

Writing OData Services in .NET

Get some REST while service enabling your data



Overview

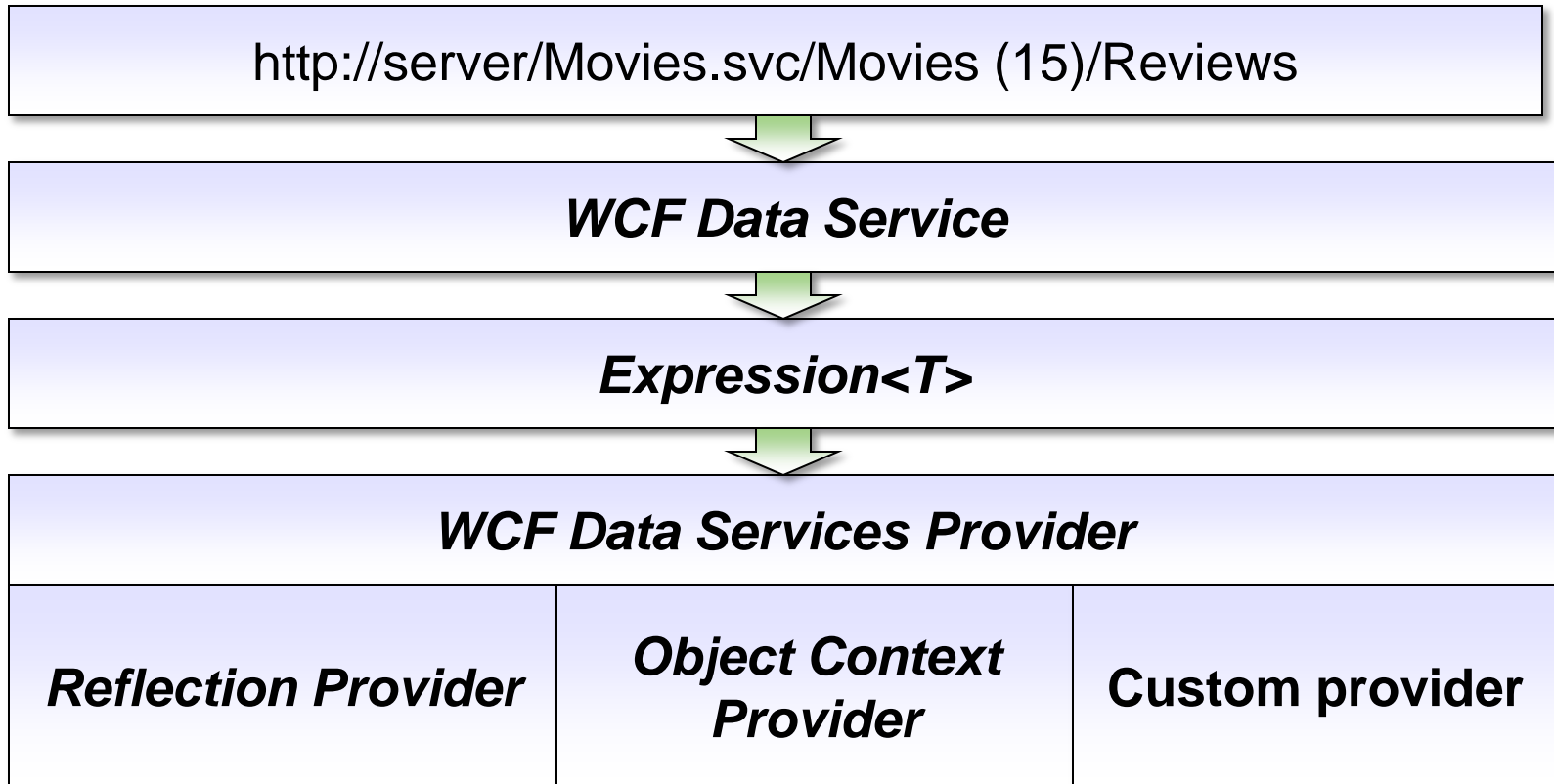
- What is WCF Data Services
- Building services
- Security, filtering and other important stuff

What is WCF Data Services?

- **Framework for building OData services**
 - Entity Data Model
 - LINQ to SQL (query only)
 - .NET objects (IQueryable<T>)
 - Custom providers
- **Built on Windows Communication Foundation**
 - WCF core
 - WCF REST / Web programming model
- **Ships in .NET 3.5 SP 1 and .NET 4**

WCF Data Services

- LINQ's server side role provides storage independence



Building services

Create class derived from DataService



Configure entity and operation access



Augment basic access with custom operations



Define filters and interceptors

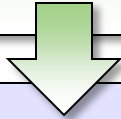
Service operations

- **An OData service is a RESTful WCF service**
 - Has one generic operation by default
 - Add custom operations to the service definition
 - Use WebGet and WebInvoke attributes

Response Formats

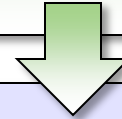
- Negotiated via the HTTP Accept header
- Can choose from Atom (default) and JSON

```
GET /movieservice.svc/Movies(1)
Accept: application/atom+xml
```



```
<?xml version="1.0" ?>
...
  <d:movie_id>1</d:movie_id>
  <d:title>
    Where the Wild Things Are
  </d:title>
...
```

```
GET /movieservice.svc/Movies(1)
Accept: application/json
```



```
{ "d" : {
  ...
  "movie_id": 1,
  "title": "Where the Wild Things Are",
  ...
} }
```

Filters and interceptors

- **Query interceptors apply to GET operations**
 - Allow business rule validation, security, further filtering
 - Only return records for the customer, by department, etc.
- **Change interceptors apply on POST, PUT, DELETE**
 - Allows for security, rule validation, rejection of input, etc.
- **Implement both with operations on the data service class**
 - Use attributes to hook the process
 - Apply to a particular entity in the model

Summary

- WCF Data Services provides a RESTful interface to data
- ADO.Net Entity Framework Models are most common source of data
- IQueryable resources can be fully exposed
- Simplified service development and easy client integration

References

- **WCF Data Services Documentation**
 - <http://msdn.microsoft.com/en-us/library/cc668792.aspx>
- **WCF Data Services team blog**
 - <http://blogs.msdn.com/astoriateam/>