LightSwitch Demo Script

# Setup

* VS 2013 installed
* Have IE set as browser
* Web Essentials
* Electic plumb stuff installed + icon on desktop
* Chrome & Firefox Nightly
* Make sure board games DB is setup in local DB
* PresentOn
* Zoomit
* Cleanup localdb
* Turn off Windows 8 notifications
* Ensure no pin tabs in chrome
* Disable JavaScript bug (may need to add it)
* 
* Offline Nuget backup
* Custom Side Waffle
* Nuget Local

# Demo

Talk about new Visual Studio – sign in, notifications, extensions, updates

New solution – discuss desktop vs. HTML application

Solution explorer, right click add to source control. Talk about Git briefly

In solution explorer - Discuss the old way of doing things

<SLIDES – show how to do this in 2012>

Add data source – board games on localdb. Discuss localdb.

Add games

Polish DB, Server perspective:

* BgID – hide
* YearPublisher – rename to yearpublished
* Ratings – hide
* Description – hide

Add browse screen for games

RUN!

Showin electric plumb

End demo

Change to tile, remove max & min players, rank



In the data source, order by rank

Add an OPTIONAL parameter to filer by name contains

Add a popup to the screen named Filter – drag on parameter

Use the built in tools to add an edit selected screen.

RUN

SHOW PROPERTIES & PUBLISH!!!!

Add new data > data sources, new table > Votes

Show new created & modified properties!!!

Add Email Address field, Discuss new person type (not suitable for our demo)

Add a relationship to the Game table!
Add screen to browse votes – use the new table type.

Add new screen – no data, call it home. Right click set as home screen

Add a button to browse games & another for votes.

Add data item, votesset.

Add custom control > Screen

Edit render code – talk about reference

myapp.Home.ScreenContent\_render = function (element, contentItem) {

 // Write code here.

 contentItem.screen.getVotesSet().then(function (x) {

 var value = "Votes Cast: ";

 value += x.count;

 element.innerText = value;

 });

};

On LightSwitch Server:

Right click on the server project – side waffle, web api empty

public void Post([FromBody]dynamic value)

{

 string emailAddress = value.emailAddress;

 int game = value.game;

 using (var server = LightSwitchApplication.ServerApplicationContext.CreateContext())

 {

 var existingVote = (from g in server.DataWorkspace.ApplicationData.VotesSet.GetQuery().Execute()

 where g.EmailAddress.Equals(emailAddress, StringComparison.InvariantCultureIgnoreCase)

 select g).SingleOrDefault();

 if (existingVote != null)

 {

 existingVote.Game\_Id = game;

 }

 else

 {

 var vote = server.DataWorkspace.ApplicationData.VotesSet.AddNew();

 vote.EmailAddress = emailAddress;

 vote.Game\_Id = game;

 }

 server.DataWorkspace.ApplicationData.SaveChanges();

 }

}

Add new empty web project

Add jquery & datajs, bootstrap through nugget – talk about nugget

Add default.html

Add Demo.js – minify it

/// <reference path="jquery-2.0.3.js" />

/// <reference path="datajs-1.1.1.js" />

$(function () {

 OData.read("http://localhost:25962/BoardgamesData.svc/Games?$select=Id,Name&$orderby=Rank&$top=100", function (data) {

 var gamesList = $("#game");

 $.each(data.results, function (index, element) {

 var option = document.createElement("option");

 option.innerText = element.Name;

 option.value = element.Id;

 gamesList.append(option);

 });

 });

 $("#voteButton").on("click", function () {

 var gameListField = $("#game");

 var emailAddressField = $("#emailAddress");

 $.ajax({

 type: "POST",

 contentType: "application/json; charset=utf-8",

 url: "http://localhost:25962/api/Votes",

 data: "{ emailAddress: '"+emailAddressField.val()+"', game: '"+gameListField.val()+"' }",

 });

 });

});

Last back in LS, do the go & vote button: window.open("http://localhost:38351/");