



A Developer's Introduction to SharePoint 2013 Apps

Ryan McIntyre, MCITP, MCPD
National Architect
[@ryanmcintyre](#)



Agenda



- Why Apps
- App Model
- App Hosting Models
- Development Strategies

Introducing SharePoint App Model



- SharePoint applications no longer live in SharePoint
- Custom code executes in the client, cloud or on-prem
- Apps are granted permissions to SharePoint via OAuth
- Apps communicate with SharePoint via REST / CSOM
- Acquire apps via centralized Marketplace
 - Corporate Marketplace
 - Public Marketplace (via submission process)
 - APIs for manual deployment

App Principles



Apps are a very different philosophical change to extending SharePoint

- More scenario-focused
 - Event Tracking
 - Ticket Management System
- Robust
 - Built-in robust semantics for install, upgrade & uninstall
- Apps are for End Users
- Cloud & Web-Oriented

User Experience



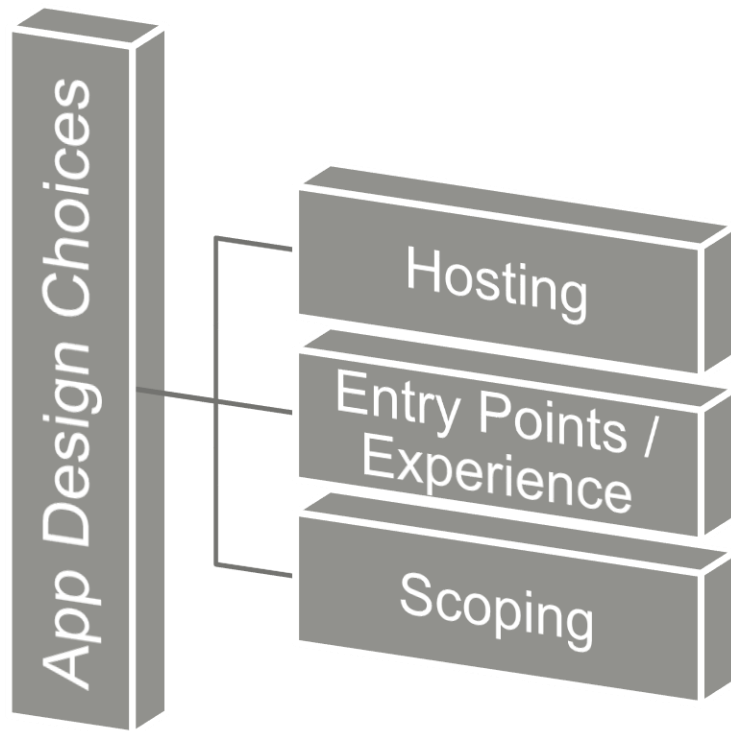
- Everything in a SharePoint site is an app
 - Contact form
 - Travel request
 - Shared Documents library
 - Contacts list
- Formerly known as **View All Content**, the **Site Contents** link shows all apps in a site & links to the SharePoint Store (public) and App Catalog (internal)
- Selecting an app redirects to the app's start page

SharePoint Solutions & Apps



	Full-Trust Solutions	Sandboxed Solutions	Apps
Use Client-Side SharePoint API	★	★	★
Use Server-Side SharePoint API	★	★	✗
Use Remote Services	✗	★	★
App-based Permissions (OAuth2)	✗	✗	★
On-Premise Deployment Friendly	★	★	★
Hosted Deployment Friendly	✗	★	★
Distribution via Marketplace	✗	✗	★
SharePoint Provided Schematics for Install / Upgrade / Uninstall	✗	✗	★

Choices when Building Apps



Hosting: Choice of Three Architecture Approaches



Cloud-based Apps

Get remote events from SharePoint
Use CSOM/REST + OAuth to work with SP

Provider-Hosted App

"Bring your own server hosting infrastructure"

SharePoint Web

Your Hosted Site

~~Autonomous App~~

Windows Azure + SharePoint
provisioned individually as apps are installed

SharePoint Web

Windows Azure Websites

SharePoint-Hosted App

Provision an isolated sub web on a parent web

- Reuse web elements (lists, files, out-of-box web parts)
- No server code allowed; use client JavaScript for logic, UX

Parent Web

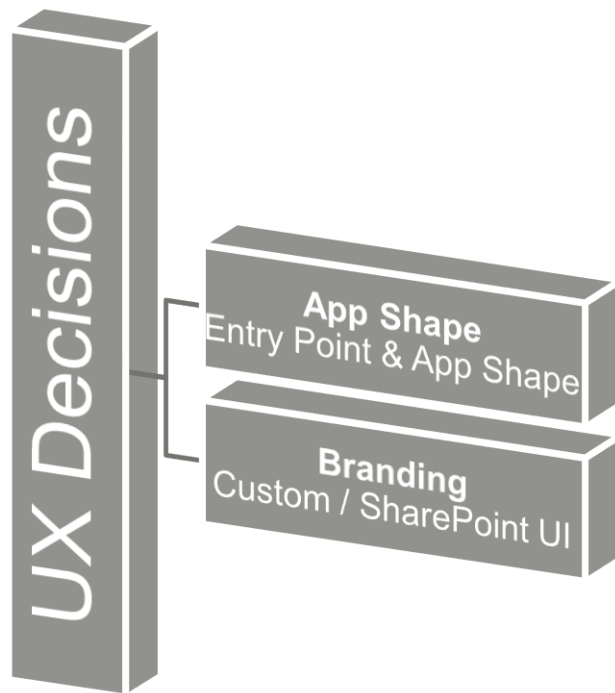
App web (from WSP)

Hosting: Cloud vs. SharePoint



Cloud Hosted Apps	SharePoint Hosted Apps
<i>Preferred hosting model for almost all types of apps</i>	<i>Good for smaller apps & resource storage</i>
Full power of web – choose your infrastructure & technology	SharePoint-based; no server-side code
May require your own hosting	Automatically hosted in SharePoint
May require you own handling of multitenancy & permission management	Inherent multitenancy & isolation

App User Experience Decisions



Entry Point / Experience: App Shapes

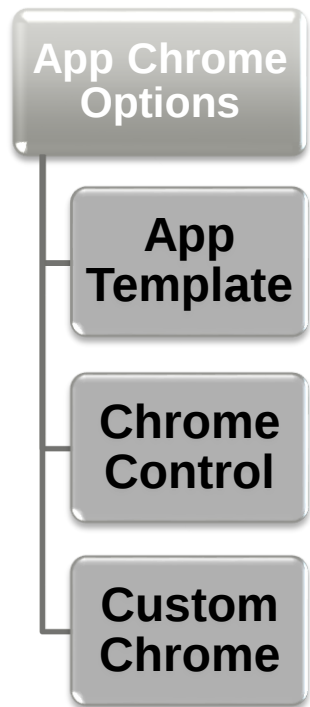


Shape	Description	Example
	Immersive Full Page App	Resource Tracking, Budgeting
	App Part	Weather, Team Mascot, News
	Extension App	Display Document Visualization, Print to Print Service Vendor

App Branding

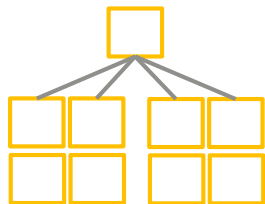


- User Experience Guide for SharePoint 2013 Apps
- Options
 - App Template
 - ASPX page hosted in SharePoint
 - app.masterpage: includes AppWeb chrome
 - Default option when creating apps with VS
 - Chrome Control
 - Custom Branding



- *Must include “Back to Site” link in upper-left*

App Scoping



- **Web Scope**

- Can register and use resources in parent site, site collection

- **Tenant Scope**

- Can register start page, custom actions
- Tenant Admins can filter-enable on sites
 - By Site Collection, Managed Path, Template Type
- (SharePoint-hosted tenant-scope apps not available)

Application Isolation



- When apps are provisioned, new SPWeb (AppWeb) created within hosting SPWeb
 - Each app resides within it's own SPWeb for isolation
 - Special DNS address configured by administrators
 - App SPWeb's live in separate domain (DNS)
- Each App hosted on it's own unique URL because:
 - Blocks XSS: isolation to special SPWeb under special domain blocks cross site scripting
 - Enforces App Permissions: apps communicate with sites via CSOM / API & must be granted to do so

App Model Type 1: SharePoint Hosted



- SharePoint hosted apps wholly reside in SharePoint
- Uses SharePoint artifacts (lists/libraries)
- Business logic executes or on the client
 - HTML5
 - JavaScript

App Model Type 2: Cloud Hosted



- Cloud hosted apps primarily execute outside of SharePoint
- May use SharePoint artifacts (lists/libraries)
 - Communicate via CSOM / REST
 - Granted permission to SharePoint via OAuth
- Business logic lives & executes outside of SharePoint
 - On-Premise hosted web application
 - Windows Azure
 - 3rd party host

New Context



```
var spContext = SharePointContextProvider.Current.GetSharePointContext(Context);

using (var clientContext = spContext.CreateUserClientContextForSPHost())
{
    clientContext.Load(clientContext.Web, web => web.Title);
    clientContext.ExecuteQuery();
    Response.Write(clientContext.Web.Title);
}
```



Separation of Concerns

Typical SharePoint 2010 code



```
protected override void CreateChildControls()
{
    //Create the list selector
    listOfSiteLists = new DropDownList();
    ListItem newItem;
    //Get the current SPWeb and find all the lists
    using (SPWeb currentWeb = SPContext.Current.Web)
    {
        foreach (SPList currentList in currentWeb.Lists)
        {
            //Add each SharePoint list to the drop-down list
            newItem = new ListItem();
            newItem.Text = currentList.Title;
            newItem.Value = currentList.ID.ToString();
            listOfSiteLists.Items.Add(newItem);
        }
    }
    this.Controls.Add(listOfSiteLists);
}
```

What if we did this...



```
protected override void CreateChildControls()
{
```

```
    //Create the list selector
    listOfSiteLists = new DropDownList();
    ListItem newItem;
```

```
    ISPContent MySPContent = Injector.Get("SPContent");
    List<ISPList> mySPListList =
        MySPContent.GetListsForCurrentWeb();
```

```
    foreach (ISPList list in mySPListList)
    {
        //Add each SharePoint list to the drop-down list
        newItem = new ListItem();
        newItem.Text = list.Title;
        newItem.Value = list.ID;
        listOfSiteLists.Items.Add(newItem);
    }
```

```
    this.Controls.Add(listOfSiteLists);
```

```
}
```

```
interface ISPList
{
    string Title { get; }
    string ID { get; }
    void Add (string title, string id);
}
```

```
public List<ISPList> GetListsForCurrentWeb()
{
    List<ISPList> mySPListList = GetSPListInstance();
    using (SPWeb currentWeb = SPContext.Current.Web)
    {
        foreach (SPList currentList in currentWeb.Lists)
        {
            mySPListList.Add(currentList.Title,
                             currentList.ID.ToString());
        }
    }
    return mySPListList;
}
```

Then we could do this...



```
protected override void CreateChildControls()
{
    //Create the list selector
    listOfSiteLists = new DropDownList();
    ListItem newItem;

    ISPContent MySPContent = Injector.Get("SPContent");
    List<ISPList> mySPListList =
        MySPContent.GetListsForCurrentWeb();

    foreach (ISPList list in mySPListList)
    {
        //Add each SharePoint list to the drop-down list
        newItem = new ListItem();
        newItem.Text = list.Title;
        newItem.Value = list.ID;
        listOfSiteLists.Items.Add(newItem);
    }

    this.Controls.Add(listOfSiteLists);
}
```

```
public List<ISPList> GetListsForCurrentWeb()
{
    List<ISPList> mySPListList = GetSPListInstance();

    #region GetContext Token and hostWeb

    using (var context =
        TokenHelper.GetClientContextWithContextToken
            (hostWeb, contextToken, Request.Url.Authority))

    Web web = context.Web;

    IEnumerable<SP.List> result =
        context.LoadQuery(web.Lists.Include(
            list => list.Title, list => list.Id));

    context.ExecuteQuery();

    foreach (List list in web.Lists)
    {
        mySPListList.Add(list.Title, list.Id);
    }

    return mySPListList;
}
```



Loosely Coupled Code

Resources



- **Branding & Site Provisioning -**
<http://www.microsoft.com/en-us/download/details.aspx?id=42030>
- **Office 365 Development Patterns & Practices -**
<https://github.com/OfficeDev/PnP>
- **Office Developer Blogs -** <http://blogs.office.com/dev/>
- **Me -** <http://blog.randomdust.com>