HTTP nicht
- ▶ **Proxies, Router, Firewalls**
Können nur schwer mit TDS umgehen
- ▶ **Proprietär, nur SQL Server**
Offene, von DB-Hersteller unabhängiges Protokoll hätte Vorteile

Was ist OData? (1/2)

- ▶ Grundidee: HTTP als Grundlage
GET, POST, PUT/PATCH, DELETE
- ▶ Standardisierte Abfragesprache über URLs
https://api.myserver.com/odata/Customers?
\$filter=CustomerID eq 15&
\$top=10&
\$select=FirstName,LastName
- ▶ Standardisierte Übertragungsformate
XML (Atom), JSON
- ▶ Standardisiertes, erweiterbares Metadatenformat

Was ist OData? (2/2)

- ▶ OData ist Herstellerunabhängig
Unabhängig von SQL Server
Es muss nicht einmal eine relationale DB im Hintergrund stecken

Demo

OData
Einführung

Abfrage eines offenen
OData Service mit
Fiddler

Praxisbeispiel: Abfrage
Active Directory Graph
API mit OData

OData Server

OData Server mit .NET Programmieren

OData Server

- ▶ Integriert in ASP.NET Web API
NuGet Package *Microsoft.AspNet.WebApi.Odata*
- ▶ Entity Framework kann als Basis genommen werden
EF ist nicht Voraussetzung für OData
Linq to Objects oder komplett individuelle Entwicklungen ebenfalls möglich
- ▶ *IQueryable* als Ausgangspunkt für Abfragen
- ▶ POST, PUT/PATCH, DELETE für Schreiben von Daten
Batching zum Abbilden von Transaktionen

Demo

OData Server

Einfaches OData Service auf Basis von OWin
Endpoint zum Lesen von Daten
Endpoint zum Schreiben

OData Service auf Basis von Entity Framework

Veröffentlichen des OData Service im Web

OData Clients
 Verwenden von OData aus .NET und aus dem Browser

OData Client

- ▶ **ServiceReferenz** aus .NET Anwendungen
 DTOs werden generiert
 Datenbankabfragen mit LINQ
 Code sharing mit Serverimplementierung (EF) möglich
 Web requests transparent
- ▶ **Manuelle Web Requests** aus .NET Anwendungen
 Volle Kontrolle über Kommunikationfluss
- ▶ **JavaScript-Bibliotheken** für OData Clients
 Beispiel: Breeze.JS
- ▶ **Manuelle Web Requests mit JSON** aus JavaScript Anwendungen
 TypeScript und Frameworks wie z.B. AngularJS sind eine große Hilfe

Demo

OData Client

- .NET Client mit ServiceReferenz
- TypeScript Client mit JQuery und AngularJS
- Zugriff auf OData Service mit Microsoft Excel

Fazit

- ▶ OData vereinfacht CRUD Web Services enorm
Ein Web Service Endpunkt statt hunderten
- ▶ Konzeptionell sehr ähnlich zu EF und LINQ
Web-Protokolle (vor allem HTTP) zwischen Client und Server
- ▶ Plattform- und Technologieunabhängig
Keine Bindung an DB-Technologie
Keine Bindung an relationale Datenbanken
Beliebige Clients

SNEK – SQL Server und .NET Entwicklerkonferenz

F&A

Danke für die Aufmerksamkeit



Rainer Stropek
software architects gmbh

Mail rainer@timecockpit.com
Web <http://www.timecockpit.com>
Twitter @rstropek



time cockpit
Saves the day.
