ЗМ						Jay.N	/lanaloto 👻
BM Community					Se	arch	٩
Profiles - Communities -	Apps -					Share	0
Wikis					I This Wiki	 Search 	Q
IBM TRIRIGA					F	ollowing Actions – 🛛 V	Viki Actions -
 > TRIRIGA Wiki Home > Facilities Management Facilities Maintenance > Environmental & Ener > Real Estate Management > Capital Project Manag > CAD Integrator Public 	You are in: IBM TRIRIGA > U Extending UX Like Updated October Edit Page Actions - UX Framework	X Framework > UX App Buil 10, 2018 by Jay.Manaloto Tag	ding > Extending UX gs: <i>None</i> Add tags UX	UX App	UX Post Presting		
 IBM TRIRIGA Connect IBM TRIRIGA Anywhere IBM TRIRIGA Applicati Release Notes 	See the <u>UX Article 3</u> "Ex user experience. But for Extending UX: Add	tending UX" PDF for pre simplicity, I'll refer to the	evious versions of this con TRIRIGA UX framework	tent. What is UX? The s as "UX".	tandard definition of "	UX" is	
 Kelease Notes Media Library Best Practices Upgrading Troubleshooting UX Framework UX Articles UX App Building Introducing UX 	STILL HUNGRY? If you' and built a simple UX ap ways to manipulate reco • What are the UX r • What are the basi • Can we add other • Can we display di • Can we display ar • Can we explore m • Still want more? What are the UX m	re ready for a third servin plication. This time, we'l rds. model and view compon- c steps to build a UX app types of fields and butto alogs and other messag nd modify existing record fore advanced functions?	ng, I admire your appetite I extend our simple applic ents? plication? ons? es? es? ds? ponents?	! In my <u>first two article</u> ation with new fields, bu	<u>s</u> , we explored the cor ttons, dialogs, toasts,	ncepts and	
Implementing UX Extending UX Implementing UX (Extending UX (Poly Converting UX to P Bundling UX (Poly Commanding UX (To refresh our memories look like this.	, if we redraw the basic Data Sc	MVC diagram with our de Model ources, DS Fields,	DS Actions	bach, our UX framewo	rk might	
 UX Perceptive Apps UX in Foundation To UX App Designer Tools UX Best Practices UX in Foundation Docs UX Component Docs 	Vie HTML Compone	ents, Elements		Wanipula C Workflov	ontroller vs, State Familie	es	
 UX Tips & Tricks UX Videos UX Archives New Page Index Index		Displays		U	ses		

Trash

Members

lags

Find a Tag

analysis application availability_section best_practices cad change_management

changes compare

compare_revisions

customizations customize database db2 exchange find_available_times gantt_chart gantt_scheduler group

memory_footprint modifications modify object_label

object_revision

operating_system oracle performance platform problem_determination reports reserve reserve_performance revision revisioning single_sign-on snapshot space sql_server **sso** support system system_performance tags: track_customizations

tririga troubleshoot tuning

upgrade ux version versioning

Cloud List

Model components.

If you remember, this is where you can define your models in whatever way you see fit to fulfill your business needs. First, you must define your models before you can develop your views.

User

Each model can be made up of the following components:

- Data Sources.
 - Child Data Sources.
 - Related Data Sources.
- Data Source Fields.
- Data Source Actions.

View components.

After your models are in place, this is where you can design your views in whatever way you require to satisfy your business scenarios. Even better, you're free to design any number of views for each model.

Each view is made up of one or more HTML files. In turn, each HTML file can be made up of the following components:

- TRIRIGA components.
- Custom components.
- · Polymer elements.
- Traditional elements.

Members

2

What are the basic steps to build a UX application?

If you also remember, this is where *you* built a simple 3-field 3-button application by (1) defining a **model** with a single data source, (2) defining the view connections to a model-and-view and application, and (3) defining and designing a **view** with a single HTML file.

Here are the basic steps:

- Define your model.
 - Optional: Add the business object.
 - 1: Add the model.
 - 2: Add the data source.
 - 3: Add a few fields for your data source.
- Define your view connections.
 - 4: Add the view.
 - 5: Add the model-and-view.
 - 6: Add the application for your model-and-view.
- <u>Define your view.</u>
 - 7: Set up the view sync.
 - 8: Add the HTML file for your view.
 - 9: Access the application.

<u>Design your view.</u>

- 10: Start the view sync.
- 11: Add a paragraph element to your HTML file.
- 12: Add a few field elements to your HTML file.
- 13: Add a few button elements to your HTML file.

Contains

triAreaNU

triDescriptionTX

Can we add other types of fields and buttons?

Sure! At this point, you should have a better idea of the application building process. For our exercise, prepare your model with a data source that contains the following field types: **Text** (like **triDescriptionTX**), **Number** (like **triAreaNU**), and **Boolean** (like **triReservableBL**).

In our example, we'll name the model jayUXBOModel2 and name the data source jayUXBODataSource2.

Data Source Metadata.

Contains

triAreaNU

otionTX

- 🔄

E StriDer

Data Source Metadata:				Pri	nt 🕜 Help
General System Associations				-	reate x
Serveral System Associations					
- General					
Model					
* Name	jayUXBODataS	ource2		ID	
Exposed Name	jayUXBODataS	ource2			
Description					
Data Source Type	BUSINESS_OBJ	ECT			
Multiple Records					
Module	jayUX	Y	Business Obje	ect jayUXBO (jayUX	BO) 🔻
List Type Name			UOM Type Na	me	
We'll add the fields to the data source by using Quick Add. Data Source Fields.					
Fields				Add Quick Add Re	emove <u>Copy</u>
Export 3 total found Apply Filters	Clear Filters			Show:	50 🔻
Name Expose	ed Name	Field Name	Data Type	Read Only?	Disable

					•	
	triReservableBL	triReservableBL	triReservableBL	BOOLEAN		
Similarly, name the	we'll name the view ja application jayUXBO	AVUXBOView2 (and ja App2 with the label Ja	y-uxbo-view2), name y UX BO Application	the model-and-view j 2 .	ayUXBOModelAn	dView2, and
Next, afte addview uxbo-vie	er you've prepared you command if needed, a w2 to add a new HTM	rr model and view conn and run the <mark>sync</mark> comn L view file.	lections, open the con nand to sync your cha	nmand prompt in your nges. In our example,	selected folder, run we'll run addview	n the with jay-
WebVie	ewSync > Add Vie	ew and Sync.				
C:\tri	riga_ux\ux_ser	ver>java -jar W	NebViewSync.ja	r addview -v ja	y-uxbo-view2	<u>2</u> -s
C:\tri Waitin	riga_ux\ux_ser g for changes	ver≻java -jar ¥ to sync	WebViewSync.ja	r sync -a		

Contains

triAreaNU

triDescriptionTX

Contains

NUMBER

STRING

Y

Contains

If needed, add the k> tag at the top to import the TRIRIGA triplat-ds (data source) component and add the <triplat-ds> tag to declare it.

Add a text area field to your HTML file.

This time, we'll add the Polymer **<paper-textarea>** tag for a multi-line text field based on the material design language by Google. If you have any questions about Polymer, its concepts, or its elements, feel free to check out the Polymer website at www.polymer-project.org.

First, add the <link> tag at the top to import the Polymer element: <link rel="import" href="../paper-input/paper-textarea.html">

HTML File > Import paper-textarea.



- 4 <link rel="import" href="../paper-material/paper-material.html">
 - <link rel="import" href="../paper-input/paper-textarea.html">

Next, add the <paper-textarea> tag: <paper-textarea label="Description" floating-label value="{{data.triDescriptionTX}}"> </paper-textarea>

HTML File > Declare paper-textarea.

15	<template></template>
16	
17	<triplat-ds data="{{data}}" id="model" name="jayUXBODataSource2"> </triplat-ds>
18	
19	<pre><div class="layout horizontal center-justified"></div></pre>
20	<pre><pre>paper-material z="1"></pre></pre>
21	Starter View for jay-uxbo-view2
22	
23	Hello World! This is my 2nd UX view!
24	
25	<pre></pre>
26	
27	<div></div>
28	<pre><pre><pre>paper-textarea label="Description" floating-label value="</pre></pre></pre>
	{{data.triDescriptionTX}}">
29	
30	

Save the file and refresh the UX view. Do you see your field? If you do, why not type a few lines? Notice how the field expands automatically?

UX App > Starter View.

Starter View for jay-uxbo-view2

Hello World! This is my 2nd UX view!

Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In ornare ultricies hendrerit. Duis vel dictum elit. Phasellus dignissim risus ac gravida aliquam. Proin mollis vehicula neque id posuere. Pellentesque consectetur ex eget pulvinar euismod. Sed blandit magna augue, eu pharetra risus posuere ac. Cras semper vitae lorem eu tristique. Cras gravida cursus pellentesque. Praesent odio orci, viverra ut blandit eget, convallis sed ante. Etiam eleifend, orci porttitor lacinia euismod, nisi massa lobortis sapien, ut pellentesque augue nisi non massa.

Add a number field to your HTML file.

This time, we'll add the Polymer **<paper-input>** tag for a 5-decimal-place number field based on the material design language by Google. Like before, make sure the **sync** command is running in the command prompt.

First, add the <link> tag at the top to import the Polymer element: <link rel="import" href="../paper-input/paper-input.html">

HTML File > Import paper-input.

1 link rel="import" href="../triplat-view-behavior/triplat-viewbehavior_bt=l"

```
behavior.html">
2 <link rel="import" href="../triplat-ds/triplat-ds.html">
3
4 <link rel="import" href="../paper-material/paper-material.html">
5 <link rel="import" href="../paper-input/paper-textarea.html">
6 <link rel="import" href="../paper-input/paper-input.html">
```

Next, add the <paper-input> tag to declare the Polymer element: <paper-input label="Decimal" floating-label auto-validate pattern="[0-9]*\.[0-9][0-9][0-9][0-9][0-9]" error-message="Invalid format for a number with 5 decimal places of precision!" value="{{data.triAreaNU}}"</pre>

HTML File > Declare paper-input.

28 <div></div>	
29 <pre><pre><pre><pre><pre>cpaper-textarea label="Description" flo</pre></pre></pre></pre></pre>	oating-label value="
{{data.triDescriptionTX}}"> <td>tarea></td>	tarea>
30	
31 <pre><pre><pre><pre>definition</pre><pre><pre>31</pre><pre><pre><pre>definition</pre><pre><pre><pre><pre><pre><pre><pre><</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>	label auto-validate
pattern="[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9])-9]" error-message="
Invalid format for a number with 5 deci	imal places of
<pre>precision!" value="{{data.triAreaNU}}"></pre>	<pre>></pre>
32	
33	

JX App > Input wi	th Error Message.	
	Starter View for jay-uxbo-view2	
	Hello World! This is my 2nd UX view!	
escription		
ecimal		
.123 valid format for a number	r with 5 decimal places of precision!	
IX App > Input wi	thout Error Message.	
	Starter View for jay-uxbo-view2	
	Hello World! This is my 2nd UX view!	
escription		
ecimal .12345		
rst, add the <link/> tag neckbox.html"> TML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="</th"/> <th>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <link href="/paper-material/paper-material.html" import"="" rel="import"/> ="import" href="/paper-material/paper-material.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-textarea.html"></th> <th></th>	ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <link href="/paper-material/paper-material.html" import"="" rel="import"/> ="import" href="/paper-material/paper-material.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-textarea.html">	
rst, add the <link/> tag neckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> 2 tink rel=	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <link href="/paper-material/paper-material.html" import"="" rel="import"/> ="import" href="/paper-material/paper-material.html"> ="import" href="/paper-input/paper-material.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-checkbox.html"> ="import" href="/paper-input/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> = "import" href="/paper-checkbox.html"> = "i</pre>	
rst, add the <link/> tag neckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> mext, add the <paper.c data.triReservableBl</paper.c 	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <link href="/paper-material/paper-material.html" import"="" rel="import"/> ="import" href="/paper-material/paper-material.html"> ="import" href="/paper-input/paper-material.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> = "import" href="/paper-checkbox/paper-checkbox.html"> = "import" href="/paper-checkbox/paper-checkbox.html"> = "import" href="/paper-checkbox/paper-checkbox.html"> = "import" href="/paper-checkbox/paper-checkbox.html"> = "import" href="/paper-checkbox/paper-checkbox.html"> = "import" href="/paper-checkbox.html"> = "import" href="/paper-checkbo</pre>	
rst, add the <link/> tag neckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> r <link rel="<br"/> r <link rel="<br"/> data.triReservableBI ITML File > Decla	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <link href="/paper-material/paper-material.html" import"="" rel="import"/> ="import" href="/paper-material/paper-material.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-checkbox.html"> ="import" href="/paper-input/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-input/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox.html"> ="import" href="/paper-checkbox.html"></pre>	="
rst, add the <link/> tag neckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> 1 TML File > Decla	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <link href="/paper-material/paper-material.html" import"="" rel="import"/> ="import" href="/paper-input/paper-material.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> = 'import" href="/paper-checkbox.html"> = 'import" href="/paper-checkbox"> = 'import" href="/paper-chec</pre>	="
rst, add the <link/> tag neckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> ext, add the <paper-c data.triReservableBI ITML File > Decla 9 0</paper-c 	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <link href="/paper-material/paper-material.html" import"="" rel="import"/> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> = 'import" href="/paper-checkbox.html"> = 'import" href="/paper-checkbox.html"> = 'import" href="/paper-checkbox.html"> = 'import" href="/paper-checkbox.html"> = 'import" href="/paper-checkbox.html"> = 'import" href="/paper-checkbox" = 'opper-checkbox.</pre>	.e =="
rst, add the <link/> tag neckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> 1 mk rel= 1 mk rel= 2 mk rel= 2 mk rel= 2 mk rel= 3 mk	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: link rel="import" href="./paper-checkbox/paper- rt paper-checkbox. ="import" href="/paper-material/paper-material.html"> ="import" href="/paper-input/paper-material.html"> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox checked=" _}]}"> = paper-checkbox. = checkbox. = checkbox. = checkbox. = checkbox. = checkbox. = checkbox. = checkbox. = checkbox. = checkbox. = checkbox checked=" { che</pre>	.e
rst, add the <link/> tag neckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> ext, add the <paper-c data.triReservableBI ITML File > Decla 9 0 1 2</paper-c 	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <link href="/paper-material/paper-material.html" import"="" rel="import"/> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> heckbox> tag to declare it: <paper-checkbox checked="
]>">Reservable</paper-checkbox> re paper-checkbox. </pre>	.e e="
rst, add the <link/> tag neckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> 1 mk rel= 1 mk rel= 2 mk rel= 2 mk rel= 3 mk rel= 3 mk rel= 3 mk rel= 3 mk rel= 4 mk rel= 5 mk rel= 5 mk rel= 7 mk	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: rimport" href="/paper-material/paper-material.html">"import" href="/paper-input/paper-textarea.html">"import" href="/paper-input/paper-input.html">"import" href="/paper-input/paper-input.html">"import" href="/paper-checkbox/paper-checkbox.html"></pre>	.e e="
rst, add the <link/> tag neckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> 8 t, add the <paper.c data.triReservableBI ITML File > Decla 9 0 1 2 3 4 5 ave the file and refrest ave the file and refrest ave the file and refrest ave the file and refrest</paper.c 	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <l< td=""><td>="" œ œ=</td></l<></pre>	="" œ œ=
rst, add the <link/> tag heckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> 8 trink rel= 9 00 1 2 3 4 5 6 ave the file and refrest ave the file ave th	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <!--</td--><td>e e="</td></pre>	e e="
rst, add the <link/> tag neckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> 1 rest, add the <paper-c data.triReservableBI ITML File > Decla 9 0 1 2 3 4 5 6 ave the file and refrest n your own? IX App > Check E</paper-c 	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <l< td=""><td>="" e=</td></l<></pre>	="" e=
rst, add the <link/> tag neckbox.html"> ITML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> ext, add the <paper-c data.triReservableBI ITML File > Decla 9 0 1 2 3 4 5 6 ave the file and refrest n your own? IX App > Check E</paper-c 	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <td>=" e=</td></pre>	=" e=
irst, add the <link/> tag heckbox.html"> ATML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> ext, add the <paper-c data.triReservableBI ITML File > Decla 9 0 1 2 3 4 5 ave the file and refrest n your own? IX App > Check E</paper-c 	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <link href="/paper-input/paper-textarea.html" import"="" rel="import"/> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-checkbox/paper-oheckbox.html"> ="import" href="/paper-checkbox/paper-oheckbox.html"> ="import" href="/paper-checkbox/paper-oheckbox.html"> = import" href="/paper-checkbox" = checkbox. = checkbox. = checkbox. = checkbox. = checkbox. = checkbox. = checkbox. = checkbox. = checkbox. = checkbox of a number with 5 decimal places of precision!" value="{data.triReservabled}!)"> = keservable = keservable = checkbox. = checkbox = checkbox = checkbox = checkbo</pre>	e e= > ta
irst, add the <link/> tag heckbox.html"> ATML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> data.triReservableBI ATML File > Decla 9 0 1 2 2 3 4 4 5 ave the file and refrest n your own? <i>IX App > Check E</i>	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <link href="/paper-input/paper-textarea.html" import"="" rel="import"/> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox.html"> ="import" href="/paper-checkbox.html"> = import" href="/paper-checkbox.html"> = import" href="/paper-checkbox" = import" href="/paper-checkbox" = import" href="/paper-checkbox" = import" href="/paper-checkbox" = import=".import" href="/paper-checkbox" = import=".import=".import=".import=".import=".import=".import=".import=".import=".import=".import=".import=".import=".import=".import=".import=".import=".import=".impo</pre>	e e=
irst, add the <link/> tag heckbox.html"> ATML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> data.triReservableBI ATML File > Decla 9 0 1 2 3 4 5 ave the file and refrest n your own? <i>IX App > Check E</i>	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <l< td=""><td>="" e=</td></l<></pre>	="" e=
irst, add the <link/> tag heckbox.html"> ATML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 6 <link rel="<br"/> 7 <link rel="<br"/> 7 <link rel="<br"/> 10 11 2 3 4 5 ave the file and refrest ave the file and refrest n your own? <i>IX App > Check E</i>	<pre>ge by Google. Like before, make sure the sync command is running. g at the top to import the Polymer element: <l< td=""><td>e e=</td></l<></pre>	e e=
irst, add the <link/> tag heckbox.html"> ATML File > Impor 4 <link rel="<br"/> 5 <link rel="<br"/> 7 <link rel="<br"/> 7 9 00 1 2 3 4 5 9 00 1 2 3 4 5	<pre>gb gcogle_Like before, make sure the sync command is running. g at the top to import the Polymer element; <link href="/paper-input/paper-textarea.html" import"="" rel="import"/> ="import" href="/paper-input/paper-textarea.html"> ="import" href="/paper-input/paper-input.html"> ="import" href="/paper-input/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox/paper-checkbox.html"> ="import" href="/paper-checkbox.html"> ="import" href="/paper-checkbox.html"> ="import" href="/paper-checkbox.html"> ="import" href="/paper-checkbox.html"> ="import" href="/paper-checkbox.html"> ="import" href="/paper-checkbox.html"> ="import" href="/paper-checkbox"> ="import" href="/paper-checkbox"> ="import" href="/paper-checkbox"> ="import" href="/paper-checkbox"> ="import" href="/paper-checkbox"> ="import" href="/paper-checkbox"> ="import" href="/paper-checkbox"> ="import" href="/paper-</pre>	="" e="

Why not? Like before, open the command prompt in your selected folder, run the **addview** command if needed, and run the **sync** command to sync your changes. In our example, we'll keep going with **jay-uxbo-view2**.

Add an action dialog to your HTML file.

This time, we'll add the Polymer **<paper-button>** tag for a button with a ripple effect and Polymer **<paper-dialog>** tag for a popup dialog box, both based on the material design language by Google. For our exercise, this dialog will be triggered from a button, and will offer two button actions.

First, add the **<link>** tags at the top to import both Polymer elements.

HTML File > Import paper-button and paper-dialog.

4	<pre><link href="/paper-material/paper-material.html" rel="import"/></pre>
5	<pre><link href="/paper-input/paper-textarea.html" rel="import"/></pre>
6	<pre><link href="/paper-input/paper-input.html" rel="import"/></pre>
7	<pre><link href="/paper-checkbox/paper-checkbox.html" rel="import"/></pre>
8	<pre><link href="/paper-button/paper-button.html" rel="import"/></pre>
9	<pre><link href="/paper-dialog/paper-dialog.html" rel="import"/></pre>

Next, add the **<paper-dialog>** tag beneath the selected **<paper-button>** tag. In this case, the **UX rocks!** button. Then, to call a JavaScript method **ontapHandler** when the button event is caught, wrap the **<section on-tap="ontapHandler">** tag around the **<paper-button>** tag.

HTML File > Declare paper-button and paper-dialog.

19	<template></template>
20	
21	<pre><triplat-ds data="{{data}}" id="model" name="jayUXBODataSource2"></triplat-ds></pre>
22	
23	<pre><div class="layout horizontal center-justified"></div></pre>
24	<pre><pre>paper-material z="1"></pre></pre>
25	Starter View for jay-uxbo-view2
26	
27	Hello World! This is my 2nd UX view!
28	
29	<pre><section on-tap="ontapHandler"></section></pre>
30	<pre><paper-button disabled="">Disabled</paper-button></pre>
31	<pre> >>paper-button raised data-dialog="actions"></pre>
	UX rocks!
32	<pre><paper-dialog id="actions"></paper-dialog></pre>
33	<h2>Confirm</h2>
34	Are you sure that UX rocks?
35	<pre><div class="buttons"></div></pre>
36	<pre><pre>constant </pre></pre>
	No
37	<pre><pre>paper-button raised dialog-confirm</pre></pre>
	autofocus>Yes
38	
39	
40	
41	<pre></pre>
42	

Then, insert the JavaScript method ontapHandler within the <script> tag.

HTML File > Insert ontapHandler.





HTML File > Import paper-toast.

- <link rel="import" href="../paper-button/paper-button.html">
- <link rel="import" href="../paper-dialog/paper-dialog.html">
- <link rel="import" href="../paper-toast/paper-toast.html">

Next, insert onclick="document.querySelector('#toast1').show()" within the selected <paper-button> tag. In this case, the Yes button. Then, add the <paper-toast> tag: <paper-toast id="toast1" text="You've confirmed that UX rocks." duration="7000" style="right:12px; left:initial;"></paper-toast>

HTML File > Declare paper-toast.

30	<pre><section on-tap="ontapHandler"></section></pre>
31	<pre><pre><pre>content</pre>disabled</pre>/paper-button></pre>
32	<pre> <pre>cbr><pre>>>paper-button raised data-dialog="actions"></pre></pre></pre>
	UX rocks!
33	<pre><paper-dialog id="actions"></paper-dialog></pre>
34	<h2>Confirm</h2>
35	Are you sure that UX rocks?
36	<pre><div class="buttons"></div></pre>
37	<pre><pre>paper-button raised dialog-dismiss></pre></pre>
	No
38	<pre><pre>paper-button raised dialog-confirm</pre></pre>
	autofocus onclick="document.
	<pre>querySelector('#toast1').show()"></pre>
	Yes
39	
40	
41	<pre><paper-toast <="" duration="7000" id="toast1" text="You've</pre></th></tr><tr><th></th><th>confirmed that UX rocks." th=""></paper-toast></pre>
	<pre>style="right:12px; left:initial;"></pre>
	toast>
42	
43	<pre></pre>
44	

Save the file and refresh the UX view. Feel free to click the **UX rocks!** button, and then the **Yes** button. Do you see your toast? Pretty sweet!

UX App > Toast Popup.



Model		
* Name	jayUXPeopleDataSource	ID
Exposed Name	jayUXPeopleDataSource	
Description		
Data Source Type Multiple Records	BUSINESS_OBJECT	
Module	triPeople	Business Object People (triPeople)

					_	T Sanche Hand T Hand	
Section 2 tot	al found Apply	Filters <u>Clear Filters</u>				Show:	50
Name		Exposed Name	Field Name	Data Type		Read Only?	Disabl
Contains	C	ontains	Contains	Contains			Contains
🔲 🛃 triFirstName	TX tri	iFirstNameTX	triFirstNameTX	STRING	T		
🔲 🌄 triLastName	eTX tri	iLastNameTX	triLastNameTX	STRING	۷		
odel and View	v Metadata. tadata:					🖶 Prir	nt 👩 He
General System	Workflow Insta	ance Associations				C	reate x
 General 							
* Name	jayUXPeopleMo	odelAnd∀iew		* ID jayUXPee	opleModel	lAndView	
Exposed Name	jayUXPeopleMo	odelAnd∀iew					
Description							
Model Name	jayUXPeopleMo	odel	P 8	View Name jayUXPe	opleView		,p e
view Type	WEB_VIEW						
plication Metada	ta:					📕 Pri	nt 😰 H4
pplication Metada General System	ta: Workflow Insta	ance Associations				🖶 Priz	nt 🕐 He reate x
General System	ta: Workflow Insta	ance Associations				Prir C	nt 🕐 He reate x
pplication Metada General System General * Name	ta: Workflow Insta jayUXPeopleAp	ance Associations		* ID jayUXPeo	pleApp	e Priz	nt 😧 He reate x
pplication Metada General System General Mame Exposed Name	ta: Workflow Insta jayUXPeopleAp jayUXPeopleAp	ance Associations pp		* ID jayUXPeo	pleApp	Fri	nt 🕑 He reate x
pplication Metada General System General * Name Exposed Name Label	ta: Workflow Insta jayUXPeopleAp jayUXPeopleAp Jay UX People	ance Associations pp Application		* ID jayUXPeo	pleApp	E Priz	nt 🕐 He reate x
pplication Metada General System General * Name Exposed Name Label Description	ta: Workflow Insta jayUXPeopleAp jayUXPeopleAp Jay UX People A	ance Associations pp Application		* ID jayUXPeo	pleApp	E Priz	nt 🕢 He
pplication Metada General System General Mame Exposed Name Label Description App Type	ta: Workflow Insta jayUXPeopleAp jayUXPeopleAp Jay UX People A WEB_MODEL_A	Application		★ ID jayUXPeo	pleApp	AndView	nt 🕐 He reate x
pplication Metada General System General * Name Exposed Name Label Description App Type Instance ID	ta: Workflow Insta jayUXPeopleAp jayUXPeopleAp Jay UX People A WEB_MODEL_A 127333408	Application		* ID jayUXPeo	pleApp	AndView	nt 🕐 He reate x
General System General System General System General Kname Exposed Name Label Description App Type Instance ID Create x	ta: Workflow Insta jayUXPeopleAp jayUXPeopleAp Jay UX People A WEB_MODEL_A 127333408	Application		* ID jayUXPeo	pleApp	AndView	nt 😢 I
General System General System General System General System Cabel Description App Type Instance ID Create x ext, after you've p Idview command cople-view to add Veb ViewSync (tririga use)	ta: Workflow Insta	Application ND_VIEW nodel and view co run the sync col view file. and Sync. and Sync.	onnections, open th mmand to sync you	* ID jayUXPeo App Name jayUXPeo e command prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. In our examination of the second prompt in yar changes. •	pleApp pleModel your sele nple, we'l	AndView	nt eate
pplication Metada General System General System General * Name Exposed Name Label Description App Type Instance ID Create x ext, after you've p dview command cople-view to add WebViewSync :\tririga_up 	ta: Workflow Insta	Application Application ND_VIEW ND_VIEW nodel and view coll rodel and sync.	onnections, open th mmand to sync you WebViewSync	 * ID jayUXPeo App Name jayUXPeo e command prompt in y ar changes. In our examt . jar addview -v . jar sync -a 	pleApp pleModel your sele nple, we'l	AndView AndView AndView AndView	nt 🕐 He reate x
pplication Metada General System General System Aname Exposed Name Label Description App Type Instance ID Create x ext, after you've p Idview command cople-view to add <i>App View Sync</i> : \tririga_up iting for of heeded, add the of declare it.	ta: Workflow Insta jayUXPeopleAp jayUXPeopleAp Jay UX People A WEB_MODEL_A 127333408 repared your m if needed, and a new HTML v > Add View x ux_serve changes to <link/> tag at th	Application ND_VIEW nodel and view coll nodel and view coll run the sync coll view file. and Sync. and Sync. ar>java - jar pr>java - jar pr>java - jar pr>java - jar pr>java - jar pr>java - jar	e TRIRIGA triplat-	 * ID jayUXPeo App Name jayUXPeo e command prompt in y ar changes. In our exam jar addview -v jar sync -a ds (data source) comption 	pleApp pleModel your sele nple, we'l	AndView	nt le with jay

This time, we'll add more Polymer **<paper-input>** tags to grab the instance data **triFirstNameTX** and **triLastNameTX** from the

existing people record.

First, add the <link> tag at the top to import the Polymer element: <link rel="import" href="../paper-input/paper-input.html">

HTML File > Import paper-input.

- <link rel="import" href="../triplat-view-behavior/triplat-viewbehavior.html">
- 2 <link rel="import" href="../triplat-ds/triplat-ds.html">
- <link rel="import" href="../paper-material/paper-material.html">
- 5 <link rel="import" href="../paper-input/paper-input.html">

Next, add the **<paper-input>** tags to declare the Polymer elements:

```
<paper-input ... value="{{data.triFirstNameTX}}"></paper-input>
```

```
<paper-input ... value="{{data.triLastNameTX}}"></paper-input>
```

HTML File > Declare paper-input.

15	<template></template>
16	
17	<pre><triplat-ds data="</pre></td></tr><tr><td></td><td>{{data}}" id="model" name="jayUXPeopleDataSource">>/triplat-ds></triplat-ds></pre>
18	
19	<pre><div class="layout horizontal center-justified"></div></pre>
20	<pre><pre>paper-material z="1"></pre></pre>
21	Starter View for jay-ux-people-view<!--<b-->b>
22	
23	Hello World! This is my 3rd UX view!
24	
25	<pre><paper-input floating-label="" label="First Name" value="</pre></td></tr><tr><td></td><td>{{data.triFirstNameTX}}"></paper-input></pre>
26	<pre><paper-input floating-label="" label="Last Name" value="</pre></td></tr><tr><td></td><td>{{data.triLastNameTX}}">>/paper-input></paper-input></pre>
27	
28	<pre></pre>
29	
30	

Save the file and refresh the UX view. Do you see your fields? Do you see the corresponding instance data from your people record? Nice, huh?

UX App > Input with Instance Data.

Starter View	/ for jay-ux-people-view
Hello World	This is my 3rd UX view!
First Name	
Jay	
Last Name	
Manufacture	er

Modify data by triggering a workflow.

For our next exercise, prepare a workflow with **Synchronous** concurrence, **Permanent** data, the **triPeople** module, and the **triPeople** business object. Add a **Modify Records** task that will map to and from the **triPeople** business object. Edit the map to modify the **triNickNameTX** (Nick Name).

In our example, we'll name the workflow **jayUX** - **Save** - **Perm** and modify the **triNickNameTX** to **Jayman**. Then **Publish** the workflow.

New Task	Modify Records	*
	Workflow Properties <u>Callers Parameters Publish</u>	Retire
	Name: jayUX - Save - Perm	

Workflow > Modify Records Task.

Lock Record For Other Users:			
Propagate Integration Status:	1		

Next, return to your data source from the last exercise, and (1) add the field **triNickNameTX**. Then, (2) add an action group, and (3) add an action that has the **WORKFLOW** action type, **triPeople** module, **triPeople** business object, and workflow name that you defined earlier.

Data Source Action Group Metadata.

Data Source Action	Group Metadata:		🖶 Print 🕜 Help
General System	Workflow Instance Associations		Create x
 General 			
Model		Data Source	
* Name	jayUXPeopleActionGroup	ID jayUXPeopleActionGrou	р
Exposed Name	jayUXPeopleActionGroup		
Description			
 Actions 			Add <u>Remove</u> <u>Copy</u>

In our example, we'll name the action group **jayUXPeopleActionGroup**, name the action **jayUXPeopleActionSavePerm**, and name the workflow **jayUX - Save - Perm**. This hooks up your workflow into your data source. Finally, **Save & Close** your action, action group, and data source.

Data Source Action Metada	ata:	📇 Print 🚷 Helj
General System Workflow Instance Associations		Create x
General		
Model		Data Source
Action Group		
* Name	jayUXPeopleActionSavePerm	ID jayUXPeopleActionSavePerm
Exposed Name	jayUXPeopleActionSavePerm	
Action Type	WORKFLOW	
Module	triPeople 💽	Business Object People (triPeople)
Workflow Name	JayUX - Save - Perm	
State Transition Name		
Description		
Description		
tton.html">	paper-button.	ei- import meipaper-buttompaper-
1 <link href="/triplat-view-k ml" rel=":</th><th><pre>import"/>	pehavior/triplat-view-	
<pre>1 <link :<br="" rel=":
behavior.htm
2 <link rel="/>3 4 <link :<br="" rel=":
5 <link rel="/>6 </pre>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa </pre>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"></pre>
<pre>1 <link 3="" 4="" 6="" :="" <link="" href="/triplat-view-k nl" rel=": </pre></td><td><pre>import"/> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-button/paper-</pre>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-button.html"></pre>	
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 5 <link rel=" td=""/> 6 <link href="/triplat-view-k ml" rel=":</td> ext, add the <paper-inputputputputputputputputputputputputputp</td><td><pre>import"/> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-button/p ut> tag to declare the Polymer element: <paper tap=""> tag to declare it and call a laveScript mot </paper>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-input.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}"></pre>	
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-input> hen, add the <paper-button on-<="" raised="" td=""></paper-button></paper-input>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-button/p ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met tap="updateActionHandler">Trigger now!</paper></pre>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-input.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tapped /paper-button></pre>
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-input< td=""> hen, add the <paper-button on-<="" raised="" td=""> TML File > Declare</paper-button></paper-input<>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-input/pa import" href="/paper-button/p ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met -tap="updateActionHandler">Trigger now!<!--/ paper-input and paper-button.</pre--></paper></pre>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-input.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tappe /paper-button></pre>
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-input> hen, add the <paper-button on-<="" raised="" td=""> TML File > Declare 1</paper-button></paper-input>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-button/p ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met tap="updateActionHandler">Trigger now!<!-- paper-input and paper-button. </paper-material z="1"--></paper></pre>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-input.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tapped /paper-button></pre>
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-input< td=""> hen, add the <paper-button on-<="" raised="" td=""> <i>TML File > Declare</i> 1 1 2 3 </paper-button></paper-input<>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-input/pa import" href="/paper-button/f ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met tap="updateActionHandler">Trigger now!<!-- paper-input and paper-button. </paper-material z="1"--> </paper></pre>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-button.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tapped paper-button></pre>
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-inpuput> hen, add the <paper-button on<="" raised="" td=""> <i>TML File > Declare</i> 1 2</paper-button></paper-inpuput>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-input/pa import" href="/paper-button/f ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met -tap="updateActionHandler">Trigger now!<!--/ paper-input and paper-button. </paper-input and paper-button. </paper-material z="1"--> Starter View for Hello World! This</paper></pre>	<pre>behavior/triplat-view- iplat-ds.html"> iplat-ds.html"> iplat-ds.html"> iplat-ds.html"> iplat-ds.html"> iplat-ds.html"> iplat-ds.html"> iplat-ds.html"> iplat-input.html"> iplat-inp</pre>
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-inputoment, <paper-button="" add="" on<="" raised="" td="" the=""> <i>TML File > Declare</i> 1 2 3 4 5</paper-inputoment,>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-input/pa import" href="/paper-button/f ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met tap="updateActionHandler">Trigger now!<!--/ paper-input and paper-button. </paper-input and paper-button. </paper-material z="1"--> Starter View for Hello World! This</paper></pre>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-button.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tappe /paper-button> > jay-ux-people-view s is my 3rd UX view!</pre>
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-inputoment, <paper-button="" add="" on-<="" raised="" td="" the=""> trmL File > Declare 1 2 3 4</paper-inputoment,>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-input/pa import" href="/paper-button/f ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met tap="updateActionHandler">Trigger now!<!--/ paper-input and paper-button. </paper-input and paper-button. </paper-material z="1"--> Starter View for Hello World! This </paper></pre>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-button.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tappe /paper-button> > jay-ux-people-view s is my 3rd UX view! 'First Name" floating-label value=' "X}) "></pre>
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-input< td=""> hen, add the <paper-button on<="" raised="" td=""> ITML File > Declare 3 4 5 6</paper-button></paper-input<>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-input/pa import" href="/paper-button/f ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met tap="updateActionHandler">Trigger now!<!--/ paper-input and paper-button. </paper-input and paper-button. </paper-material z="1"--> <paper-material z="1"> <paper-input <="" and="" paper-button.="" paper-material="" z="1"> <paper-input <="" and="" paper-button.="" pre=""></paper-input></paper-input></paper-material></paper></pre>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-button.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tappe /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-input> /paper-input> /paper-input> /paper-input></pre>
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-inputoment< td=""> behavior.htm a 4 5 <link href="/triplat-view-k ml" rel=":</td> ext, add the <paper-inputoment</td> behaver-button raised on ITML File > Declare 1 2 3 4 5 6 7 8</td><td><pre>import"/> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-input/pa import" href="/paper-button/f ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met tap="updateActionHandler">Trigger now! ton> tag to declare it and call a JavaScript met tap="updateActionHandler">Trigger now! paper-input and paper-button. </paper></paper-inputoment<>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-button.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tappe /paper-button> /paper-button> s is my 3rd UX view! 'First Name" floating-label value=" 'X})"> 'Last Name" floating-label value=" (})"> 'Nick Name" floating-label value="</pre>	
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-inputomen, <paper-button="" add="" on-<="" raised="" td="" the=""> image: button raised on- image: butt</paper-inputomen,>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-input/pa import" href="/paper-button/f ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met tap="updateActionHandler">Trigger now!<!-- paper-input and paper-button. </paper-input and paper-button. </paper-material z="1"--> Starter View for Hello World! This </paper></pre> (data.triFirstNameT)	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tappe /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-button> /paper-input> /paper-input> /paper-input> /paper-input> /paper-input> /paper-input> /paper-input> /paper-input> /paper-input></pre>
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-inpuput> hen, add the <paper-button on<="" raised="" td=""> ITML File > Declare 1 2 3 4 5 6 1 2 3 4 5 6 7 8 9</paper-button></paper-inpuput>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-input/pa import" href="/paper-button/f ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript mett tap="updateActionHandler">Trigger now!<!--/ paper-input and paper-button. </pre--> <pre> <pre> (paper-material z="1"> Starter View for Hello World! This Hello World! This <paper-input label=" {{data.triNickNameTx {adata.triNickNameTx } } }</td><td><pre>behavior/triplat-view-
iplat-ds.html" {paper-input="" {{data.trilastnametx=""> l/paper-material.html"> aper-input.html"> paper-button.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tapper /paper-button> > jay-ux-people-view</paper-input> s is my 3rd UX view! 'First Name" floating-label value=" 'X} }"> 'Last Name" floating-label value=" {} "> 'Last Name" floating-label value=" {} "> 'Nick Name" floating-label value=" {} "></pre></pre></paper></pre>	
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-input> nen, add the <paper-button on-<="" raised="" td=""> ITML File > Declare 1 5 6 1 2 3 6 1 2 3 6 7 8 9 0</paper-button></paper-input>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-input/pa import" href="/paper-button/f ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met tap="updateActionHandler">Trigger now!<!-- paper-input and paper-button. </paper-input and paper-button. </paper-material z="1"--> Starter View for Hello World! This </paper></pre>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-button.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tapper /paper-button> /paper-button> /paper-button> /First Name" floating-label value=' rX} "> /Last Name" floating-label value=" (}) "> /Last Name" floating-label value=" (}) "> /Nick Name" floating-label value=" (}) "> /Nick Name" floating-label value=" (}) "> /Nick Name" floating-label value=" (}) "> /Nick Name" floating-label value=" (}) "></pre>
1 <link :<="" rel=":</td> behavior.htm 2 <link rel=" td=""/> 3 4 <link :<="" rel=":</td> 4 <link rel=" td=""/> 5 <link :<="" rel=":</td> 6 <link rel=" td=""/> ext, add the <paper-inpuput> nen, add the <paper-button on-<="" raised="" td=""> <i>ITML File > Declare</i> 1 2 3 4 5 6 7 8 9 0 1</paper-button></paper-inpuput>	<pre>import" href="/triplat-view-k ml"> import" href="/triplat-ds/tri import" href="/paper-material import" href="/paper-input/pa import" href="/paper-input/pa import" href="/paper-button/p ut> tag to declare the Polymer element: <paper ton=""> tag to declare it and call a JavaScript met tap="updateActionHandler">Trigger now!<!-- paper-input and paper-button. </pre--> <pre> <pre></pre></pre></paper></pre>	<pre>behavior/triplat-view- iplat-ds.html"> l/paper-material.html"> aper-input.html"> paper-button.html"> paper-button.html"> r-input value="{{data.triNickNameTX}}">hod updateActionHandler when the button is tapped paper-button> paper-button> 'paper-button> 'First Name" floating-label value=' 'X}}"> 'Last Name" floating-label value=" (})"> 'Nick Name" floating-label value=" (})"></pre>

HTML File > Insert updateActionHandler.

36	<script></script>	

38		
39		<pre>is: "jay-ux-people-view",</pre>
40		
41		behaviors: [TriPlatViewBehavior],
42		
43		updateActionHandler: function() {
44		<pre>this.\$.model.updateRecord(127333408,TriPlatDs.RefreshType.</pre>
		BOTH,"jayUXPeopleActionGroup","
		jayUXPeopleActionSavePerm");
45		$\mathbf{F}_{\mathbf{r}}$
46		
47	});	
48		

Save the file and refresh the UX view. Click the **Trigger now!** button and refresh again. Do you see your modified instance data? Pretty nice!

UX App > Input with No Nick Name.

eople-view	Starter View for iav-ux-people-view
3rd UX view!	Hello World! This is my 3rd UX view!
	First Name
	Jay
	Last Name Manufacturer
	Nick Name
	Trigger now!
	UX App > Input with Modified Nick Name.
	UX App > Input with Modified Nick Name.

Hello World! This is my 3rd U	JX vie
First Name	
Jay	
Last Name	
Manufacturer	
Nick Name	
Jayman	
Triggor poul	

Can we explore more advanced functions?

Maybe a few more? At this point, you should have an even better idea of how UX applications are built and how UX views are designed with Polymer components. But what about designing a view with your own customized reusable component? Since you're already here, why not?

Start a card view as a custom component.

For our next exercise, we'll prepare a new HTML file in the same folder as your existing HTML file. **But don't move it into the same folder yet!** First, edit the new HTML file in a different folder to begin with the simplest running design. This new HTML file will start your custom component.

In our example, we'll name the new HTML file **jay-ux-people-card.html** for our card-view custom component, and edit its HTML to look like this.

HTML File > Custom Component.

1	<pre><dom-module id="jay-ux-people-card"></dom-module></pre>
2	<template></template>
3	
4	This is my card-view custom component.
5	
6	
7	
8	<script></script>



Next, after you've prepared your HTML file, open the command prompt in your selected folder, and run the **sync** command to sync your changes.

```
WebViewSync > Sync.
```

```
C:\tririga_ux\ux_server>java -jar WebViewSync.jar sync -a
Waiting for changes to sync...
```

This time, you're ready to move your new HTML file. After we move this file, and hook it up into your main view, we can expand the component.

In our example, the C:\tririga_ux\ux_server folder contains the jay-ux-people-view folder, which contains the existing jay-ux-people-view.html file from the last exercise. Go ahead and move the new HTML file jay-ux-people-card.html beside the existing HTML file in the same folder.

Notice how the WebViewSync tool detects and pushes the new HTML file.

WebViewSync > Push.

```
C:\tririga_ux\ux_server>java -jar WebViewSync.jar sync -a
Waiting for changes to sync...
Signing On To TRIRIGA [success]
[2015-09-02 20:30:29] [push] /jay-ux-people-card.html
[2015-09-02 20:30:29] [push] /jay-ux-people-card.html
[ok]
```

Add a card view component to your HTML file.

This time, we'll add a custom tag to our existing HTML file so it can grab the new custom component. In our example, we'll add a custom <jay-ux-people-card> tag to our existing jay-ux-people-view.html file.

First, add the <link> tag at the top to import the custom component: <link rel="import" href="jay-ux-people-card.html">

HTML File > Import Custom Component.

```
1 <link rel="import" href="../triplat-view-behavior/triplat-view-
behavior.html">
2 <link rel="import" href="../triplat-ds/triplat-ds.html">
3
4 <link rel="import" href="../paper-material/paper-material.html">
5 <link rel="import" href="../paper-material/paper-material.html">
6 <link rel="import" href="../paper-input/paper-input.html">
7 <link rel="import" href="../paper-button/paper-button.html">
```

Next, add the <jay-ux-people-card> tag to declare the custom component: <jay-ux-people-card></jay-ux-people-card>

HTML File > Declare Custom Component.

22	<pre><paper-material z="1"></paper-material></pre>
23	Starter View for <b jay-ux-people-view
24	
25	Hello World! This is my 3rd UX view!>
26	
27	<jay-ux-people-card>>/jay-ux-people-card></jay-ux-people-card>
28	
29	<pre><pre>cpaper-input label="First Name" floating-label value="</pre></pre>
	{{data.triFirstNameTX}}">
30	<pre><paper-input floating-label="" label="Last Name" value="</pre></td></tr><tr><td></td><td>{{data.triLastNameTX}}">>/paper-input></paper-input></pre>
31	<pre><paper-input floating-label="" label="Nick Name" value="</pre></td></tr><tr><td></td><td>{{data.triNickNameTX}}">>/paper-input></paper-input></pre>
32	
33	<pre><paper-button on-tap="updateActionHandler" raised=""></paper-button></pre>
	Trigger now!
34	
35	<pre></pre>

Save the file and refresh the UX view. Do you see your new custom component? Now that it's hooked up, we can expand it! Sweet, huh?

UX App > Custom Component.

This is my card-view custom component.	
First Name	
Jay	
Last Name	
Manufacturer	
Nick Name	
Jayman	



Expand your card view component.

This time, we'll add a simple **peopleData** property to our component **jay-ux-people-card.html** to show that we can use component properties. Like before, we'll also add instance data **triFirstNameTX** and **triLastNameTX** from an existing people record, the same record from previous exercises.

First, insert the **peopleData** property within the **<script>** tag.

HTML File > Insert peopleData.

8	<script></th><th></th></tr><tr><th>9</th><th>Polym</th><th>er ({</th></tr><tr><th>10</th><th></th><th></th></tr><tr><th>11</th><th></th><th><pre>is: "jay-ux-people-card",</pre></th></tr><tr><th>12</th><th></th><th></th></tr><tr><th>13</th><th></th><th>properties: {</th></tr><tr><th>14</th><th></th><th></th></tr><tr><th>15</th><th></th><th>peopleData: {</th></tr><tr><th>16</th><th></th><th>type: <i>Object</i>,</th></tr><tr><th>17</th><th></th><th>$\mathbf{F}_{\mathbf{r}}$</th></tr><tr><th>18</th><th></th><th>}</th></tr><tr><th>19</th><th></th><th></th></tr><tr><th>20</th><th>});</th><th></th></tr><tr><th>21</th><th></script>	

Next, replace the placeholder text with **peopleData** values and a profile image. Ideally, the image data should be pulled from an existing people record. But for now, we'll add the profile image **jay-profile.png** locally. Go ahead and move the profile image to the same **jay-ux-people-view** folder.

HTML File > Add peopleData Values and Profile Image.

1	<pre><dom-module id="jay-ux-people-card"></dom-module></pre>
2	<template></template>
3	
4	<pre><!-- <p-->This is my card-view custom component></pre>
5	
6	<pre><div class="layout vertical center"></div></pre>
7	<pre></pre>
8	<h2>{{peopleData.triFirstNameTX}}</h2>
9	< <u>span</u> >{{peopleData.triLastNameTX}} <u span>
10	
11	
12	
13	

At this point, how do you pass the model **data** values from your main view to the **peopleData** values in your custom component? Easy! In our example, we'll return to our main **jay-ux-people-view.html** file, and add the attribute **people-data="{{data}}"** to our custom **<jay-ux-people-card>** tag. This will assign **data** values to the **peopleData** property.

HTML File > Add people-data Attribute.

22	<pre><pre>paper-material z="1"></pre></pre>
23	Starter View for <b jay-ux-people-view
24	
25	Hello World! This is my 3rd UX view!>
26	
27	<jay-ux-people-card people-data="{{data}}">>/jay-ux-</jay-ux-people-card>
	people-card>
28	
29	<pre><paper-input floating-label="" label="First Name" value="</pre></td></tr><tr><td></td><td>{{data.triFirstNameTX}}">>/paper-input></paper-input></pre>
30	<pre><paper-input floating-label="" label="Last Name" value="</pre></td></tr><tr><td></td><td>{{data.triLastNameTX}}"></paper-input></pre>
31	<pre><paper-input floating-label="" label="Nick Name" value="</pre></td></tr><tr><td></td><td>{{data.triNickNameTX}}">>/paper-input></paper-input></pre>
32	
33	<pre><paper-button on-tap="updateActionHandler" raised=""></paper-button></pre>
	Trigger now!
34	
35	<pre></pre>

Why do we need a dash in the HTML tag attribute **people-data** instead of simply using **peopleData** like the component property? *Well, Polymer maps any attribute name with dashes to the corresponding property name by automatically converting attribute* dash-case to property camelCase.

Save the file and refresh the UX view. Do you see your expanded custom component? Now it's time to replace the local profile image!

UX App > Expanded Custom Component.

1 Corres
11 CENT
approximation in the
and a state of the
1111 Mar 11-
Jay Manufacturer
Jay Manufacturer First Name Jay Last Name Manufacturer
Jay Manufacturer First Name Jay Last Name Manufacturer
Jay Manufacturer First Name Jay Last Name Manufacturer Nick Name
Jay Manufacturer First Name Jay Last Name Manufacturer Nick Name Jayman

Display image data from an existing record.

Before we forget, return to your existing model and data source from the last exercise, and add the field trilmageIM (Image).

If you remember, we added the profile image **jay-profile.png** locally. This time, we'll replace it and add the TRIRIGA <**triplat-image**> tag to our component **jay-ux-people-card.html** to grab the instance data **triImageIM** from an existing people record, the same record from previous exercises.

First, add the tag at the top to import the TRIRIGA element: k rel="import" href="../triplat-image/triplat-image.html">

Next, add the <triplat-image> tag to declare the TRIRIGA element: <triplat-image src="{{peopleData.triImageIM}}"></triplat-image>

HTML File > Import and Declare triplat-image.

1	<pre><link href="/triplat-image/triplat-image.html" rel="import"/></pre>
2	
3	<pre><dom-module id="jay-ux-people-card"></dom-module></pre>
4	<template></template>
5	
6	<pre><!-- <p-->This is my card-view custom component></pre>
7	
8	<pre><div class="layout vertical center"></div></pre>
9	<pre><triplat-image src="{{peopleData.triImageIM}}"></triplat-image></pre>
	image>
10	<h2>{{peopleData.triFirstNameTX}}</h2>
11	{{peopleData.triLastNameTX}}
12	
13	
14	
15	

Save the file and refresh the UX view. Do you see your completed custom component? Now you're ready to reuse it in other applications!

UX App > Completed Custom Component.



Jay Manufacturer

First Name

Jay

Last Name

Manufacturer

Nick Name

Jayman

Trigger now!

Still want more?

If you have any questions about UX that weren't answered in this article, feel free to reach out to your IBM TRIRIGA representative or business partner. In the meantime, here are some more background questions and answers from my previous articles that might help to fill in the gaps or give you a better idea of what we're trying to do. In any case, stay tuned!

Background Q & A.

Question	Answer
How do we build applications in the new	While we can't promise any specific dates, we plan to develop a new "model designer" or "model builder" metadata construct that supports the model. Similarly, we plan to develop a new "view designer" or "view builder" metadata construct that supports the view. Fortunately, we don't need to develop a new metadata construct for the controller since
	existing workflows and state families can already serve this function. Meanwhile, if we store the new platform metadata as records that can be accessed through forms, we can more quickly react to business requirements and add features.
How do we simplify the interface or view?	Our existing technology ties forms to "things" like people and locations. So why not change the pattern so that views are tied to "actions" like creating and submitting requests? This change could be accomplished by designing views that are specific to a user role. Then we could still reuse our existing business objects and workflows to support the new role- based interfaces.
What happens to our existing customers?	Because the new views will be "bolt-on" interfaces that are "bolted onto" existing applications, customers who don't choose the new MVC framework won't be affected. But for customers who choose the new framework, results could vary depending on how new role-based interfaces are applied and how much the application is customized. Fortunately, a flexible MVC model would offer customers a more efficient customization and upgrade strategy. For example, customers could add their own business objects instead of adding fields to our shipped business objects. This scenario would be easier to track during upgrade.
	Next

 Comments (0)
 Versions (15)
 Attachments (49)
 About

There are no comments.

Add a comment

Privacy

S Feed for this page | Feed for these comments

Contact

Terms of use

Accessibility

Report abuse

Cookie Preferences