## General Notes about Delivering the Class

Make sure instructor and student VMs have a reliable Internet connection. If they do not, some labs cause problems because the use NuGet Package manager. If you do not have a good internet connection (like my last class at TwechSherpas) then students need to grab .js files from Solutions folder of the lab.

I am not sure which version of the VM you are using but my guess is that it has only VS 2012 installed but not VS 2013. That isn't a huge issue because the first three days of labs are still based on VS 2012. You can use either version of VS but the instructions in the lab manual tell students to launch VS2012. but I think you should have the students install VS 2013 on their VMs at the end of the day on Monday or Tuesday. The first lab which tells students to use VS2013 is WebAPI lab on Thursday morning. The lab uses CORS and I have not tested it with VS2012 which is why I would recommend having folks install VS2013 sometime before Thursday at the end of the day so it does not interfere with lab work.

## Day 1

### SharePoint App Model Architecture

Overview issues with SharePoint solutions and cover basics of SharePoint app model. I do 2-3 demos with SharePoint hosted apps. I make sure I show apps from the user's perspective by creating a Hello World app and hitting F5 and then show how app shows up on Site Content page.

Basic lab with exercises to create "hello world" app for SharePoint hosted app model and for provider-hosted app model. I usually take lab up through lunch time

### JavaScript and jQuery Primer

Basic but quick coverage of JavaScript language building up to understanding JavaScript modules and the idea of promises and deferred objects. I demo with the following SharePoint-hosted app.

* C:\Student \Modules\JavaScript\Demo\jsFiesta

Lab is pretty good at having them build a project from scratch and add in JavaScript files with links

### Developing SharePoint-hosted Apps

The lecture discusses basics of SharePoint hosted apps as well as the creation of app parts and custom UI actions. During section title "User Interface Design Techniques" I demo this SharePoint-hosted app project to show a multipage app design using an ASP.NET master page.

* C:\Student\Modules\SharePointHostedApps\Demo\MultiPageSharePointApp

After that I show them this simple demo and talk about the benefits of an SPA over a multipage app.

* C:\Student\Modules\SharePointHostedApps\Demo\SinglePageSharePointApp

In the last two sections I do on the fly demos using new SharePoint-hosted app projects to show the development process for app parts and UI custom actions. You can also demo the lab solution for the last exercise to show an app part with custom properties.

Not that the lab is all about building app parts. The lab does not currently contain an exercise on creating a custom UI action so I make sure to demo that.

## Day 2

### Programming with REST and OData

Note that this lecture introduces REST and OData without involving SharePoint. The next lecture is when SharePoint is brought into the story.

Run through the lab once because the lecture and lab follow each other quite closely. You will create a SQL database named WingtipCRM which is used in this lab and many upcoming labs. Next, you will create an ASP.NET project to create a OData web service using the Entity Framework and a WCF Data Service. This works great with a demo because you can then run OData queries through the browser and show students the returned XML results in the browser. This makes it possible to demo around the slides which show using OData query options such as $select, $filter and $orderby.

The second part of the lecture covers writing JavaScript to use the OData service. Here is where the course starts pitching the idea of creating reusable modules with Data Access code. Therefore, slides and labs both emphasize separation-of-concern practice of separating view code from data access code.

For the demo and lab, the customer.aspx page and the JavaScript code behind it goes into the same ASP.NET project with the WCF web service. You can use the lab solution to figure out what to demo.

* C:\Student \Modules\REST\Lab\Solution\WingtipCRM\WingtipCRM

### The SharePoint REST API

*Note this lecture does not currently have a lab. It needs one but the lab in-progress is not ready yet.*

This lecture adds SharePoint into the OData story that started last lecture. Also note that I recently added many slides to this lecture from my REST/OData breakout session at SPC2014. Since there is no lab for this lecture, I usually do long demos. Here is a list of the demos I present in this lecture.

* C:\Student\Modules\SharePointRest\Demo\HostWebExplorer
* C:\Student\Modules\SharePointRest\Demo\SPRestPagingDemo
* C:\Student\Modules\SharePointRest\Demo\SPRestPagingDemo2
* C:\Student\Modules\SharePointRest\Demo\SharePointCRM
* C:\Student\Modules\SharePointRest\Demo\HtmlSlingersSmackdown

You should watch the SPC2014 recording if you want to see me demo the following projects.

<http://channel9.msdn.com/Events/SharePoint-Conference/2014/SPC323>

The last demo named HtmlSlingersSmackdown is a demo on the "fun" side of things. Use it if you have students with a sense of humor. It shows 6 different pages which all use the exact same OData result to show a table but they all use different techniques. I like to show this demo as the last thing in this module because it starts building anticipation for the next two lectures on Knockout and Angular.

### Developing Apps using MVVM and the KnockoutJS Library

Basic Introduction to Knockout. When I teach this lecture, I baby-step students through getting going with Knockout using the **LearningKnockout** demo which is an ASP.NET site and not a SharePoint app.

* C:\Student\Modules\MVVM\Demo\LearningKnockout

At the end of the lecture, you can demo this SharePoint-hosted app to show using Knockout can be used against a SharePoint list.

* C:\Student\Modules\MVVM\Demo\RestMVVM

## Day 3

### Developing Apps using the AngularJS Framework

This provides a basic intro to using the Angular framework. I do not have much in the way to demo yet other than the lab.

### Developing Provider-hosted Apps

This lecture has topics you have covered many times before.

For final demo Cross Domain Library use this project.

C:\Student\Modules\ProviderHostedApps\Demo\CrossDomainDemo

### Developing SharePoint Apps with ASP.NET MVC

Basic intro to MVC framework.

## Day 4

### Creating Custom Web Services using the Web API

Basic intro to Web API

This is the lab that requires VS2013.

### App Authentication using S2S Trusts

This lecture covers S2S but not OAuth which is covered later.

Lab is short and sweet. Should probably only take about 20 because all they do is run PowerShell scripts to create a cert and register SPSecureTokenIssuer and then do hello world CSOM and verify it carried the Authorization header in fiddler.

One to do item I have is to add a section to slides and another lab exercise about writing server-side C# code which executes REST calls back to the host web in the SharePoint host. Just about all the REST code in the demos and labs for the course are currently client-side examples so it will be nice to get some server-side C# code using REST to round things out. It will also be a good motivation for CSOM programming with C# in the next lecture once they see how grungy and loosely-typed REST programming in server-side scenarios using C#.

### CSOM Programming

*Note this lecture does not currently have a lab. It needs one but the lab in-progress is not ready yet.*

This lecture covers basic to advanced coverage of CSOM. I usually do the first demo by creating a C# Console App and programming against a local SharePoint site. I use the following Provider-hosted app project to show examples of server-side CSOM code for demo #2 and demo #4.

* C:\Student\Modules\CSOM\Demo\ServerSideCSOM

Note that the ServerSideCSOM demo assumes you have complete the S2S lab and that you have created the certificate at c:\Certs\appserver.wingtip.com.pfx and that you have registered the SPSecurityTokenIssuer.

For demo #3, I show and run the CreateProductsList project which is a C# Console App which create site columns and a Products content type and then a Products list based on it. The motivation behind the demo is that CSOM makes it easy to create site columns, content types, lists and document libraries.

* C:\Student \Modules\CSOM\Demo\CreateProductsList

## Day 5

### Building Apps with SharePoint Services

This lecture covers a selection of SP services including User Profile Service, Managed Metadata Service, SharePoint Search Service and Business Connectivity Services (BCS). Currently, there is no coverage of workflow in the course. You should feel free to go into that as you obviously hae a lot to say. I would love to hear from you after teaching the course, what you would recommend as far as adding workflow content into the class.

### App Authentication using OAuth

This lecture pretty short and it discusses various aspects of using OAuth in Office 365. I use my Office Dev site from MSDN to demo this app.

C:\Student\Modules\OAuth\Demo\AutohostedOAuthDemo

However, we will not be creating a lab for this module as we do not know how to ensure that each student will have an Office 365 dev site. However, this might change over time.

### Publishing, Installing and Versioning Apps

Grungy details. Demo creating a App Catalog in the <http://wingtipserver> web app and then publishing an app. Demo site-scoped installation and then demo tenancy-scoped installation. At time I have then demos using the App Catalog in the tenancy of my Office 365 dev site to show them it works the same way.

Note this lab is a bit buggy and I need to correct the URLS used to deploy the provider-hosted apps. Hopefully, I can work through that early this week and send you updated lab files for this lecture you will give next Friday.