SharePoint 2010
Managed Client Object Model

Jason Cribbet
Summit 7 Systems Inc.
Senior Developer
http://summit7systems.com/blogs/jasoncribbet
@jcribbet
Agenda

• Intro
• Architecture
• Usage
• OM Objects
• Demo
• Benefits & Limitations
What is the SP Client Object Model?

- Object Oriented approach to access SP Data
- 3 APIs for interacting with SharePoint Sites
  - .Net App (.Net Framework 3.5+)
  - Silverlight App (2.0+)
  - ECMAScript (JavaScript, Jscript)
- Create and Modify many SharePoint objects
  - Add and Remove Lists
  - Add, Update, and Delete List Items
  - Change documents in document libraries
  - Create Sites
  - Manage Permissions of items
  - Add and remove Web Parts from page
2007 Architecture

Server Side

- Web Parts, Custom Controls, etc.
- JavaScript/Managed Code
- Web Services
- Server OM
- Content Database

Client Side
2010 Architecture

JavaScript Controls and Logic

Browser

JavaScript OM

Proxy

Server OM

Content Database

Client.svc

Proxy

Managed OM

Managed Client

Managed Controls and Logic

SharePoint Server

XML Request

JSON Response

XML Request

JSON Response
Usage

• .Net Assemblies
  – Microsoft.SharePoint.Client.dll
  – Microsoft.SharePoint.Client.Runtime.dll

• ECMAScript
  – Include “sp.js” within the page

<Sharepoint:ScriptLink name="SP.js" runat="server" OnDemand="true" localizable="false"/>

  – Ensure “sp.js” loaded before execution

  // Trigger the loading of data after the page completes the load of sp.js
  ExecuteOrDelayUntilScriptLoaded(populateForm, "sp.js");
OM Objects

- Server OM vs. Client OM

<table>
<thead>
<tr>
<th>Server Side Classes</th>
<th>Client Side Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPContext</td>
<td>ClientContext</td>
</tr>
<tr>
<td>SPSite</td>
<td>Site</td>
</tr>
<tr>
<td>SPWeb</td>
<td>Web</td>
</tr>
<tr>
<td>SPList</td>
<td>List</td>
</tr>
<tr>
<td>SPListItem</td>
<td>ListItem</td>
</tr>
<tr>
<td>SPField</td>
<td>Field</td>
</tr>
</tbody>
</table>
Key Code Objects

- `ClientContext.Load(clientObject, retrievals);`
- `ClientContext.ExecuteQuery();`

```csharp
using System;
using Microsoft.SharePoint.Client;

class DisplayWebTitle
{
    static void Main()
    {
        ClientContext clientContext = new ClientContext("http://intranet.contoso.com");
        Web site = clientContext.Web;
        clientContext.Load(site);
        clientContext.ExecuteQuery();
        Console.WriteLine("Title: {0}", site.Title);
    }
}
```
Demo

1. Respond to link with a SP Dialog
2. Accept user input
3. Create a List Item
Benefits

• Client OM looks similar to Server OM
• Common API across ECMA, Silverlight, and .Net managed code
• Abstracts the request details
• Can access SP data remotely
• Access data directly without using Web Services
Limitations

- No elevation of privilege
- Can’t access WebApplication or Farm objects
- Requests are throttled
- Silverlight and JavaScript are asynchronous
Questions?

Jason Cribbet
Summit 7 Systems Inc.
Senior Developer
Twitter: @jcribbet
http://summit7systems.com/blogs/jasoncribbet