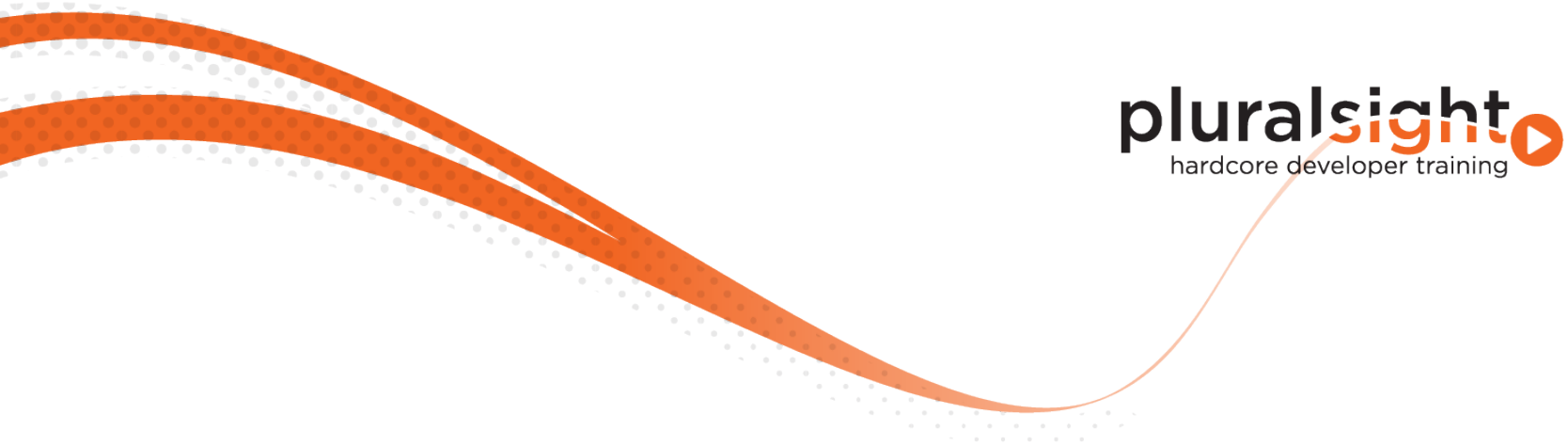
SharePoint 2013 Dev with CSOM and REST – Custom and Legacy Web Services

Rob Windsor
@robwindsor



Module Outline

- **Custom WCF Services**
  - Status and Notifications
- **SOAP Web Services**
- **SharePoint 2010 REST API**

Custom WCF REST Services

- **Why?**
  - Can't do it using CSOM or REST
  - Hard to do it using CSOM or REST
  - Abstraction layer
- **Implementation details**
  - Service must be ASP.NET compatible
    - AspNetCompatibilityRequirements attribute
  - Dynamic configuration factory
    - MultipleBaseAddressWebServiceHostFactory
  - Remainder is just WCF Web/REST service development

```
<%@ServiceHost
    Language="C#"
    Service="<class name>, <assembly name>"
    Factory="Microsoft.SharePoint.Client.Services.
       MultipleBaseAddressWebServiceHostFactory,
       Microsoft.SharePoint.Client.ServerRuntime, Version=15.0.0.0,
       Culture=neutral, PublicKeyToken=71e9bce111e9429c" %>
```

Status and Notifications

- **Status Bar**
  - SP.UI.Status
  - Persistent information
  - Shows below ribbon
  - Different color for different levels of importance
- **Notification**
  - SP.UI.Notify
  - Transient messages
  - Show on right side of page below ribbon
  - Show for five seconds by default

```
var messageId = SP.UI.Notify.addNotification(html, sticky);
...
SP.UI.Notify.removeNotification(messageId);
```

SOAP Web Services

- **ASP.NET (ASMX) Web Services**
  - SOAP
  - XML heavy
- **Have been "deprecated" for last couple versions**

- **Managed code**
  - Use WSDL to add Web reference
- **JavaScript**
  - Can construct SOAP envelopes and call services using jQuery
  - Easier to use SPServices
    - jQuery plugin that wraps calls to SOAP services
    - http://spservices.codeplex.com/

SharePoint 2010 REST API (ListData.svc)

- **CRUD operations on List data**
- **Service does expose metadata**

- **Managed code**
  - Add WCF service reference
  - Use LINQ to query data
- **JavaScript**
  - Use service basically the same way as SharePoint 2013 REST API

```
var serviceUrl = "http://localhost/sites/demo/_vti_bin/ListData.svc";
var dataContext = new DemoProxy.DemoDataContext(new Uri(serviceUrl));
dataContext.Credentials = System.Net.CredentialCache.DefaultCredentials;

var category = "Beverages";
var products = from product in dataContext.Products
               where product.Category.Title == category
               select product;

ResultsListBox.Items.Add(category);
foreach (var product in products) {
    ResultsListBox.Items.Add(product.Title);
}
```

Summary

- **Custom WCF Services**
  - Status and Notifications
- **SOAP Web Services**
- **SharePoint 2010 REST API**