LightSwitch Demo Script

# Setup

* Visual Studio 2012
* Board Games Database
* Update 2
* IE as default browser
* PresentOn
* Copy highcharts to desktop
* Zoomit
* Cleanup localdb
* Turn off Windows 8 notifications
* Ensure no pin tabs in chrome

# Demo 1

Start with data, create table:



Talk about custom field types when adding money

Add screen – list & details. Run. Add a product, show no auto-product id & negative numbers.

Fix the negative issue and then, add to the on creating event the following code:

this.ProductId = this.DataWorkspace.ApplicationData.ProductsSet.GetQuery().Execute().Count().ToString("000000");

Run.

Swop to web. `And custom UI & Turn on authentication. Run again.

Publish

# Demo 2: Present



Create a screen and add some text.

Show service publishing story.

Add a customer, then add new empty web application.

Add an HTML page. Add jQuery & datajs (Nuget)

    <script src="Scripts/jquery-1.9.1.js"></script>

    <script src="Scripts/datajs-1.1.0.js"></script>

    <script type="text/javascript">

        $(function(){

            OData.read("http://localhost:5036/applicationdata.svc/Customers",

                function (data) {

                    var body = $("body");

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

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

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

                        li.innerText = customer.Name;

                        ul.appendChild(li);

                    })

                    body.append(ul);

                }

            );

        });

    </script>

Show the publish options

# Demo 3: Future

Add HTML client, Add board games data source.

Add a screen, make it tiles, polish, add filter and view & edit.

Run

Swop to tiles & run.

Run

Add Item Tap to tile list:



Add web api controller (GamesController) to server project.

Add global.asax to server project, add **namespace: System.Web.Http;** Add following to APPLICATION start up:

   RouteTable.Routes.MapHttpRoute("DefaultApi", "api/{controller}/{id}", new { id = System.Web.Http.RouteParameter.Optional });

In the controller add the following namespace: **using Microsoft.LightSwitch;**

public IEnumerable<Tuple<int,int>> Get()

        {

            using (var serverContext = ServerApplicationContext.CreateContext())

            {

                return (from g in serverContext.DataWorkspace.BoardgamesData.Games.GetQuery().Execute()

                        where g.YearPublisher > 0

                        group g by g.YearPublisher into groups

                        let count = groups.Count()

                        orderby count descending

                        select Tuple.Create(groups.Key, count)).Take(10);

            }

        }

Back in, switch to file view, add Highcharts to to the html client project. You need to edit the HTML page to add in the bits. Also good time to do demo of colour changes



Format the UI, add a content control (Screen) and drag on the highcharts code

$.get("http://localhost:6997/api/games", function (data) {

 var dataSet = [];

 $.each(data, function (index, v) {

 dataSet.push([v.m\_Item1.toString(), v.m\_Item2]);

 });

 $(element).highcharts({

 chart: {

 plotBackgroundColor: null,

 plotBorderWidth: null,

 plotShadow: false

 },

 title: {

 text: "Games per year"

 },

 tooltip: {

 pointFormat: "{series.name}: <strong>{point.percentage}%</strong>",

 percentageDecimals: 1

 },

 plotOptions: {

 pie: {

 allowPointSelect: true,

 cursor: 'pointer',

 dataLabels: {

 enabled: true,

 color: '#000000',

 connectorColor: '#000000',

 formatter: function () {

 return '<strong>' + this.point.name + '</strong>: ' + this.percentage.toFixed(2) + ' %';

 }

 }

 }

 },

 series: [{

 type: 'pie',

 name: 'Percentage',

 data: dataSet

 }]

 });