

Paweł Rajba

pawel@ii.uni.wroc.pl

<http://www.itcourses.eu/>

OData

Agenda

- Wprowadzenie
- WCF Data Services
- Obsługa żądania OData
- Podstawy języka OData
- Narzędzia i biblioteki
 - Gdzie można skorzystać z OData
- OData w Web API
- Weryfikacja zapytań
- Formaty danych OData

OData

- Otwarty protokół
- Służy do udostępniania danych dla zapytań i modyfikacji przez REST
- Dwie główne składowe
 - OData query syntax: składnia URL dla zapytań
 - OData Formatting: do wyboru ATOM Publishing Protocol (XML) lub JSON

Dostawcy danych przez OData

- WCF Data Services
- WebApi
- Node.js, PHP, Java
- SQL Server Reporting Services
- Sharepoint
- Windows Azure Data Storage
- IBM WebSphere
 - poprzez dodatkowe narzędzia

WCF Data Services

- Framework do budowania usług danych OData
- Dzięki niemu można udostępniać
 - Entity Data Model
 - LINQ2SQL
 - Obiekty .NET (IQueryable)

WCF Data Services

- Po stronie serwera
 - Tworzymy providera dla danych
 - Klasa z kolekcją IQueryable
 - EDM
 - Service tworzymy w oparciu o klasę DataService<T>, gdzie T to źródło danych (np. EF)
 - Service udostępnia kolekcję danych w oparciu o T
 - Jest możliwość dodawania własnych metod
 - Ustawiamy prawa dostępu do service'u
 - Domyślnie wszystko jest zablokowane
 - Opcjonalnie ustawiamy „interceptors”
 - QueryInterceptor (np. security)
 - ChangeInterceptor (np. walidacja dodawanych rekordów)

WCF Data Services

- Klient do WCF Data Services
 - Tworzy klasę proxy na podstawie metadanych OData
 - Dostępny dla .NET, SL, Windows Phone, aplikacji z Windows Store
 - Proxy reprezentuje repozytorium po stronie klienta
 - Udostępnia kolekcję usługi po stronie klienta
 - Konwertuje wyrażenie LINQ na zapytania OData
 - Udostępnia cache dla obiektów
 - Zapytanie wykonuje asynchronicznie

Obsługa żądania OData

- Klient wysyła żądanie z zapytaniem OData
- WCF Data Service przejmuje żądanie i konwertuje je do Expression<T>
- WCF Data Services provider uruchamia odpowiedni provider do pobrania danych
 - Object Context Provider
 - Custom Provider
- Dane wracają do klienta

Podstawy języka OData

- Zapytania
 - \$filter, \$orderby, \$top, \$skip, \$select, \$expand
 - \$inlinecount=allpages - liczba wszystkich rekordów
 - nawet jeśli użyjemy top, skip
- Operatory logiczne
 - and, or, not, eq, ne, lt, gt, le, ge
- Operatory arytmetyczne
 - add, sub, mul, div, mod
- Funkcje napisów
 - substring, length, indexof, startswith, endswith, insert, remove, replace, tolower, toupper, concat, trim
- Funkcje daty
 - day, hour, minute, month, second, year
- Funkcje matematyczne
 - add, sub, mult, div, round, floor, ceiling

Narzędzia i biblioteki

- Narzędzia do prezentacji danych z OData Services
 - OData Explorer (z odata.org)
 - LINQPad (wersja darmowa obsługuje OData)
 - Excel
 - Kontrolki różnych firm

Jak wykorzystać OData Service?

- HTTP
 - natywne programowanie XML/JSON
- AJAX
 - jQuery, DataJS, JayData
- Obecnie są dostępne biblioteki dla większości technologii
 - .NET, Silverlight, WPF, WP7
 - Java, Android
 - iPhone

DEMO

- SimpleDataServices
 - Z solution WcfSamples

Wsparcie OData dla WebAPI

- Uruchomienie OData w WebAPI
 - Możemy utworzyć kontroler z szablonu dla konkretnych encji
 - Metody Get muszą zwracać `IQueryable<T>`
 - Dodajemy atrybut `[Queryable]` nad akcją

Weryfikacja zapytań

- ...czyli inaczej Query Validation
- Można nakładać ograniczenia:
 - Na maksymalną liczbę zwracanych wierszy
 - Nie pozwalać sortowania na nieindeksowanych kolumnach
 - Określać, które operacje są dozwolone
 - Dodatkowo określić parametry atrybutu [Queryable], co pozwala na jeszcze dodatkowe restrykcje

DEMO

- ODataServices

Formaty danych OData

- Dane zwracane przez OData service zawierają
 - Metadane o typach
 - Properties dla danych
 - Relacje pomiędzy encjami w postaci linków
 - Linki do akcji na encjach
- OData może mieć postać
 - ATOM Publishing Protocol
 - JSON-Verbose
 - JSON-Light
 - Mniej metadanych i linków do powiązanych obiektów

Formaty danych OData

- Aby określić format:
 - ATOM Publishing Protocol
 - Accept: application/atom+xml
 - \$format=xml
 - JSON-Verbose
 - Accept: application/json;odata=verbose
 - JSON-Light
 - Accept: application/json
 - \$format=json

DEMO

- Patrzymy na ODataService
 - Wyciągamy dane na 3 sposoby
 - Patrzymy na zawartość
 - Porównujemy rozmiar zwróconych danych