Things to show off

Icenium

Blend

* -ms-grid
* Interactive mode

Remote Debugging

DROW -

* Pickers
* PLM

Short Cuts

* Refresh in place - ctrl + shift + r

# Web Demo

1. Open Template
2. Browse to the default page just to show it’s blank.
3. Copy over the required files.

**(POWERSHELL)** add the necessary files for the

ps .\**webCopy.ps1**

**(d21)** Add the JavaScript libraries to the header

<script src="/scripts/jquery-2.0.0.min.js"></script>

<script src="/scripts/knockout-2.2.1.js"></script>

**(d22)** Create the knockout template

<script type="text/html" id="speaker-template">

<div class="item">

<h2 class="title">

<span data-bind="text: FirstName"></span> <span data-bind="text: LastName"></span>

</h2>

<p class="desc" data-bind="text: Bio"></p>

</div>

</script>

**(d23)** Reference the template

<header>Speakers</header>

<div id="content" data-bind="template: { name: 'speaker-template', foreach: d }"></div>

**(d24)** Create the JavaScript block to pull from That Conference

<script type="text/javascript">

$(document).ready(function () {

$.ajax({

url: 'http://www.thatconference.com/odata/api.svc/People',

dataType: "json",

success: function (data) {

ko.applyBindings(data);

}

});

});

</script>

**(d25)** Style the controls

@font-face {

font-family: "Shadows";

src: url("/fonts/ShadowsIntoLight.woff") format('woff');

}

@font-face {

font-family: "Source";

src: url("/fonts/SourceCodePro-Regular.woff") format('woff');

}

header, h1, h2 {

font-family: 'Shadows', cursive;

}

header {

font-size: 4em;

background-color: white;

color: white;

text-shadow: black 2px 2px 10px;

}

#content {

margin: 25px;

}

.desc {

font-family: 'Source', sans-serif;

margin: 25px;

}

**(d26)** Style the controls

<link href="/style.css" rel="stylesheet" />

1. Run
2. Icenium

# HTML5 App Demo

1. Open the starting template
2. **RUN** Browse to the default page just to show it’s blank.

**(POWERSHELL)** add the necessary files for the

ps .\**w8Copy.ps1**

1. Re-Run – **(D21) – (D26)**
2. Now start to break up the JS explaining some of how W8 works
3. Remove the script block and add our app onloaded

**(d27)** add the necessary files for the

app.onloaded = function () {

}

1. Remove the script block and add our app onloaded

**(d24)** Re-add

1. Copy the previous jquery xhr call to our new onloaded event

**(d28)** – replace our jquery call with winjs.xhr

WinJS.xhr({

url: "http://www.thatConference.com/odata/api.svc/People",

headers: { accept: "application/json" }

}).then(

function (args) {

var obj = JSON.parse(args.responseText);

ko.applyBindings(obj);

},

function (args) {

//error

});

# HTML5 AppBar Demo

For the app bar demo what we’re going to do is create an app bar button that we will later launch the media capture.

1. Create just the app bar

**(ab1)** add our app bar and button

<div id="appbar" data-win-control="WinJS.UI.AppBar">

</div>

1. Run to show off the app bar button

**(ab2)** add our app bar and button

<button data-win-control="WinJS.UI.AppBarCommand"

data-win-options="{id:'cmd', label:'picture', icon:'placeholder'}"

type="button"

id="picture">

</button>

1. Run to show the button
2. Change the icon to ‘Camera’
3. Rerun

**(ab3)** Wire up a button handler

document.querySelector("#picture").addEventListener("click", imageCapture);

**(ab4)** implement the button handler

function imageCapture() {

var \_capture = Windows.Media.Capture;

var captureUI = new \_capture.CameraCaptureUI();

captureUI.photoSettings.format = \_capture.CameraCaptureUIPhotoFormat.png;

captureUI.photoSettings.croppedAspectRatio = { height: 4, width: 3 };

captureUI.captureFileAsync(\_capture.CameraCaptureUIMode.photo)

.then(function (capturedItem) {

if (capturedItem) {

var photoBlobUrl = URL.createObjectURL(

capturedItem,

{ oneTimeOnly: true });

var imageElement = document.createElement("img");

imageElement.setAttribute("src", photoBlobUrl);

document.querySelector(".pic").appendChild(imageElement);

}

});

}

1. Run, should fail because of manifest capabilities
2. Update capabilities
3. Re-Run

**(ab5)** We need a div to dump our image to, **before appbar**

<div class="pic"></div>

# HTML5 Settings Demo

Here we’re going to wire up the settings charm.

1. Pages are already there.
2. Add the onsettings function to wire up settings to the app

**(ss1)** add our app bar and button

app.onsettings = function (e) {

e.detail.applicationcommands = {

"about": {

href: "/pages/settings/aboutflyout.html",

title: "not about"

},

"help": {

href: "/pages/settings/helpFlyout.html",

title: "Help"

}

}

WinJS.UI.SettingsFlyout.populateSettings(e);

};

1. Show off the pages.
2. How would we create a new page

# HTML5 Style It

Let’s just have fun with some CSS

1. Pages are already there.
2. Add the onsettings function to wire up settings to the app

**(css1)** Style the Body and setup the grid that we will later use

body {

height: 100`%;

width: 100`%;

overflow-y: scroll;

display: -ms-grid;

-ms-grid-columns: 120px 1fr 120px;

-ms-grid-rows: 120px 1fr 120px;

}

**(css2)** Style the header

header {

-ms-grid-column: 2;

-ms-grid-row: 1;

font-size: 500`%;

}

**(css3)** Style the content but also now add the flex box

#content {

-ms-grid-column: 2;

-ms-grid-row: 2;

display: -ms-box;

-ms-flex-align: middle;

-ms-flex-pack: justify;

-ms-flex-direction: normal;

-ms-flex-wrap: normal;

}

**(css4)** Style each item

.item {

display: -ms-grid;

-ms-grid-columns: 1fr 120px;

-ms-grid-rows: 25px 1fr;

margin-bottom: 10px;

}

**(css5)** Style each title

.title {

-ms-grid-row: 1;

-ms-grid-column: 1;

}

**(css6)** Style the description

.desc {

-ms-grid-row: 2;

-ms-grid-column: 1;

padding-left: 25px;

}

**(css7)** Style the description

.pic {

-ms-grid-row-span: 2;

-ms-grid-column: 2;

width: 100px;

}

**(css8)** Add the media queries

@media screen and (-ms-view-state: fullscreen-landscape) {

}

@media screen and (-ms-view-state: filled) {

body {

background-color: dimgrey;

}

}

@media screen and (-ms-view-state: snapped) {

body {

background-color: blue;

display: -ms-grid;

-ms-grid-columns: 5px 1fr 5px;

-ms-grid-rows: 120px 1fr 120px;

}

}

@media screen and (-ms-view-state: fullscreen-portrait) {

}

)

1. **Open in Blend**
2. **Show off interactive design**
3. **Show Debugging**
4. **Live DOM Explorer**
5. **JavaScipt Console**
6. **Remote Debugging**
7. **Show Simulator**

* Run
* Talk about what is going on with WWAHOST
* Debugger
  + Open another instance of VS on the same solution
  + Attach to the WWAHOST – explain what WWAHOST
    - Will have to change debugger to managed code
* Open in Blend
  + Interactive mode
  + Change to grid mode and clean up