

# Client-Side Rendering (CSR) for SharePoint: from Intro to Practical Application

Wes Preston

# Abstract

- Audience: Info Workers, Dev
- SharePoint in O365, 2013, and now 2016 continue to offer stellar functionality through lists, libraries and views. Client-Side Rendering (CSR) allows power users to employ JavaScript and HTML to extend these capabilities beyond out of the box limitations without using managed code.
- This session will walk through a primer for using CSR while showing methods to improve a common SharePoint solution scenario.

# Wes Preston

- Owner / Principal Consultant - TrecStone
  - Based in Minneapolis, MN
  - MVP – SharePoint Server (2009 - )
  - SharePointing since 2003
- Contact:
  - [www.idubbs.com/blog](http://www.idubbs.com/blog)
  -  @idubbs



# Agenda

- Overview
- The Basics
- Examples and Templates
- Next steps

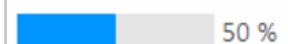
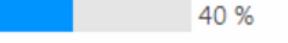
# Why? Use Cases

- Improve user experience
- Need more control over formatting and display
- SharePoint Designer Design View
- Changes without deploying managed code
- Build more ‘web-user-friendly’ environments

# Tasks

[+ new task or edit this list](#)

All Tasks	Calendar	Sample1	...	Find an item	🔍
✓	Task Name	Due Date	Assigned To	Priority	
Task1 *	...	October 05	<input type="checkbox"/> Muawiyah Shannak	(1) High	
Task2 *	...	September 10	<input type="checkbox"/> Muawiyah Shannak	(2) Normal	
Task3 *	...	September 28	<input type="checkbox"/> Muawiyah Shannak	(3) Low	
Task4 *	...	October 05	<input type="checkbox"/> Muawiyah Shannak	(1) High	
Task5 *	...	October 19	<input type="checkbox"/> Muawiyah Shannak	(3) Low	

 Name		
 Doc1 	...	
 Doc2 	...	
 Doc3	...	
 Age	22	
<b>% Complete</b>		
 59 %		
 40 %		
 100 %		
 26 %		
 50 %		
<b>% Complete</b>	 40 %	
<b>% Complete</b>	 63%	
Email	test	
		Invalid email address

# Availability

- SharePoint 2013
- SharePoint 2016\*
- Office 365 / SharePoint Online\*

## Notes

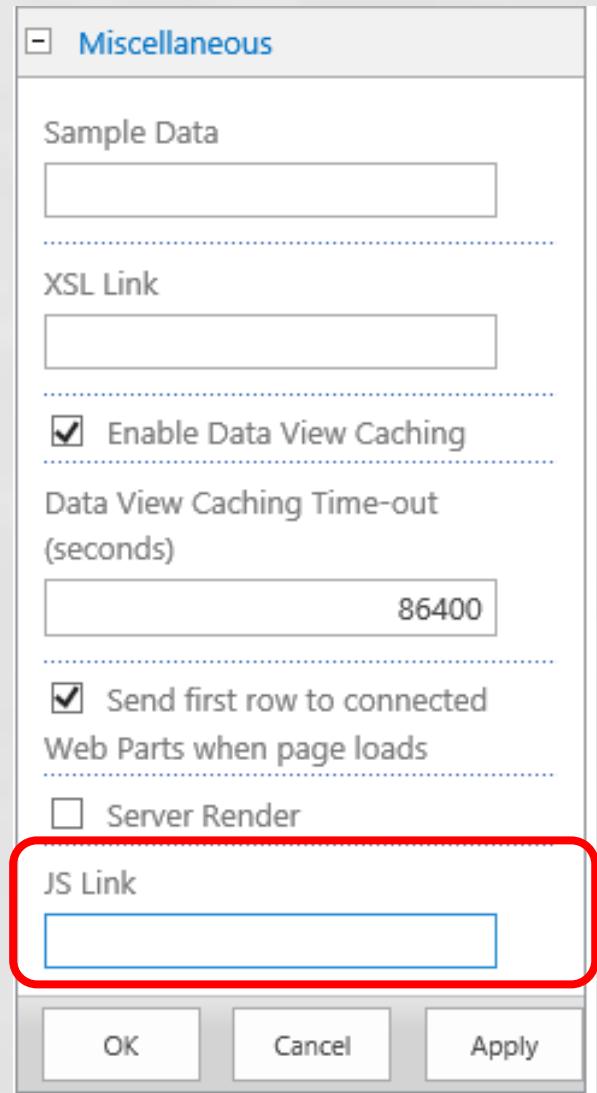
- New list and library experience vs. ‘Classic’ mode
- PowerApps

# What is ‘Client-Side Rendering’?

- Changing the way list data is *displayed*
- Does NOT change the data
- Same method SharePoint uses to display lists and pages
- JS Link property
- Code.

# The Basics

- An app -> A view
- Create a document library:  
‘Scripts’
- JavaScript file
- JS Link property
  - ~site/scripts/[ScriptName].js



# Override Types and Techniques

- Fields
- Items
- Header
- Footer
- Group
- PreRender – Modify client-side data before it is rendered
- Overrides – Overriding how different components of the Fields or View will be displayed
- PostRender – Traditional DOM manipulation

# CSR Implementation Options

- Deploy as an App (Dev)
- Deploy as a solution (Dev)
- Deploy manually and configure JS Link ←
- Choose the implementation approach that best fits your needs.
- “It depends”...

# Notes

- Internal column names
- Adding columns using the edit grid
- Master Page Gallery
  - Don't need to set metadata properties
  - In fact – better to store scripts in a 'Scripts' library
- SharePoint Designer

# Troubleshooting

- If the JavaScript code is broken, the override won't work...
- Use JavaScript 'alert'
- Check the column name
- Turn off MDS feature
- Start with a single List View on a page

# List Column content

## Demo

# Sample: List Content

- Most common change made with CSR
  - Formatting
- ‘Tweak’ Functionality
- Each column type is different
  - Start with Text columns

## Templates:

Files to start from for these examples

- CSR\_TextInline.js
- CSR\_TextString.js
- CSR\_TextArray.js

Inline:

```
'Title':{'View':'<b><#=ctx.CurrentItem.Title#></b>'}
```

```
15 function theLinkTitle(ctx) {  
16     var a = "<b>" + ctx.CurrentItem.Title + " by building an HTML string" + "</b>";  
17     return a;  
18 }  
19  
20 }
```

### TextFormatting

✓ Title

**Some text for title...**

### TextFormatting

✓ Title

**Some text for title...**

### TextFormatting

✓ Title

~~Some text for title...~~

# Field Override

```
(function () {
    var overrideCtx = {};
    overrideCtx.Templates = {};
    overrideCtx.Templates.Fields = {
        'Status': {'View':
            '<b><#=ctx.CurrentItem.Status#></b>'}
    };
});
```

```
SPClientTemplates.TemplateManager.RegisterTemplateOverrides(overrideCtx);
})();
```

TextFormatting

✓ Title

Some text for title...

# Conditional Formatting

*Demo*

# Sample: Conditional Formatting

- Functionality lost with SPD change
- Requires more JavaScript, but worth the effort
- Different column types require different handling...

## Templates:

Files to start from for these examples

- CSR\_CondFormatField.js

Blog post:  
JS Link – CSR View Conditional Formatting

```
12 function ConditionalStatus(ctx) {
13     var ret;
14     |
15     if (ctx.CurrentItem.Status == "Active" ) {
16         ret = "<b>" + ctx.CurrentItem.Status + "</b>";
17     }
18     else {
19         ret = ctx.CurrentItem.Status;
20     }
21
22     return ret;
23 }
```

## Conditional Format - Individual Fields

### Office Issues

 new item or [edit](#) this list

✓	Issue ID	Title	Assigned To	Issue Status	Priority	Due Date
	1	Can't print	... <input type="checkbox"/> Wesley Preston	<b>Active</b>	<b>(1) High</b>	
	2	Carpet is dirty	... <input type="checkbox"/> Raymond Mitchell	Closed	(3) Low	

# Field Override

```
overrideCtx.Templates.Fields = {  
    'Status': {'View': ConditionalStatus},  
    'Priority': {'View': ConditionalPriority}  
};
```

function ConditionalStatus...

# Field Override

```
function ConditionalStatus(ctx) {  
    var ret;  
    if (ctx.CurrentItem.Status == "Active" ) {  
        ret = "<b>" + ctx.CurrentItem.Status +  
        "</b>";  
    }  
    else {ret = ctx.CurrentItem.Status;}  
    return ret; //HTML String  
}
```

# Item Override

*Demo*

# Sample: Item Override

- More control, more work
- You're building out *all* the data and formatting

DemoAnnouncements

Welcome!

by  Wesley Preston

Welcome to the CSR and JS Link demonstration page.

 Add new announcement

# Item Override

```
(function () {
    var overrideCtx = {};
    overrideCtx.Templates = {};
        overrideCtx.Templates.Item = CustomItem;
        overrideCtx.Templates.Header = "<b>More of a
'digest' form of output</b><br><br>";
        overrideCtx.Templates.Footer = "<br>";
    SPClientTemplates.TemplateManager.RegisterTemplate
    Overrides(overrideCtx);
})();
```

# Item Override

```
function CustomItem(ctx) {  
    var ret = "Issue Title: <b>" +  
        ctx.CurrentItem.Title + "</b>" ;  
    ret += "<br>Created Date: " +  
        ctx.CurrentItem.Created ;  
    ret += "<br>Assigned To: " +  
        ctx.CurrentItem["AssignedTo"][0].title +  
    "</br>";  
    ret += "<br>" + ctx.CurrentItem.Comment ;  
    return ret;  
}
```

## Templates:

Files to start from for these examples

- CSR\_Item.js

## Item Override

### Office Issues

#### More of a 'digest' form of output

Issue Title: **Can't print**

Created Date: 2/18/2014 8:40 PM

Assigned To: Wesley Preston

I can't print from my iPad

Issue Title: **Carpet is dirty**

Created Date: 2/18/2014 8:40 PM

Assigned To: Raymond Mitchell

Issue Title: **Reboot didn't fix the problem**

Created Date: 3/2/2014 7:57 PM

Assigned To: Raymond Mitchell

## DemoList

My view in list form...

- 1 - Terrible demo content...
- 2 - Random blathering
- 3 - Cloudy day
- 4 - A Duck
- 5 - That important stuff

## My Lunches

Click item to see or edit meal and gift choices

SPTest2 4/22/2015 1:00 PM

Robyn Manecke 4/29/2015 1:00 PM

# Header Text

## Demo

# Sample: Column Header Text

- Utility use case— Better view usability
- Columns typically named like database columns and may not be user friendly

# Sample: The crude method

## Templates:

Files to start from for these examples

- CSR\_ColumnHeader.js

We'll call this our 'v1' solution...

```
1 (function () {
2
3 // Initialize the variables for overrides objects
4 var overrideCtx = {};
5 overrideCtx.Templates = {};
6
7 // alert("Override call worked");
8
9 overrideCtx.Templates.OnPreRender = csrHeaderText;
10
11 SPClientTemplates.TemplateManager.RegisterTemplateOverrides(overrideCtx);
12 })();
13
14 function csrHeaderText(ctx) {
15 // Hide a header to make the page cleaner looking
16 ctx.ListSchema.Field[0].DisplayName="New Name";
17
18 // Rather than 'content type' we can use more user-friendly 'Type of Document'
19 ctx.ListSchema.Field[2].DisplayName="Type of Document";
20
21 }
22 }
```

## TextFormatting

✓ New Name Title Type of Document Created Modified

There are no items to show in this view of the "TextFormatting" list.

# Sample: The better approach

## Templates:

File to start with for this example:

- CSR\_ColumnHeader2.js

```
14 function csrHeaderText(ctx) {
15 //  Specifying which header you are replacing is based on the displayed field name.
16 //  As written this will work across multiple web parts on the same page
17
18     var numColumns = ctx.ListSchema.Field.length;
19     var columnArr = ctx.ListSchema.Field;
20     for (var i = 0; i < numColumns; i++)
21     {
22         if (columnArr[i].RealFieldName == "Created") //example 1
23         {
24             columnArr[i].DisplayName = "Date created";
25         }
26         else if (columnArr[i].RealFieldName == "Title") //example 2
27         {
28             columnArr[i].DisplayName = "Not Title";
29         }
30     }
31 }
```

TextFormatting					
✓	ID	Not Title	Content Type	Date created	Modified
2	This is some text	Item		April 21	April 21

# App Part Default Text

*Demo*

# Sample: Default App Part Text

- Utility use case
- When building dashboards, may be better user experience to have a specific message, or even action available when the view is empty

## Templates:

Files to start from for these examples

- CSR\_EmptyList.js

The image displays two screenshots of SharePoint lists. Both screenshots show a header with a blue 'new item' button, a 'Find an item' search bar, and a 'All Items' link. Below the header, there is a column labeled 'Title' with a checkmark icon.

The top screenshot shows a message in a red-bordered box: "There are no items to show in this view of the 'DefaultList' list."

The bottom screenshot shows a message in a red-bordered box: "No expense reports need to be approved at this time."

```
(function () {

// Initialize the variables for overrides objects
var overrideCtx = {};
overrideCtx.Templates = {};

// alert("Override call worked");

overrideCtx.Templates.OnPreRender = csrNoListItem;

SPClientTemplates.TemplateManager.RegisterTemplateOverrides(overrideCtx);
})();

function csrNoListItem(ctx) {
    ctx.ListSchema.NoListItem = "No expense reports need to be approved at this time.";
}
```

# Unlink Lookup Field

*Demo*

# Sample: Remove link from Lookup

- User experience
- When building views or dashboards we try to keep the view clear and concise for users. A link usually means “more info here”. The default Lookup functionality is a little deceiving and usually doesn’t provide much value.

The screenshot shows a SharePoint interface. On the left, a list titled "LEGO Sets" is displayed with items like "Power Crane", "Imperial Shuttle", "Christmas Train", "Ghost Busters", and "Rencore Truck", each associated with a theme (Technic, Star Wars, Holiday, Ideas). On the right, a detailed view of an item titled "LEGOThemes - Holiday" is shown. The item has a description "Seasonal sets". Below the item details, a note indicates it was created by Wes Preston at 10/23/2015 12:52:00 AM and last modified by Wes Preston at 10/23/2015 1:08:00 AM. A "Close" button is visible in the bottom right corner of the dialog.

LEGO Sets

+ new item or edit this list

Title	Theme
Power Crane	... Technic
Imperial Shuttle	... Star Wars
Christmas Train	... Holiday
Ghost Busters	... Ideas
Rencore Truck	... Technic

LEGOThemes - Holiday

VIEW

Actions
<a href="#">Edit Item</a>
<a href="#">Version History</a>
<a href="#">Shared With</a>
<a href="#">Delete Item</a>
<a href="#">Manage</a>
<a href="#">Workflows</a>

Title: Holiday  
Desc: Seasonal sets

Created at 10/23/2015 12:52:00 AM by Wes Preston  
Last modified at 10/23/2015 1:08:00 AM by Wes Preston

Close

## Templates:

Files to start from for these examples

- CSR\_ClearLookup.js

Key is retaining the value of the Lookup field:

ctx.CurrentItem.Theme[0].lookupValue;

### LEGO Sets

[+ new item or edit this list](#)

✓ Title

Power Crane \*

... Technic

Imperial Shuttle \*

... Star Wars

Christmas Train \*

... Holiday

Ghost Busters \*

... Ideas

```
9     overrideCtx.Templates.Fields = {  
10         'Theme': {'View': csrNoThemeLink}  
11     };  
12  
13     SPCClientTemplates.TemplateManager.RegisterTemplateOverrides(overrideCtx);  
14 })();  
15  
16 function csrNoThemeLink(ctx) {  
17     //Theme.length would be the number of items if the field had more than one value  
18     var a = ctx.CurrentItem.Theme[0].lookupValue;  
19     return a;  
20 }|
```

# Sample: Lookup Link to Filtered View

## Templates:

Files to start from for these examples

- CSR\_LookupSelfFilter.js  
(no blog post yet)

Key is retaining the value of the Lookup field:  
ctx.CurrentItem.Theme[0].lookupValue;

LEGO Sets

[+ new item or edit this list](#)

✓ Title	Theme
Power Crane	... <a href="#">Technic</a>
Imperial Shuttle	... <a href="#">Star Wars</a>
Christmas Train	... <a href="#">Holiday</a>
Ghost Busters	... <a href="#">Ideas</a>
✓ Rencore Truck	... <a href="#">Technic</a>

LEGO Sets

[+ new item or edit this list](#)

✓ Title	Theme
Power Crane	... <a href="#">Technic</a>
Rencore Truck	... <a href="#">Technic</a>

# Link to Forms

# Demo

# Sample: Custom Links

- Forms, filtered pages, etc.
- More ‘web-like’ experience for users
- Direct links to non-standard forms (New, Display, Edit)
- Control navigation after forms
  - By default, SharePoint forms return to the default view page for a list



## Templates:

Files to start from for these examples

- CSR\_CustomLinks.js

DemoList				
<a href="#">+ new item or edit this list</a>				
✓	ID	Title	Title	ACategory
	1	Terrible demo content...	Terrible demo content... 	... <a href="#">Cool Category 1</a>
	2	Random blathering	Random blathering 	... <a href="#">Mediocre Topic</a>
	3	Cloudy day	Cloudy day 	... <a href="#">Terrible Content</a>
	4	A Duck	A Duck 	... <a href="#">Cool Category 1</a>
	5	That important stuff	That important stuff 	... <a href="#">Mediocre Topic</a>

TheCSR1

[Edit](#) | [Details](#)

Group by

*Demo*

# Sample: Group By

- Cleaner view
- Must rebuild the Item as well

## Templates:

Files to start from for these examples

- CSR\_Groups.js

ACategory : Cool Category 1 (2)			
1	Terrible demo content...	Terrible demo content...	... Cool Category 1
✓ 4	A Duck	A Duck	... Cool Category 1
ACategory : Mediocre Topic (2)			
2	Random blathering	Random blathering	... Mediocre Topic
5	That important stuff	That important stuff	... Mediocre Topic
ACategory : Terrible Content (1)			
3	Cloudy day	Cloudy day	... Terrible Content

## DemoList

My group by view in list form...

### Cool Category 1

- 1 - Terrible demo content...
- 4 - A Duck

### Mediocre Topic

- 2 - Random blathering
- 5 - That important stuff

### Terrible Content

- 3 - Cloudy day

# Quick Sample: View Attachments

- More direct access to attachments
- Better user experience
- Template: CSR\_ViewAttach.js

TestTasks

[⊕ new task or edit this list](#)

	✓	Task Name	Due Date	Assigned To	📎 Description
	No	Figure this out	...		<a href="#">WP_000062.jpg</a>
	No	Or this	...		<a href="#">WP_000095.jpg</a> <a href="#">WP_000096.jpg</a>
	No	nothing to show	...		

# Hide Columns

# Demo

# Sample: Hide Column

- Utility use case
- May need the column data in a view (sorting, group by, calculated fields, connected web parts, etc.) but don't want to display it
- Uses jQuery

## Templates:

Files to start from for these examples

- CSR\_HideColumn.js

```
1 (function () {
2
3 // Initialize the variables for overrides objects
4 var overrideCtx = {};
5 overrideCtx.Templates = {};
6
7 // alert("Override call worked");
8
9 overrideCtx.Templates.OnPostRender = HideColumn;
10
11 SPClientTemplates.TemplateManager.RegisterTemplateOverrides(overrideCtx);
12 })();
13
14 function HideColumn(ctx) {
15 // Similar to changing the header text, hiding columns (and column headers) is based on view columns
16
17 // Hiding the 'ID' column and header. ID is needed for other CSR fields, but we don't want it displayed.
18 $('td.ms-vb2:nth-child(3),th.ms-vh2:nth-child(3)').hide();
19 }
```

Note: Sample template will affect ALL web parts on the page

# Sample: Hide using Utility function

- Improved approach
  - Works with multiple List Views on a page
  - Uses jQuery
  - Simple to use:
    - `HideColumnFromWebPart("WebPartTitle", "Column Display Name")`
  - Still executes in PostRender because of timing

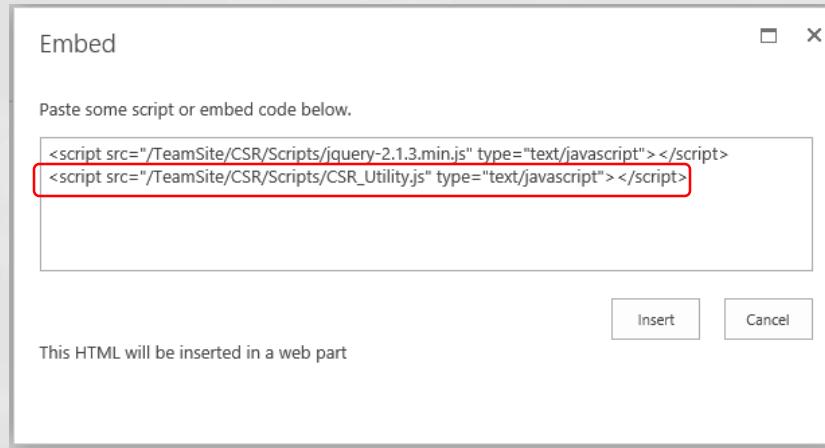
# Sample: Hide Column using Utility

## Templates:

File to start with for this example:

- CSR.Utility.js
- CSR.HideColumnUtility.js

```
function HideColumn(ctx) {  
    HideColumnFromWebPart("DemoList", "TheCSR1");  
}
```



```
1   function HideColumnFromWebPart(webPartTitle, columnDisplayName) {  
2  
3       var wpTitle = $('.js-webpart-titleCell[title="' + webPartTitle + '"]');  
16  
17       // get the WPQ#  
18       var wpqNum = wpTitle[0].id.replace('WebPartTitle', '');  
19  
20       // find the column with the display name  
21       var columnHeader = $('div #WebPart' + wpqNum).find('.ms-vh-div[displayname="' + columnDisplayName + '"]');  
22  
23  
30       var columnIndex = $(columnHeader[0]).parent().prevAll().length;  
31  
32       // hide the column (nth-child uses 1-based index...)  
33       $('div #WebPart' + wpqNum + ' .ms-listviewtable th:nth-child(' + (columnIndex + 1) + ')').hide();  
34       $('div #WebPart' + wpqNum + ' .ms-listviewtable td:nth-child(' + (columnIndex + 1) + ')').hide();  
35  
36  
37  
38 }
```

# Image Rotator

*Demo*

# Sample: SPSTC Sponsor Rotator

- SharePoint Library with sponsor logos
- ‘Paid’ field (yes/no)
- CSR with Item override that outputs all images
- Combined with PostRender to rotate the images

## Templates:

Files to start from for these examples

- CSR\_AdRotate1.js

```
function theItem(ctx) {  
    var a = [];  
    a.push("<img src='");  
    a.push(ctx.CurrentItem.Logo);  
    a.push("/>");  
    return a.join("");  
}
```

```
function csrPostRender(ctx) {  
  
    $('#fader img:not(:first)').hide();  
    $('#fader img').css('position', 'absolute');  
    $('#fader img').css('top', '0px');  
    $('#fader img').css('left', '50%');  
    $('#fader img').each(function() {  
        var img = $(this);  
        $('').attr('src', $(this).attr('src')).load(function() {  
            img.css('margin-left', -this.width / 2 + 'px');  
        });  
    });  
  
    var rotate = setInterval(doRotate, 2500);  
}
```



# Notes

- ‘Good’ practices...
- Work with IT on rules/guidelines
  - Where to store scripts
  - When to engage developers
- Governance

# Notes: Tools

- Use F12 Developer tools
  - Investigating the DOM (select element)
  - Find hooks for use in pre/post render
  - General troubleshooting

# Next Steps for IW

- Dig in to languages:
  - JavaScript
  - HTML
  - CSS
- Other session (tomorrow) on building web-user-focused solutions
- Check out PowerApps

# Next Steps for Dev

- Look at the alternate JSLink approach – extending or customized list definitions
- Reference links...

[http://www.sharepointnutsandbolts.com/2013/01/using-jslink-to-change-ui-of-sharepoint\\_20.html](http://www.sharepointnutsandbolts.com/2013/01/using-jslink-to-change-ui-of-sharepoint_20.html)

- What will you use CSR for?
- Where should you \*not\* use CSR?

# References

- Blog reference page  
<http://www.idubbs.com/blog/js-link-and-csr/>
- JS Link targeting web parts  
<http://www.n8d.at/blog/bind-jslink-overrides-to-certain-web-parts-only/>
- Modify Forms Using Content Types – Sarah Haase  
<http://blog.splibrarian.com/2011/03/21/using-content-types-to-modify-the-newform-aspx-and-editform-aspx-pages/>

# References

- jQuery and JS Link

<http://prasadpathak.wordpress.com/category/microsoft-sharepoint/>

- MDS

<http://blogs.msdn.com/b/sridhara/archive/2013/02/08/register-csr-override-on-mds-enabled-sharepoint-2013-site.aspx>

- List Type Reference (MS)

[http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.client.listtemplatetype\(v=office.15\).aspx](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.client.listtemplatetype(v=office.15).aspx)

- More Microsoft Examples

<https://code.msdn.microsoft.com/office/Client-side-rendering-JS-2ed3538a>

Session tomorrow: PWR202  
Building web-user-friendly  
solutions on SharePoint...

11:20 - same room

# Thank you for attending!

Wes Preston  
@idubbs