10 Ways to Build Web Services in .NET
WCF Data Web Services

Chad McCallum
@ChadEmm

pluralsight
hardcore developer training

Module Outline

- Introduction to WCF Data Services & OData
- Understanding WCF Data Service Configuration
- Building WCF Data Services with ADO.NET Entity Data Models
- Retrieving and filtering data from a WCF Data Service
- Creating, updating and deleting data from a WCF Data Service
- Retrieving data from a custom data source object
- Creating, updating and deleting data with a custom data source object
- Accessing a WCF Data Service from a client .NET application
- Debugging a WCF Data Service from a client .NET application
- Review

WCF Data Services

- **Windows Communication Foundation**
  - Framework for message-based communication on both client and server
- **… with a specific ASP.NET Runtime Integration**
  - Integrates with the ASP.NET Runtime & IIS to intercept requests and forward them to a data layer

OData

- Application-level protocol for interacting with data via restful web services
- Supports the description of entity sets and entities
- Advanced querying and editing capabilities

- Attempt to make a standard for exposing, structuring, querying, and manipulating data via an API
- Released by Microsoft under the Open Specification Promise

Review

- Introduction to WCF Data Services & OData
- Understanding WCF Data Service Configuration
- Building WCF Data Services with ADO.NET Entity Data Models
- Retrieving and filtering data from a WCF Data Service
- Creating, updating and deleting data from a WCF Data Service
- Retrieving data from a custom data source object
- Creating, updating and deleting data with a custom data source object
- Accessing a WCF Data Service from a client .NET application
- Debugging a WCF Data Service from a client .NET application

OData Data Sources

- **ADO.NET Entity Data Model**
  - Automatically supports read, create, update, and delete operations
- **Custom Type**
  - IQueryable<T> properties exposed as entity sets
  - IUpdatable interface defines methods used to create, update, and delete ALL exposed entities and entity sets

OData Querying

- **Represents data as entity sets and entities**
- **Queries are serialized as a IQueryable query object**
- **Gives a whole bunch of built in capabilities for querying**
  - $filter – filter results by one or more conditions
    - Includes 40 functions for operators, strings, dates & times, math, and types
  - $select – select a subset of properties from an entity
  - $orderby – order by any property, ascending or descending
  - $top, $skip – take the top X results, skip X results
  - $inlinecount, $count – return the number of entities that match the query, return the number of entities in the entity set

OData Updating

- **POST verb – allows creation of new entities**
  - Returns the created content as well as a Location header pointing to the new resource
- **PUT verb – replace an existing resource**
- **PATCH verb – partially update a resource**
- **DELETE verb – delete a resource**
- **Specify the Content-Type header to match the data format you're sending**

OData .NET Client

- **Add reference to an OData service using the Add Service Reference dialog**
- **Access data by querying the generated IQueryable<T> collections**
  - Some LINQ statements are not supported and will throw a NotSupportedException
- **Create, Update, and Delete data by using the appropriate methods**
  - Always call SaveChanges(); when you’re finished to commit the changes