USING ODATA TO CREATE YOUR WEB-API

Alex James (@adjames)
OData Team, Microsoft
What is OData?

- HTTP
- Reach (devices and platforms)
- Metadata
- Formats (ATOM & JSON)
- Semantics (GET/PUT/POST/DELETE)
- Uri Conventions
- Query Language
OData...

- Builds upon
  - Standard HTTP verbs
  - Standard HTTP status codes
  - Standard HTTP headers
  - Standard Formats
  - Standard Interactions

- Adds
  - Flexible Query
  - Metadata
Credit: Flickr/Marcus Hansson, “The best days are not planned”
Why OData?

- Modeling
  - Triples vs Types vs Rows vs Blobs
- Formats
  - Json or Atom
- Platforms
  - iOS or Android or .NET or Javascript or PHP...
- Momentum
  - Excel
  - SharePoint
  - SAP
Demo
Intro and Best Practices
What is JSONP?
Best Practices - Recap

- Lock it down
  - EntitySet Access Rights
  - Server Driven Paging
  - AuthN – Basic, OAuth 2.0, Windows etc.
  - AuthZ – Query & Change interceptors

- Reach
  - Browser friendly feeds
  - Silverlight
  - JSONP
Challenges
Constraints
How is OData improving?

- Better Query – Any/All & Inheritance
- Better Models – MultiValue & Spatial Types
  - /Stores$filter=Category/Name eq "coffee" and distanceto(Location, POINT(-127.89734578345, 45.234534534)) lt 900.0
- Better Efficiency
- Better Behaviors - Actions
Summary

- OData is good for sharing data to many platforms.
- OData is becoming even better:
  - Any/All & Casts & Embedded Collections
  - Spatial
  - Efficient Format
- OData can now handle behaviors without compromise
Resources

- **OData.org** - In particular the Blog and Mailing List
- blogs.msdn.com/astoriateam
- blogs.msdn.com/alexj