

DX Integration with HCL Volt MX for Developers



Edition: March 2024

- Herbert Hilhorst <u>herbert.hilhorst@hcl-software.com</u>
- Dai Nguyen <u>dai.nguyen@hcl-software.com</u>
- Mario D'Anna mario.danna@hcl-software.com

Table of Contents

Author(s)	3
Introduction	4
Prerequisites	5
Lab Overview	6
Part 1: Use DX Digital Assets in a new HCL Volt MX Mobile Application	9
Part 2: Use DX Content in a new HCL Volt MX Web Application	33
Part 3: Use Volt MX Foundry to Integrate External Data Sources in DX	63
Part 4: Integrate Volt MX Foundry Web Applications in DX	64
Part 5: Use Volt MX Iris to turn a DX Site into a Native Mobile Application	67
Conclusion	73
Resources	74
Legal statements	75
Disclaimers	76

Author(s)

This document was created by the following Subject Matter Experts:



Herbert Hilhorst Company: HCLSoftware

Bio

Herbert Hilhorst is an HCL Digital Experience (DX) Technical Advisor at HCLSoftware.

Contact: herbert.hilhorst@hcl-software.com

Bio

Dai Nguyen is an HCL Volt MX Technical Advisor at HCLSoftware.



Dai Nguyen Company: HCLSoftware Contact: <u>dai.nguyen@hcl-software.com</u>

Bio

Mario Danna is an HCL Volt MX Senior Manager at HCLSoftware.

Mario D'Anna Company: HCL Software Contact: mario.danna@hcl-software.com

Introduction

HCL Volt MX Foundry is a back-end service provider that helps developers build omni-channel digital applications. Volt MX Foundry allows you to define the back-end to build native mobile apps for iOS, Android, and Windows and SPAs and Responsive web apps for browsers. Volt MX Foundry ensures that developers build mobile applications quickly and obtain secured back-end services instantly. Volt MX Foundry has multiple features, such as - Identity, Integration, Objects, Orchestration, and Engagement Services. These features can be accessed through a common, centralized Volt MX Foundry Console.

Many applications require imagery, videos, and content to bring them to life and create an engaging user experience. Such content and assets often come from a central content & asset management system, which is used by multiple mobile and non-mobile applications, to ensure consistent communication and branding across all customer interactions. This also allows business users to update the content & assets without needing to republish the application. HCL Digital Experience (DX) provides content and asset services, as well as services to personalize them, accessible via a set of GraphQL and REST APIs via HCL Volt MX Foundry.

Example: include news, help guides, tutorials, or other corporate content in your application; retrieve imagery or support videos from the asset management service for consistent branding & quick update.

This hands-on lab gets you started on the HCL Digital Experience (DX) platform and its integration with HCL Volt MX. You will learn how to manage digital assets and content centrally and use it in HCL Volt MX to manage your applications.

You will also learn how to use Foundry to integrate external data sources and Volt MX Web applications into DX. And you will learn how you may HCL Volt MX Iris to turn your existing DX sites into native mobile applications.

In this DX developer lab, you play the role of Gene, a developer for the fictitious Woodburn Studio company.



Gene Hayes, Developer, based in Chicago (USA)

As Gene, you will learn how to manage the REST APIs for DX assets in Foundry and build a mobile application using Iris. Then you learn how to manage the REST API for DX content in Foundry and build a web application with Iris. You learn how to use Foundry to integrate external data sources into DX and how to integrate Volt MX web applications into DX. And you will create a native mobile application of the Woodburn Stores DX site.

Prerequisites

- 1. Completion of the <u>HDX-INTRO</u> course as this gives you access to your own DX Solution Modules instance on HCL SoFy
- 2. Completion of Digital Assets and Web Content lessons of <u>HDX-BU-100</u>, as this gives you some assets and content to are used in this lab
- 3. Completion of Experience API of <u>HDX-DEV-100</u> as this helps you setting up the security to use the REST APIs and gets you started using them
- 4. Completion of Introduction of <u>HDX-INT-DEV</u> as this helps you understanding how to access the assets and content externally easily
- 5. A HCL Volt MX trial or live environment to develop HCL Volt MX applications that integrate your DX assets and content and Volt MX application installed on your favorite device(s). You may use HCL Volt MX Go on HCL SoFy (<u>https://hclsofy.com/catalog/hcl-voltmxgo</u>) or the Volt MX Hackathon Environment Setup guide to create your trial environment: <u>https://drive.google.com/file/d/17kP86SH7NCQarMXwtiag5OMitSrmG_hR/view</u>

You will be using the following user IDs and passwords:

Purpose	User	Password
SoFy Login	Your official email id	Your password
SoFy Solution Console Login	sol-admin	<from console="" solution=""></from>
DX Developer Login	ghayes	HCL-Dem0

Lab Overview

In this lab, you will explore the integration of HCL Digital Experience assets and content in HCL Volt MX.

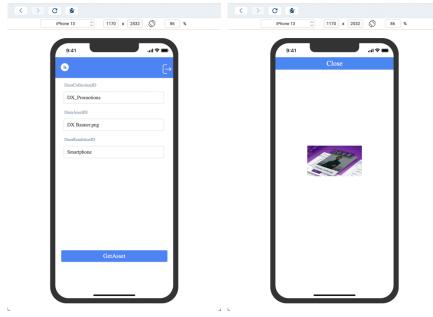
There are several parts in this lab, shortly introduced now.

Part 1: Use DX Assets in a new HCL Volt MX Mobile Application

In this part, you will develop, build, and run your Volt MX mobile application that uses your HCL Digital Experience assets. You will build a Volt Foundry Integration service to get the right details on an asset.

2	DXAssetsContentInteg	rationSample 🖉 🛛 1.0 🗸					
F	Configure Services Manage Cl	lient App Assets P	ublish				
2	Montity Integration	➢ Orchestration ☐ Ob	jects □+☆ Workflow	E Rules	ngagement		
0	+ 🖸 🕫 🖄 🛱 Q	Service Definition Operation	ns List getAsset 🗶				
6		Name*					Operation Security Level (?)
	 {} DXAssetsService (1.0) getAsset 	getAsset					Anonymous App Users 🗸
~~ ~		Target URL					Target HTTP Method 🕐
X		https://dx.sbx0392.play.hclsofy.co	m/dx/api/dam/v1/collections	/\$damCollectionI	0/items/\$damAssetID/renditions	/\$damRenditionID	GET -
E.	Test Log						×
	Request Payload	S	Backend Response			Output Result	
	{ "damCollectionID"; "DX_Promotions", "damAssetID": "DX Banner,png", "damRenditionID"; "Desktop" }		renditionType rendit	etadata	ø	api/dam/v1", 3. "opstatus" 4. "binaryUrl -eb245b517c7 p?binary=tru	": "/collections/675a2e80-3c4e-4ea1-9635 4/items/DX Banner.png/renditions/Deskto

And a Volt Iris project to access DX assets. This is how your mobile application will look like, using the collection, asset and rendition IDs.



Part 2: Use DX Content in a new HCL Volt MX Web Application

In this part, you will learn how to develop, build, and run a new Volt MX web application that uses your HCL Digital Experience content. To facilitate communication between the DX backend content and Foundry, you will create integration services in Foundry that will search DX content, based on a Content Template ID and Parent ID (Site Area) as input to the service, and a service to get details on a single content. Then you will build your web application with Iris that uses these services. It allows you to select a ContentTemplateID and ParentID and show the titles and IDs of the content that correspond.

		Volt MX I	ris Preview	
< > C *				
	Responsive	\$	1280 x 1769	7d) %
Ĩk.				Welcome, User name
ContentTemplateID			Title	ld
36a33863-6ea3-4d44-8430-aa528768	2aa2		Product 05	6c5abb2b-bdbd-461c-97ba-43c5994ed3f9
ParentID			Product 08	f4437e8b-6bf2-4e5f-939f-9e841781c0d7
			Product 04	bbaaf30f-8895-4c8f-99ca-60cfafd934c3
0407455c-68ed-455b-84c6-f916cedff	549		Product 01	4282f9fd-799d-48fb-a49b-65c9598016cd
			Product 03	5e89611c-b55f-47e4-a82d-16e8985cffe4
			Product 02	0929caaf-1ead-4fb4-b919-ae000e00e1e6
			Product 07	1fd886d1-e655-48da-a354-471adc11bcc8
			Product 06	374a89e4-ebde-4f5d-8c39-3f897c863705

And when selecting a product, you will see the details.

• • •	Volt MX Iris	Preview	
< > C	*		
	Responsive 🗘 128	0 x 1769 70 %	
Product Name	Product One		
Designer Name	Mariana Bowes		
Description	Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimem book. It has survived not only five centuries, but also the leap into lectronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letras thesets containing Lorem Ipsum		
Images			
Product-Im	age-01		
Product-Im	age-02		
Product-Im	age-03		

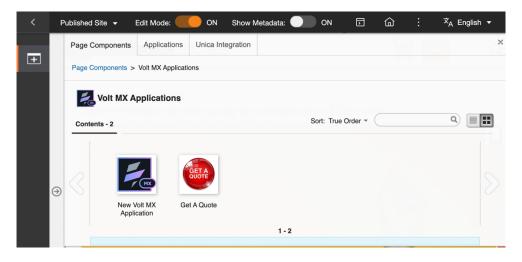
Part 3: Use Volt MX Foundry to Integrate External Data Sources in DX

In this part, you will learn how use Digital Data Connector to integrate data sources from HCL Volt MX Foundry into HCL Digital Experience.

<	Published Site 👻 Edit Mode: CN			۵	G	ម ខ	Ż _A English ▼	HCL	- i
Ŧ	← Back to home	۲	wpsedmin Actions Log Out		Sea	rch Site	0		
	Site Manager								
	≞ ⊵ ६ ६	6	Web Content Viewer (Contextual)				E* D*		
	Volt MX DDC		 Leanne Graham Brot Sinoern@april.biz 						
	Volt MX DDC								

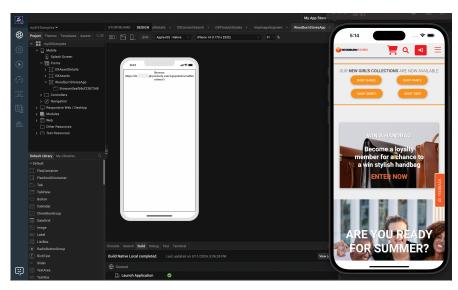
Part 4: Integrate Volt MX Foundry Web Application in DX

And you learn how to integrate Foundry web applications into HCL Digital Experience.



Part 5: Use Volt MX Iris to turn a DX Site into a Native Mobile Application

And you learn how you may use Volt MX Iris to build a native mobile application from any DX site and deploy this to any store, like Apple AppStore and Google Play.



Part 1: Use DX Digital Assets in a new HCL Volt MX Mobile Application

In this part, you will develop, build, and run your Volt MX mobile application that uses your HCL Digital Experience assets. You will build a Volt Foundry Integration service to get the right details on an asset and a Volt Iris project to access DX assets.

As you have configured the DX DAM to provide anonymous access, you can easily integrate them in Volt MX. You can directly use the DAM asset URL in your application. In this part you will build a Volt Foundry Integration service and a Volt Iris project to access DX assets using the REST API. It fetches a DX DAM asset, based on three input parameters. In the DX Integration Introduction, you learned how to access the asset with this REST API (see https://opensource.hcltechsw.com/experience-api-documentation/dam-api/#operation/RenditionController.getRenditionById): API URL:

<host>/dx/api/dam/v1/collections/<collection_id>/items/<content_id>/renditions/<rendition_id>

The first parameter is the "DAM Collection ID", and the second is the "DAM Asset ID". The third parameter "DAM Rendition ID" may be determined automatically depending on the device that is using the application. The output parameters, that your Volt Iris app will display, are the DAM binaryUrl and the DX Host. Your application will generate a valid URL to the DAM image.

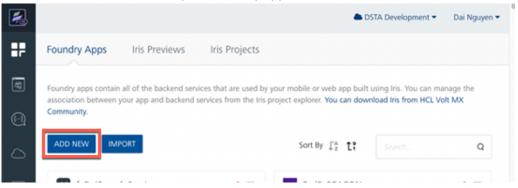
Ensure you have the console URL (for example, <u>https://manage.demo-hclvoltmx.com/</u>) to your cloud-Foundry and the required Volt MX credentials (email and password) to login to the Foundry.

If you are working with Volt Iris, in any of the steps below, ensure you save your updates regularly, at least once per step, using the menu Project – Save All or pressing CTRL S.

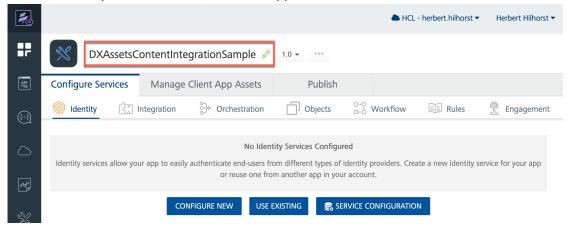
 You first need to create a Foundry application that contains the service to access the DAM to invoke that you will link to in your Iris application. While you could use the REST APIs directly, here you will use the Foundry API Application Management capability to create more consumable APIs to access resources in DX. Expand the left-hand-navigation-bar of Foundry and click **Apps** to start creating a new or viewing an existing application.

1	HCL Volt MX			📥 DST.	A Development 🔻	Dai Nguyen 🔻
:7	Apps	Iris Previews	Iris Projects			
Ę۳	API Management		es that are used by your mobile or web approvention of the set of			
\bigcirc	Developer Portals					
6	Environments	MPORT	Sort By ↓	٦î		Q

2. Then click Add New to create your Foundry application.



And update the application name with a good name, e.g.
 DXAssetsContentIntegrationSample. Note that the Foundry application name does need not be the same as your Iris web and/or mobile application.



4. Then add the integration services to access the assets in DX that will contain your new REST API calls. They are based on JSON. Under the **Configure Services** tab, click on **Integration** and then **Configure New**.

\$					📥 НС	L - herbert.hilhorst 🔻	Herbert Hilhorst 🔻					
₽₽	DXAsset	sContentInte	grationSample 🤌	1.0 👻 🚥								
	Configure Services	Manage	Client App Assets	Publish								
({···})	🔊 Identity [] Integration	⇒ Orchestration	Objects	^{O→} □←O Workflow	Rules	ngagement					
2	+ 🖸				gration Services Con							
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	All Services	8	0	Integration services allow your app to easily consume data from various external sources across a wide variety of API protocols. Configure a new integration service or reuse one from another app in your account.								
X				CONFIG	JURE NEW USE E	XISTING						

5. Name your new Integration Service, such as DXAssetsService. Note that service names need to be unique to a Foundry instance. You cannot have another service in your Foundry with the same name, even in another application. Select JSON as Service Type. Now you will configure this Integration service to be perform the GET DAM REST API call with your SoFy instance, which has this format (see <a href="https://opensource.hcltechsw.com/experience-api-documentation/dam-api/#operation/RenditionController.getRenditionById">https://opensource.hcltechsw.com/experience-api-documentation/dam-api/#operation/RenditionController.getRenditionById</a>):

<host>/dx/api/dam/v1/collections/<collection_id> /items/<asset_id>/renditions/<rendition_id>

First set the **Base URL** to https://<host>/dx/api/dam/v1/collections from this DAM REST API URL (e.g. https://dx.sbx0000.play.hclsofy.com/dx/api/dam/v1/collections with your SoFy host). Click SAVE & ADD OPERATION.

			HCL - herbert.hilhorst  Herbert Hilhorst
#	DXAssetsContentInte	grationSample 🖉 1.0 - ····	
API	Configure Services Manage	Client App Assets Publish	
	Montity	$\Rightarrow$ Orchestration $\square$ Objects $\stackrel{\circ \leftrightarrow}{}_{\square \circ \circ}$ Workflow	Rules ngagement
<u>م</u>	+ 🖸 🕫 🖄 🗊 Q	Service Definition *	
	All Services	Name*	Service Type Path Expression ⑦ Version
*	\ominus 📄 NewService (1.0)	DXAssetsService	JSON VEISION
X		Base URL*	
F.		https://dxplay.hclsofy.com/dx/api/dam/v1/collection	ns
L®		Web Service Authentication	Identity Service for Backend Token ⑦
		None Basic NTLM	None -
		> Advanced	
		Description	
			CANCEL SAVE SAVE & ADD OPERATION

6. Services are bundles of operations, each connecting to a full-qualified endpoint relative to the BASE URL of the service. You will need to add an operation to your service definition above to make the service usable. Name your new Operation getAsset. Set the Operation Security Level to Anonymous App Users, as it does not require a specific Identity to authenticate against before connecting to this operation. Ensure your Target HTTP Method is set to GET. Using \$ parameters to compose the Target URL means those parameter values will come from your Request Input parameter list (path params). To replace a param \$SomeName, you will need to pass a Request Input SomeName from the application invoking this Service/Operation.

Note that the Request Input parameter name must match those in the Target URL field.

Set the **Target URL** to match the GET DAM REST API call using these path parameters for collection_id, asset_id and rendition_id:

2	DXAssetsContentInte	grationSample 🖉 💷 🚥	
F	Configure Services Manage	Client App Assets Publish	
<b></b>	Montity	$\Rightarrow$ Orchestration $\square$ Objects $\overset{\circ}{}_{D-\diamond}$ Workflow $\blacksquare$ Rules $\P$ Engagement	
Ð	+ 🖸 🕫 🕲 🕅 Q	Service Definition Operations List getAsset X	
0	<ul> <li>All Services</li> <li>() DXAssetsService (1.0)</li> <li>getAsset</li> </ul>	Name* getAsset	Operation Security Level ⑦ Anonymous App Users
*		Target URL	Target HTTP Method
X		https://dx. play.hclsofy.com/dx/api/dam/v1/collections /\$damCollectionID/items/\$damAssetID/renditions/\$damRenditionID	GET -
F.		> Advanced Description	
		Request input Response Output	
		Body Header	
			Enable pass-through: Input Body 📀
		+ Add Parameter	Request Template ⑦ Show Hide

/\$damCollectionID/items/\$damAssetID/renditions/\$damRenditionID

7. Then add the path and query parameters and set their default values. Scroll down and under Request Input, click Add Parameter to add each parameter: damCollectionID with the Default Value to your DX DAM Collection ID, such as ccfd0442-36ba-44e5-9c27-05299727635b or DX_Promotions (Unique name), and damAssetID with the Default Value to your DX DAM Asset ID, such as ccfd0442- 1bdf5cdd-3f5d-45be-b595-d5126832b597 or asset file name, such as DX Banner.png and the damRenditionID with the Default Value for mobile (as you are creating a mobile application this should be Smartphone). Select your preferred Foundry runtime environment in the bottom to test this service against (ideally the same runtime environment you'll eventually be publishing this service to) and click SAVE AND FETCH RESPONSE to save your Integration service operation and make a test call immediately.

+ Ad	ld Parameter	/ 🕞 Paste 🛱 Delete <			Re	equest Template 🥐	Show Hide
							Q
	NAME 🗘	VALUE (?)	TEST VALUE	DEFAULT VALUE	DATA TYPE	ENCODE	DESCRIPTION
	damCollectionID	request		DX_Promotions	string		
	dam Asset ID	request		DX Banner.png	string		
	damRenditionID	request		Smartphone	string		
ault va	lue will be used if Test value is e	impty.		m1000	(	▼ SAVE AND	D FETCH RESPONS
						CANCEL	SAVE OPERATIO

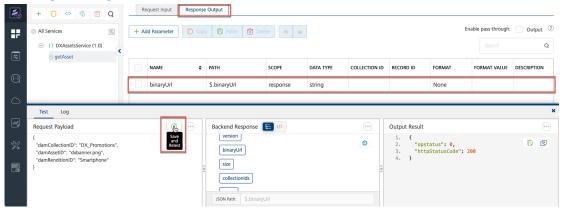
8. The Test panel is appearing now and shows the Test and Log sections. The Test sections shows the Request Payload, the Backend Response and the Output Result. Backend Response displays what DX communicated back to Foundry, and Output Result is what Foundry will relay to the Iris application invoking this service. Since we have not specified what data-points to pick and choose for our application or enabled the entire output to *pass-through* to the application, Output Result only includes opstatus and httpStatusCode by default. You want also the binaryURL in the output result.

Ş۳	🖲 getAsset	Re	quest Input Response	e Output						
			Header Id Parameter Cop	y 🍞 Paste 🖞	Delete				nable pass-through: equest Template ⑦ Search	Input Body ? Show Hide
X			NAME \$	VALUE ?		TEST VALUE	DEFAULT VALUE	DATA TYPE	ENCODE	DESCRIPTION
Ę			damCollectionID	request			DX_Promotions	string		
			damAssetID	request			dxbanner.png	string		
			damRenditionID	request			Smartphone	string		
	Test Log									×
	Request Payload		• ••	Backend Respo	nse 🔚 🗘	•••	Output Resul	t		
	{ 'damCollectionID": "DX_Promotions", 'damAssetID': 'dxbanner.png', 'damRenditionID': "Smartphone" }			emb	yyDate eddedMetadata nateText o generate JSON Path.	٥		pstatus": 0, ttpStatusCode"	: 200	r ø

9. You want to add the binaryURL to this result. In this lab, you are focusing on images only, so you know the format of the asset to display. If you want to manage different asset types, you may want to add a mime-type here, as this may help you identify what asset you are finding and how to display it correctly. You will focus on image assets for now, but you may use the DAM for managing videos and other files too. Scroll down in the **Backend Response** to locate the **binaryURL** of your asset in the REST API response structure. Hover over the **binaryURL** node, so it will show **binaryURL – Create Response** and then **click** it.



10. This switches you to the **Response Output** and you see the added **binaryUrl** parameter. Now test it again, using the **Save and retest** icon.



11. Now you see the **binaryUrl** added in the **Output Result**. The DAM REST API response does not contain a full URL to the asset, but a relative part that starts after the <host>/dx/api. To be able to directly point to the image, you need to complete this URL. You want to add a post process to this result that adds the host to this output. You do this under the advanced section. Click **Advanced** to expand.

2	+ 🖸 🍫 🚳 💼 Q	https:	//dxplay.hclsof	.com/dx/api/dam	/v1/collections /\$damC	ns=\$damRenditionID	GET		•				
<b>₽</b>	<ul> <li>All Services</li> <li>All Services</li> <li>DXAssetsService (1.0)</li> <li>getAsset</li> </ul>	> Adv Descript											
0		Re	equest Input Respo	nse Output									
$\bigcirc$		+ A0	dd Parameter	eter 🔽 Copy 😨 Paste 💼 Delete 😞 😽 Enable pass-through: 🗌 Output 🖗									
<b>*</b>									Q				
X			NAME	E + PATH SCOPE DATA TYPE COLLECTION ID RE						FORMAT VALUE	DESCRIPTION		
E.			binaryUrl	\$.binaryUrl	response	string			None				
	Test Log										×		
	Request Payload		<b>S</b>	Backend F	Response 🔚 🛷			Output Result					
	{ "damCollectionID": "DX_Promotions", "damAssetID": "dxbanner.png", "damRenditionID": "Smartphone" }			size Select any for the second secon		h.	٥	3. "binar 5-eb245b 9b-136e-	517c74/items	/dxbanner.png/r becefeleba?bina	ت لی 80-3c4-4ea1-963 enditions/c76c27 ry=true",		

12. You can create any JSON structure in the output response by manually entering paths/names/data-types, or even generate completely custom response JSON structures (including fields not present in original backend response, but generated based on custom-business-logic) using <u>Data Processors</u> (Pre and Post processors). Post processors are used to modify the backend-response using custom business logic. In this case, the custom logic is written in JavaScript and injects a new field into the response. Select JavaScript under **Postprocessor** and add the two JavaScript lines below, with your own DX host (change sbx0000 to your instance):

var tmpDXHost = "https://dx.sbx0000.play.hclsofy.com/dx/api/dam/v1";

result.addParam("DXHost", tmpDXHost);

Then select **Response Output** and click **Add Parameter** and add **DXHost** as a parameter. Click **Save And Fetch Response** to save your service and to make a test call.

2	+ 🖸 🕫 🕲 Q	✓ Adv	ranced								
	All Services	Custom	Code Invocation Stub Ba	ckend Response Propertie	is Front End API Pass-t	through Cookies Server Ev	ints				
<b>;</b> ;	<ul> <li>Al Jervices</li> <li>DXAssetsService (1.0)</li> </ul>	Preproc	cessor (?)				Postprocess	or (?)			
4	getAsset	💿 Java	a 🔿 JavaScript 🔿 Rules				💿 sval 🔵	JavaScript 🔵 Rules			
		Class									<b>B</b> (2)
O		Select	a Class				<ul> <li>var tmpb, result.add</li> </ul>	(Host = "https://dxp Param("DXHost", tmpDXHost	olay.hclsofy.com/dx/api/dam/\ ();	a 5	· C
$\sim$											
æ				No class /	Available						
×											
~											
8		Descript	tion								
		Re	equest Input Response	Output							
			dd Parameter					Enable pa	s-through: Output 🕐		
			Parameter Copy C Paste î Delete <								earch Q
											<b>4</b>
			NAME \$	PATH	SCOPE	DATA TYPE	COLLECTION ID	RECORD ID	FORMAT	FORMAT VALUE	DESCRIPTION
			binaryUrl	\$.binaryUrl	response	string			None		
			DXHost		response	string			None		
		Default v	alue will be used if Test value is e	impty.					m10000	0625002 -	AVE AND FETCH RESPONSE

13. The **Output Result** now has **DXHost** and **binaryUrl** output parameters. Your Volt Iris project will use both to compose a valid URL to your DAM asset.

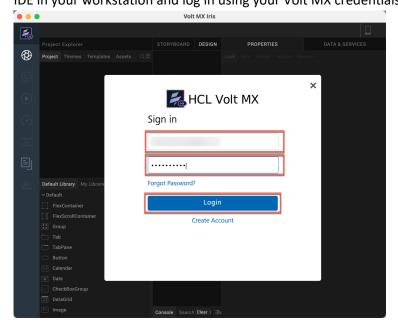
				Output	te 😞 🗞				Enable	e pass-through: [ Search	Output Q	
		NAME	¢	PATH	SCOPE	DATA TYPE	COLLECTION IE	RECORD ID	FORMAT	FORMAT VAL	DESCRIPTION	
		binaryUrl		\$.binaryUrl	response	string			None			
		DXHost			response	string			None			
Test Log												×
Request Payload	•			Backend Response 🔚 🚸			•	Output Result				)
{ "damCollectionID": "DX_Promotions", "damAssetID": "dxbanner.png", "damRenditionID": "Smartphone" }	g",						1. {					a1–

14. You are ready to publish your new Foundry application **DXAssetsContentIntegrationSample** for consumption in Volt Iris. Click **Publish**, select your server and click **Publish**.

	DXAssetsCo	ontentIntegrationSam	ple 🖉 1.0 🗸			
<b>I</b> F	Configure Services	Manage Client App As	ssets	Publish		
API	Service & Web Clien	t 🔛 Native Client				
	m1000					
$\bigcirc$	CONSOLES		SERVICE CONFIG	APP STATUS	ENVIRONMENT STATUS ?	HISTORY & ROLLBACK
~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~			-			
21/					CONFIGURE &	PUBLISH PUBLISH

1 DXAssetsContentIntegrationSample 🤌 1.0 👻 −₽ Manage Client App Assets Publish **Configure Services** (Service & Web Client Native Client API (-)m1000 DEFAULT VERSION 🕵 🕹 1.0 CONSOLES HISTORY & ROLLBACK SERVICE CONFIG PP STATUS ENVIRONMENT STATUS (?) 🚯 Custom Published Available Snapshots (C) (D) (C) (C) (C) 24 Mar 2023 08:40 LITC ₩₽ CONFIGURE & PUBLISH PUBLISH UNPUBLISH X × Published DXAssetsContentIntegrationSamp.. (v1.0) to m10000 F Skipping publish of Engagement Services as they are not configured Updated publish resources successfully App Information Primary App Key d96072a122cfe93d Primary App Secret 48ff4520d0256b93 f8fd7017b712f67c6 Secondary App Key Secondary App Secret eaae3e86b57b5410 Service URL https://10(auth.demo-hclvoltmx.net/appconfig

16. Now you will use the Volt Iris IDE to create a web application that displays the DX image resource links, using low-code. First, you will create a linkage to the Foundry service you just created: **DXAssetsService**. Ensure you have set up Volt Iris IDE and have the right credentials. See the prerequisites for instructions on how to set this up. Then launch Volt Iris IDE in your workstation and log in using your Volt MX credentials.



15. And then it gets published and finally shows the application status as published.

DX INTEGRATION WITH HCL VOLT MX FOR DEVELOPERS

17. Ensure you have selected the right cloud account.

				i 2	📥 HCL - herbert.hilhorst	👤 Herbert Hilhorst
	Sample 🛩	STORYBOARD	DESIGN	DATA & SE	HCL - herbert.hilhorst Account ID: 1	
֎	Default Library My Libraries Q	Mobile		Volt Found	iy backenu. No app attaciled	Refresh 📃
<u>~~</u>	→ Buttons			> SAMPLE	SERVICES	
्रि				> PROJEC	I SERVICES	

18. Create a new project. In the menu, open **Project** and click **New Project**.

é	Volt MX Iris	Project	Edit	Build	Debug	Forge	Window	Help
••	•	New Proje	ect			NΣ		Volt MX Iris
		Open				>		
		Open File						
	Sample 🗸	-					DESIGN	

19. Select Native App and click Next.

What do you want to start with now?	×
···· ···· ···· or ···· Veb App ···· ···· ···· ···· Web App ···· ···· ···· ···· Sample Apps	
Start with building either a web app or native app. Alternatively, you can also choose from one of our sample app projects to load and navigate around. Not sure what to start with? Help me decide	
You can later add or modify your selection.	
Next	

20. Name your Iris project, for example **myDXSamples** and click **Create**.

	You started with a Native App .	×
	Project Name	
myDXSample		
	Advanced Project Settings	
Back		Create

21. Link your new Iris application to your Foundry application first. Under **DATA & SERVICES**, to the right of your Volt Foundry Backend application, open the menu and select **Link to Existing App**.

			📋 🚦 🌰 HCL - herbert.hilhorst 💄 Herbert Hilhorst
	myDXSamples 🗸	STORYBOARD DESIGN	DATA & SERVICES
\bigotimes	Default Library My Libraries	Mobile 🗘	Volt Foundry Backend: No app attached
~~~	✓ Buttons		SAMPLE SERVICES     Create New App
ŝ	Button Button 1 Button 2		PROJECT SERVICES     Link to Existing App
$(\mathbf{b})$	BorderButtonRound BorderButtonRound	DXAssets	Manage Linked Apps
	Button 1 Button 2 Button 1 Button 2 Button 3	5,1,000,0	Export App to Foundry

22. This shows all the published Foundry applications. Click **ASSOCIATE** for your **DXAssetsContentIntegrationSample** application.

1						HCL - herbert.hilhorst	💄 Herbert Hilhorst
					DATA &		
$\bigotimes$		Q Mobile					
		B-05/F2					
	BorderButtonRound BorderB	Volt MX Applications				×	
	Bullon1 Bullon2 Bullon1			Search		۹	
	BorderButtonRound BorderB						
	Https://www.akodulf.p.com	FOUNDRY APPS	CREATED ON	LAST MODIFIED ON	VERSION		
	HyperLinkButton Paddeo	DXAssetsContentIntegrationSamp	le 21 Mar 2023 09:42 UTC	21 Mar 2023 09:48 UTC		ASSOCIATE	
	Budon 1 Budon 2 Budon 1	News Weather Sample	27 Jun 2022 18:23 UTC	27 Jun 2022 18:23 UTC		ASSOCIATE	
	PaddedButtonRound Padded	PetStore Management Sample	27 Jun 2022 18:23 UTC	27 Jun 2022 18:23 UTC	$\circ$	ASSOCIATE	
	Button 2 Button 3	VoltMX Sample OAuth	27 Jun 2022 18:23 UTC	27 Jun 2022 18:23 UTC	$\bigcirc$	ASSOCIATE	
	PaddedButtonRound RoundF						

23. Then refresh the DATA & SERVICES first. Click Refresh.

			🛄 📋 📕 MCL - herbert.hilhorst 💄 Herbert Hilhorst
	myDXSamples 🛩	STORYBOARD DESIGN	DATA & SERVICES
⊗	Default Library My Libraries	Mobile ¢	Volt Foundry Backend: DXAssetsContentIntegratio Refresh
~	✓ Buttons		> SAMPLE SERVICES
හි	BorderButtonBound BorderButtonBound		> PROJECT SERVICES

24. Then go to the design view. Click **Design**.

1						Ë		📥 HCL - herbert.hilhorst	💄 Herbert	Hilhorst	d
	myDXSamples 🛩		STORYBOARD	DESIGN		DATA	& SE	RVICES			
	Default Library My Lib		Mobile	0				lry Backend: DXAssetsConte	ntIntegratio	Refresh 📃	
	✓ Buttons					> sa	MPLE	SERVICES			Į,
	BorderButtonBound	BurderButtonBound				> PF	OJEC	T SERVICES			

25. Expand **Mobile** and **Forms** and you will see a default first form, named **Form1**. Click it and the rename it to a better name, e.g. **DXAssets**, using the down-control to the right of **Form1** using **Rename**.

					📋 🚦 🌰 HCL - herbert.hilhors	st 💄 Herbert Hilhorst
	myDXSamples 🛩	STORYBOARD DESIGN	Form1 × App Events [Mo		PROPERTIES	DATA & SERVICES
	Project Themes Templates Assets $Q\equiv$	OIO 📇 🋄 BVR	Apple-iOS : Native 🗘 iPh	one 13 Pro Max (1284 x 27👧	Volt Foundry Backend: DXAssetsCo	ntentIntegratio Refresh 📃
ණ	<ul> <li>✓ ■ myDXSamples</li> <li>✓ □ Mobile</li> </ul>				> SAMPLE SERVICES > PROJECT SERVICES	
ightarrow	Splash Screen	9:41	· ■ ♦ Ito.			
(?)	Controllers Js Form1Col Rename					
	Js Form1Cor Duplicate					

 Under DATA & SERVICES, expand the PROJECT SERVICES and then expand your DXAssetService service and then drag the getAsset operation to your newly created form DXAssets, as shown.



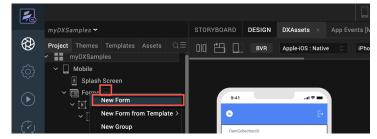
27. Select Entry form for Request when asked, then click OK.

What would you like to	generate?
Details Using Response	Entry form for Request
	Cancel

28. You should see **DXAssets** updated, generated via Iris low-code capability, with a UI to input the DAM Collection ID, DAM Asset ID and Rendition ID, along with a button **GetAsset** at the bottom. You may change the labels and button texts by selecting the items and modifying the properties. You could also hide the DamRenditionID and generate this automatically for the device using the asset. To keep this lab simple, leave it as it is.

					🗒 🚦 🌰 HCL - herbert.hilhorst	💄 Herbert Hilhorst
	myDXSamples 🗸	TORYBOARD DESIGN D	XAssets × App Ev	rents [M $\equiv$	PROPERTIES	DATA & SERVICES
$\bigotimes$	Project Themes Templates Assets Q ≡ ✓ myDXSamples		opple-iOS : Native 🗘	iPho 🕑 :	Volt Foundry Backend: DXAssetsCont	entIntegratio Refresh 🚞
ු	✓ 🛄 Mobile ∰ Splash Screen ⊠				<ul> <li>SAMPLE SERVICES</li> <li>PROJECT SERVICES</li> </ul>	
lacksquare	✓ I Forms ✓ DXAssets	9:41			<ul> <li>         ⊕ Configure New         <ul> <li>             ⊕ Use Existing</li></ul></li></ul>	
(~)	✓ ☐ flexHeader ImgHeaderLogo ○ btnLogout	DamCollectionID	[→		> () getAsset	View Mapping
Æ	binLogout     flexFieldsContainer     flexGamCollectionID	DamAssetID				
Ej	flexdamAssetID     Image: flexdamRenditionID	DamRenditionID				
×	✓ ☐ flexFooter > ☐ flexSquaredBorderedButton					
	Default Library My Libraries			Ë		
	∽ Default					
	FlexContainer					
	FlexScrollContainer					
	🛅 Tab					
	TabPane					
	Button	GetAsset				
	📰 Calendar					
	CheckBoxGroup					
	DataGrid					
	📨 Image	onsole Search Build Debu	n Test Terminal	Clear 📃		
	abc Label	TTHE TAKEN [TOTHS]				
	E. ListBox	Opening Form 'DXAssets'.				
	RadioButtonGroup	Time Taken [124ms]	ded in DeteDanel			
	T RichText	DP : Parameters are load	ueu in DataPanel			

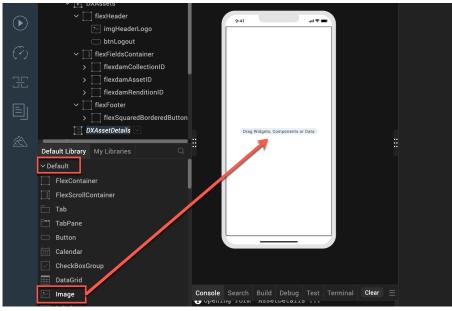
29. Now you want to add a new view to render your asset. Add a new form to your Iris mobile channel. Open the menu to the right of Forms and click **New Form**.



30. Then rename your new form to **DXAssetDetails**.

	Splash Screen		
	∽ 💼 Forms		
$\bigcirc$	🗸 💽 DXAssets		41
igvee	✓ ☐ flexHeader		
	📩 imgHeaderL	ogo	
	🗆 btnLogout		
	✓ ☐ flexFieldsConta	iner	
	> 🗌 flexdamColle	ectionID	
	> 🗌 flexdamAsse	etID	
	> 🗌 flexdamRen	ditionID	
	∽ 🗌 flexFooter		
	> 🗌 flexSquared	BorderedButton	
ź	DXAssetDetails		Drag Widgets, Components or Data
		Open	
	Default Library My Libraries	Rename	
	∼ Default	Duplicate	

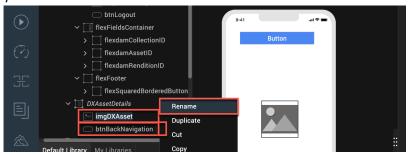
31. You are focusing on image assets for now. To render the asset, you will add an image widget to your new form. From the **Default Library**, **Default** widgets, drag the **Image** widget to your new form.



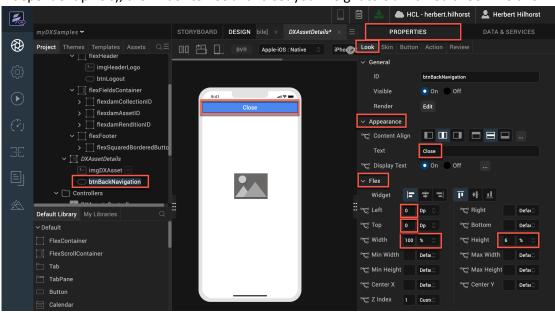
32. Your image widget now shows under your new form. Now add a button that allows you to go back to the previous DXAssets form. Drag the **Button** widget to the top of your new form.



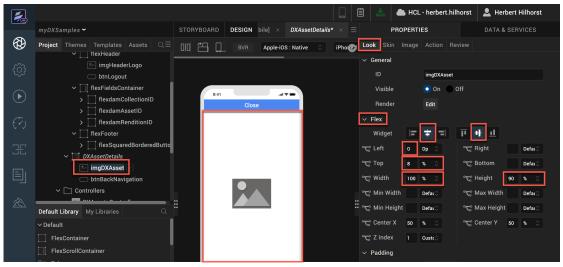
33. You now see your new button. Similar as with your forms, rename these widgets to a more easily understandable name, like **imgDXAsset** for your image and **btnBackNavigation** for your button.



34. Set the properties of your button to appear at the top of the screen and occupying the entire width of the viewport. To do this, select the button from the project explorer or click on the button in the canvas, then select **PROPERTIES**. Now change, under **Look** and **Appearance**, the button **Text** to **Close**, and under **Flex**, the **Left** and **Top** margins to **0** Dp (Density independent pixels), the **Width** to **100** % and set your Height to **6** %. It would look like this.



35. Now change the image. Select it and then under Look, change the Flex settings to pin the Widget to the Center and Middle, set Left to 0 Dp, Top to 8 %, Width to 100 % and Height to 90 %.



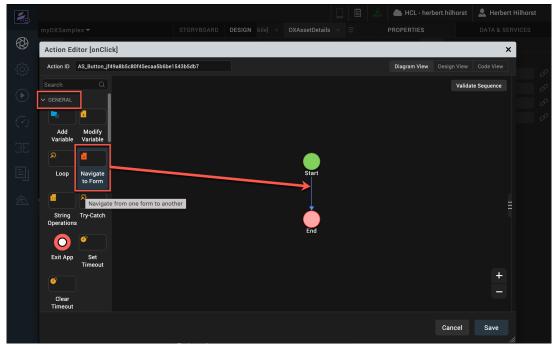
36. Then go to the **Image** settings and ensure the **Scale Mode** is set to **Maintain Aspect Ratio** to ensure the image is rendered in its initial aspect ratio.

				🗄 📩 🌰 HCL - herbert.hill	norst 💄 Herbert Hilhorst
	myDXSamples 🗸	STORYBOARD DESIGN bile] × DXAssetDetails		PROPERTIES	DATA & SERVICES
	Project Themes Templates Assets $Q\equiv$	010 EVR Apple-iOS : Native 🗘	iPhore:	Look Skin Image Action Rev	view
~~~~	✓ IlexHeader ImgHeaderLogo			V General	t to Dimensions
ţ	btnLogout				aintain Aspect Ratio
(\mathbf{b})	flexFieldsContainer flexdamCollectionID	9:41		Course	rop Contagraphy

37. You want your users to use the Close button to go back to the DXAsset form. Therefore, you will use an Iris generated low-code action script for the onClick event of your btnBackNavigation button. Ensure the focus in on the btnBackNavigation button. Select PROPERTIES - Action and click Edit button next to the onClick event.



38. The Action Editor shows and from the action palette, drag the **General** - **Navigate to Form** action to the flow-line between **Start** and **End** in the action tree.



		STORYBOARD DESI			PROPERTIES	
Action Ec	litor [onClick]					×
Action ID	AS_Button_jf49a8b5c80f45ec	caa5b6be1543b5db7			Diagram View Design View	Code View
Search General Add Variable Cop Cop Exit App G	Navigate to Form ? Try-Catch s	Start Navigate to Form DXAssets End	Validate Sequence	Create New DXAssets DXAssetDetails	Pass data w	ith navigation

39. Then select the **DXAssets** form it should navigate to and click **Save**.

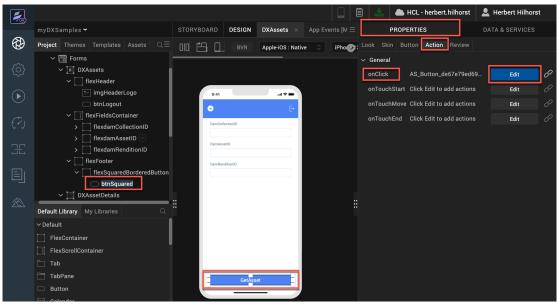
40. To display a DX image, you will need the fully qualified DX asset URL. You will use a global variable to hold the DX URL. Add this using the menu **Edit – Global Variables**.

É	Volt MX Iris Project	Edit Build Debug Forge	Window	Help 💿 🛂 🛞 🗔 🔺
••	•	Undo	жZ	Volt MX Iris
F		Redo	<u> </u>	🔲 📋 📩 🌰 HCL - herbert.hilhorst 💄 Herbert Hilhorst
	myDXSamples 🗸	Cut	ж Х	bile] × DXAssetDetails × = PROPERTIES DATA & SERVICES
	Project Themes Temp	Copy Paste	ж С ж V	Apple-iOS : Native \Diamond iPhone: Look Skin Button Action Review
~~	∽	Delete		✓ General
ŝ	> 🔝 fle> ~ 📃 DXAssetI	DeleLast Edit Locationte	τQ	onClick AS_Button_jf49a8b5c80f Edit \mathscr{O}
(\mathbf{b})	🥍 imgD>	Search	<mark>ት</mark> ዝ F	onTouchStart Click Edit to add actions Edit 6
\odot	── btnBa ✓	Select All	жA	e onTouchMove Click Edit to add actions Edit 🖉
(~)	Js DXAssets	Find & Replace Skin Properties		onTouchEnd Click Edit to add actions Edit 🔗
Ъ.	Js DXAssets Js DXAssets Js DXAssetl	Manage Cordova Plugins Manage Native Function API(s)		
Ej	> Ø Navigation	Manage Web Libraries		
	> Modules	Global Variables		
ź	> 📰 Web	Start Dictation Emoji & Symbols	¥ @	
	∽ Default			

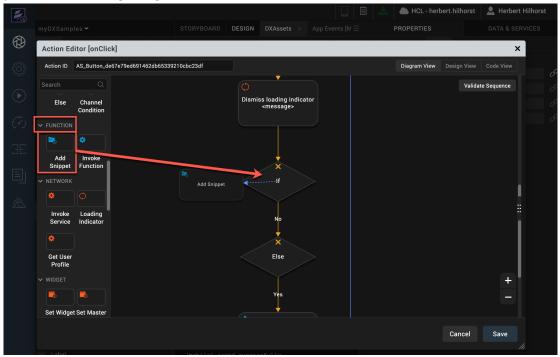
41. Keep it as a **Simple Type**, name it **gblDXAssetURL** to hold the DX asset URL and close the dialog box.

			UNCIEK AS_BUIL
	9:41		onTouchStart Click Ed
	Variables		× ouchMove Click Ed
	Type: Simple	٥	TouchEnd Click Ed
ns V	+ × 1 +		
Action	Variable	Default Value	
	gbIDXAssetURL		
;			

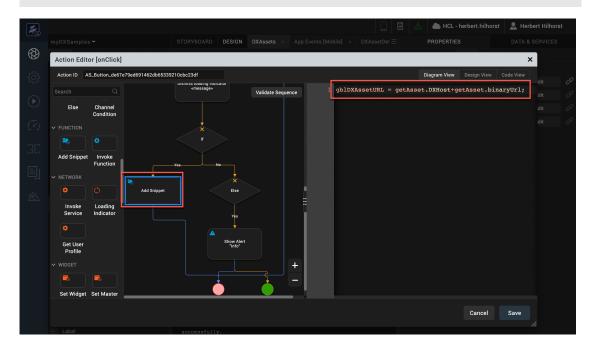
42. You will set this variable when your user clicks the GetAsset button on your DXAssets form and then open your DXAssetsDetails form. Select your **DXAssets** form and **GetAsset** button (it may have a different default ID), then **PROPERTIES**, **Action** and click **Edit** for the **onClick** event.



43. In the Action Editor you will see the auto-generated action flow for the click event of the button. You will notice that this covers the failure scenario for the service call Show Alert "Info" stating that the service has failed. You will need to create the action flow for the success condition. Here, if the service succeeds, you will initialize the global variable you just created and then navigate to the next form DXAssetDetails. Drag the Add Snippet action under FUNCTION to the If node attached to the service Callback in the script tree. The If node currently only has the No and Else branch. Drop the Snippet next to the If node when you see the "blue" guiding arrow.

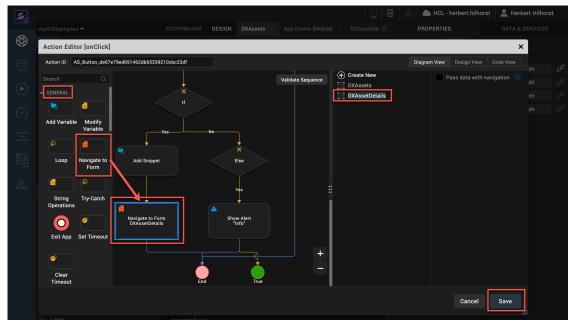


44. Then add the JavaScript line below that generates the global variable to the fully qualified URL.



gblDXAssetURL = getAsset.DXHost+getAsset.binaryUrl;

45. Then add the navigation to the DXAssetDetails form. Add the **GENERAL** – **Navigate to Form** between the **Add Snippet** and **End** using drag & drop, select the **DXAssetDetails** form and click **Save**.

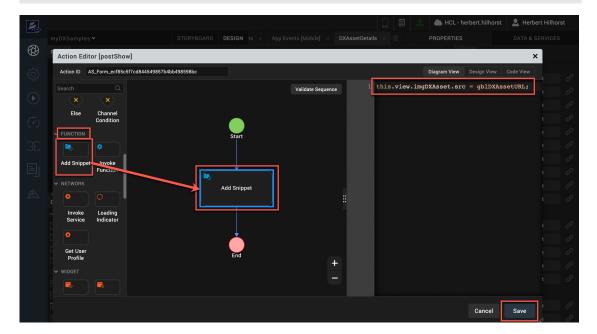


46. Then you need to use this URL in the global variable in your DXAssetDetails form. Select your DXAssetDetails form. Then under PROPERTIES – Action click the Edit button for the preShow event.

] 📥 HCL - herbert.hilhorst	👤 Herbert Hilhorst
			DESIGN ts × App E	vents [Mobile] × DXAssetDetail	Is $\times \equiv$	PROPERTIES	DATA & SERVICES
	Project Themes Templates Assets $Q\equiv$	00 📇 🗋	BVR Apple-iOS : Nat	ive 🗘 iPhone 13 Pro Max (1284	4 x 2778	Look Skin Form Action Review	
-0-	🗇 btnLogout					 General 	
ි	 flexFieldsContainer flexdamCollectionID 					init Click Edit to add action	s Edit 🖉
\bigcirc	> flexdamAssetID	9:41				onMapping Click Edit to add action	s Edit 🖉
	> flexdamRenditionID		Close		[preShow Click Edit to add action	s Edit 🖉
(~)	→ ☐ flexFooter					postShow Click Edit to add action	s Edit 🖉
(*)	flexSquaredBorderedButton btnSquared					onHide Click Edit to add action	s Edit 🖉
E	V DXAssetDetails					onDestroy Click Edit to add action	s Edit 🖉
	imgDXAsset					onNavigate Click Edit to add action	s Edit 🖉
E	btnBackNavigation Controllers					onTouchStart Click Edit to add action	s Edit 🖉
	Js DXAssetsController					onTouchMove Click Edit to add action	s Edit 🖉
Ŕ	Default Library My Libraries					onTouchEnd Click Edit to add action	s Edit 🖉
	∽ Default					√ iPhone	
	FlexContainer					onOrientatio Click Edit to add action	s Edit 🖉

47. Then add the **Add Snippet** between **Start** and **End** and add the code that sets the source of your **imgDXAsset** image widget to your global URL **gbIDXAssetURL**. You do that with the following JavaScript. Then click **Save**.

this.view.imgDXAsset.src = gblDXAssetURL;



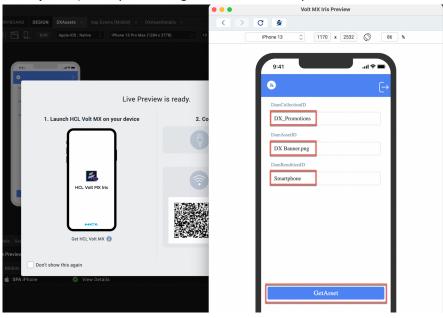
48. You are now ready to build and test your mobile channel application. Click **Build** -> Live **Preview Settings**.

É	Volt MX Iris	Project	Edit	Build	Debug	Forge	Window	Help	
•••				Run Liv	e Preview	,	ЖR		Volt MX Iris
				Live Pr	eview Set	tings			
2				Custom	nize Volt N	ИХ Арр			
0	myDXSampl	es 💙		Publish	Live Prev	view	iN		App Events [Mobile] ×
	Project The	mes Temp	lates /		ive Previe		R	Apple-	iOS : Native 🗘 iPhone

49. Then check any **ADAPTIVE WEB** option that matches your favorite mobile device, for example **iOS**. Then ensure you have the same environment selected as the one where you published your Foundry application. Select **Clean Preview** to clear build caches and click **Save & Run**.

STUNTBUARD DESIGN IS A App Events [woulde] A DA	Asserberans ~	= PROPERTI
Live Preview Settings		×
Platform and Channels		
다. RESPONSIVE WEB		
	NATIVE	ADAPTIVE WEB 🚯
MOBILE		
é ios		
🛱 Android	\Box	
TABLET		
tios 🔹		
🗰 Android		
Environment m100 v9.5 Change		
Preview Mode Debug 🕥 🕄 🔽 Clean Preview		
Get HCL Volt MX 🚯		
Project Settings Cance	el Save	Save & Run
United		

50. You should see your application running as an iPhone SPA. Provide a valid **DAM Collection ID** (e.g. **DX_Promotions**), **DAM Asset ID** (e.g. **DX Banner.png**) and **DAM Rendition ID** (e.g. **Smartphone**) from your DX Digital Assets to the input text boxes. Click the button **GetAsset**.



51. You should see your DAM asset. Your business user can easily update these assets with a user-friendly interface. Feel free to close and try with a different collection, asset ID and rendition.



Congratulations! You have successfully developed, built, and ran your Volt MX mobile application using the Volt MX low-code capability that renders a DX asset, managed by business users.

Part 2: Use DX Content in a new HCL Volt MX Web Application

In this part, you will learn how to develop, build, and run a new Volt MX web application that uses your HCL Digital Experience content. To facilitate communication between the DX backend content and Foundry, you will create integration services in Foundry that will search DX content, based on a Content Template ID and Parent ID (Site Area) as input to the service, and a service to get details on a single content. You will also learn to use the HCL Digital Experience Content Adaptor that is available on the Volt MX Marketplace. Then you will build your web application with Iris that uses these services. It allows you to select a ContentTemplateID and ParentID and show the titles and IDs of the content that correspond.

And when selecting a product, you will see the details.

1. First select **Apps** in Foundry and click your **DXAssetsContentIntegrationSample** Foundry app you built in **Part 1**.

K	Foundry Apps Iris Previews Iris Projects	
	Foundry apps contain all of the backend services that are used by association between your app and backend services from the Iris Community.	
{···}	ADD NEW IMPORT	Sort By $\[\]_z^A \] 2$ Search Q
	DXAssetsContentIntegrationSam $\Rightarrow \equiv$	🚳 VoltMX Sample OAuth 🛛 😒 🖃 🚍
*	Version Modified By Modified On 1.0 Herbert Hilhorst 21 Mar 2023 09:48 UTC	Version Modified By Modified On 1.0 N/A 27 Jun 2022 18:23 UTC

2. Then create a new integration service. Select Integration, CONFIGURE NEW.

	S DXAS	ssetsCor	ntentInte	egrationSample 🤌	1.0 👻 🚥		
	Configure Serv	rices	Manage	Client App Assets	Publish		
API	M Identity	िर्दे Inte	egration	→ Orchestration	Objects	Sector Workflow	≚≝ Rules
({···})	+ 0	Š Í	Q	CONFIGURE NEW	USE EXISTING		
	All Services		0-0	NAME	SERVICE T	YPE VER	SION
\bigtriangleup	⊕ {} DXAsset	sService (1.	.0)	DXAssetsService	JSON	1.0	

3. Then **name** it **DXContentService** and select **JSON Service Type**. Now configure this integration service with your DX SoFy instance. In the DX Integration Introduction, you have seen the structure of the REST API calls to get DX content. Assign the **Base URL** to the web content REST API URL for your DX SoFy instance:

<host>/dx/api/core/v1/<access_type>/webcontent, for example,

https://dx.sbx0000.play.hclsofy.com/dx/api/core/v1/dxrest/webcontent (replace 0000 by the one of your instance). You have anonymous access to your content with the REST API, so you can use **Web Service Authentication** - **None**. If you would use protected content, you need use **dxmyrest** and **Basic** Web Service Authentication using the **User ID** and **Password** of a user that has access to your content provided in your SoFy instance. Then click **SAVE & ADD OPERATION**.

%	DXAssetsContentInte	grationSample 🖉 1.0 - 🚥	
₽₽	Configure Services Manage	Client App Assets Publish	
API	Montity	$\stackrel{\circ}{\Rightarrow}$ Orchestration \square Objects $\stackrel{\circ}{\underset{\square}{\rightarrow}}$ Workflow	E Rules ngagement
()	+ 🖸 🕫 🖏 🛱 Q	Service Definition *	
	 All Services NewService (1.0) { DXAssetsService (1.0) 	Name* DXContentService	Service Type Path Expression ⑦ Version JSON
*		Base URL* https://dxplay.hclsofy.com/dx/api/core/v1/dxrest/web	content
		Web Service Authentication One Basic NTLM	Identity Service for Backend Token ⑦ None
		> Advanced Description	
			CANCEL SAVE SAVE & ADD OPERATION

4. You first want to add an operation that gives you the list of product contents based on a specific Content Template and in a specific Site Area (using its parent ID) in DX. Give your new operation a name that reflects this, e.g.

searchWebContentByContentTemplateAndParentID. Set the Operation Security Level to Anonymous App Users and set the target URL to the search API:

/search?type=Content&authoringTemplateID=\$contentTemplateID&parentID=\$parentID

2	DXAssetsContentInte	grationSample 🖉 1.0	
F	Configure Services Manage	Client App Assets Publish	
57	Identity C Integration	\mathbb{P} Orchestration 🗍 Objects \mathbb{P} Workflow 🗐 Rules \mathbb{R} Engagement	
0	+ 🗋 🕫 🕲 Q	Service Definition Operations List search/WebContent@yContentTemplateAndParentID x	
	All Services	Name* searchWebContentByContentTemplateAndParentID Target URL https://dxplay.hclsofy.com/dx/ap/coreV1/dxrest/we /search?type=ContentBauthoringTemplateID=ScontentTemplateIDEparentID=SparentID Advanced Description	Operation Security Level ⑦ Anonymous App Users • Target HTTP Method ⑦ GET •
		Request Input Response Output Body Header + Add Parameter Copy © Paste Image: Copy	Enable pass-through: Input Body ⑦ Request Template ⑦ Show Hide

5. As done in the previous part for Assets, we should define the two parameters of the REST API URL, and we should provide some default value to them. In your DX instance, look up the UUID of the Site Area (parent ID), e.g. for Woodburn Studio Content – Products for all the product content items: 0407455c-68ed-455b-84c6-f916cedff649. Hover over it to have the pop up show.

Publishe	d Site 👻							奋	: :	ኛ _A English ▼	HCL	- í)
	Content Composer					Create v Q Sear	ch Title					\supset
	Woodburn Studio Content >	8	Content >	晟	Blogs	•		Produc	ct 08		•	
	Woodburn Studio Content French	8	Authoring Templates	晟	Exhibitions	•	8	Produc	ct 07		•	
	Woodburn Studio Content Spanish			晟	Products	>	8	Produc	ct 06		•	
8	Woodburn Studio Design			뮮	هه ر	Products Last modified: 2/8/2021 9:34 AM		ud	ct 05		•	
8	Woodburn Studio Extras			뮮	Type: Status:	Site Area Published		ud	ct 04		•	
	hclsoftware			뮮	UUID:	0407455c-68ed-455b-84c6-1 Copy UUID 🖆	916cedff	⁶⁴⁹ ud	ct 03		•	
	iodagroup			晟	Location: Created:	/ Woodburn Studio Content 8/8/2019 2:26 PM		uc	ct 02		•	
Items p	er page 100 • (i) < >			Items		wpsadmin		÷	100 🔻	í		

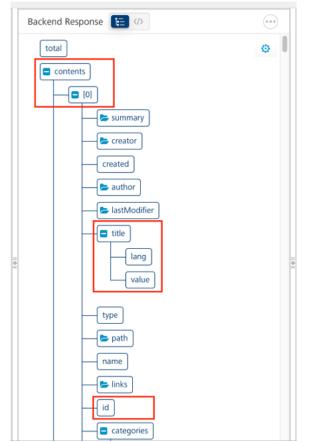
6. Also look up the UUID of a Content Template, e.g. for Woodburn Studio Design - AT-Product-Item: **36a33863-6ea3-4d44-8430-aa5287682aa2**.

Publish	ed Site 🔻						☑ ŵ ː ≭ _A English → <i>HCL</i> (Ì
	Content Composer				Create 🔻		Search Title
8	Woodburn Stores Content		Content	8	AT-Contact-Us	•	Name AT-Product-Item
	Woodburn Studio Content	品	Authoring Templates >		AT-Product-Item		Display Title
	Woodburn Studio Content French						AT-Product-Item
				6	AT-Page-Title	٠	Description None
	Woodburn Studio Content Spanish			8	AT-Social-Media-Item	•	Status
	Woodburn Studio Design >						Published
				8	AT-Social-Media	٠	000D Copy 000D C
	Woodburn Studio Extras			R	Component-Ref	•	Date modified
							2/8/2021 9:25 AM

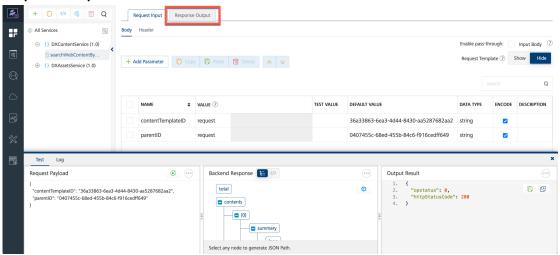
7. And set these as the Default Value for a new request parameters **contentTemplateID** and **parentID** using **Add Parameter**. Then click **SAVE AND FETCH RESPONSE**.

 Add Parameter 	Сор	/ 🕞 Paste	î Delete	* *			Request Tem		_
NAME	¢	VALUE ?			TEST VALUE	DEFAULT VALUE	DATA TYPE	ENCODE	DESCRIPTIO
contentTer	mplateID	request				36a33863-6ea3-4d44-8430-aa5287682aa	a2 string		
parentID		request				0407455c-68ed-455b-84c6-f916cedff649	string		

8. Check the Output Result and Backend Response. If you have a 401, you may not have correctly set up the security of the virtual resource WCM REST SERVICE, under the Resource Permissions where you should have edit access to. You may refer to HDX-DEV-100 Experience API lab to configure this properly. Then you want to add the **title** and **id** of each product to your output result. Have a look at the **Backend Response** and you will see that the title is addressed with **contents[*].title.value** and the id with **contents[*].id**.



9. In this case, you will manage these parameters under the Response Output directly. Click **Response Output**.



10. First add the **content** parameter and set it to **\$.contents[*]** and change this to **Collection**. This will make it become a list in the output.

Add Parameter	🖸 Сору	🕞 Paste	💼 Delet	e	~					Enable pass-throu	gh: Output
NAME	÷	PATH		SCOPE		DATA TYPE	COLLECTION ID	RECORD ID	FORMAT	FORMAT VALUE	DESCRIPTION
content	- [\$.contents[*]		respons	se .	collection 🝷			None		
						string boolean					
						number collection	1				
						record					

11. And then add a **title** parameter, **name** it **title**, set the **PATH** to **title.value** and set the **COLLECTION ID** to the collection you just created: **content**.

+ Ad	d Parameter	🖸 Сор	y 🕞 Paste 🛱 Dele	te 😞 😣					Enable pass-throug	h: Output 🤅
										Q
	NAME	¢	РАТН	SCOPE	DATA TYPE	COLLECTION ID	RECORD ID	FORMAT	FORMAT VALUE	DESCRIPTION
	content		\$.contents[*]	response	collection			None		
	title		title.value	response	string	content		None		
						None				
						content Path - \$.contents['	1			
							content	Path - \$.contents[*]		

12. And add the **id** parameter, **name** it **id**, set the **PATH** to **id** and again select **content** as the **COLLECTION ID**. Then click **SAVE AND FETCH RESPONSE**.

NAME \$	PATH	SCOPE	DATA TYPE	COLLECTION ID	RECORD ID	FORMAT	FORMAT VALUE	DESCRIPTION
content	\$.contents[*]	response	collection			None		
title	title.value	response	string	content		None		
id	id	response	string	content		None		

13. Your **Output Result** contains the list of content title and id. Now add an operation to get an individual content. Click **Operations List**.

۶.	DXAssetsContentIntegrationSample 🖉 💷							
i,	Configure Services Manage Client App Assets	Publish						
S1	M Identity	Objects	Workflow Rules	ngagement				
0		perations List search	WebContentByContentTemplat	eAndParentID ×				
۵	All Services () DXContentService (1.0) SearchWebContentBy Name* SearchWebContentBy.con	tentTemplateAndParen	tID				Operation Security Level ⑦ Anonymous App Users	•
æ,	(1) DXAssateSanica (1,0) Test Log							×
×	Request Payload 📀	Backend F	Response 🔚 🕪			Output Result		
	{ "contentTemplateID": "36a33863-6ea3-4d44-8430-aa5287682aa2", "parentID": "0407455c-68ed-455b-84c6-1916cedff649" }	con	tents (0) (1) (1) (1) (1) (1) (1) (1) (1		٥	6. "title 7. }, 8. { 9. "id":		

14. Click ADD OPERATION.

	+ 🖸 🛷 🚳 💼 Q	Service Definition Operations List	searchWebContentByContentTemplateAndPa	rentID ×		
	 All Services () DXContentService (1.0) g searchWebContentBy 	ADD OPERATION				
æŗ	() DXAssetsService (1.0)	Configured Operations				Q
X		□ NAME	URL	MODIFIED BY	MODIFIED ON	
F		searchWebContentByC	https://dx. pla	Herbert Hilhorst	06 Apr 2023 09:07 UTC	•••

15. Name it getProductItem as it is dedicated to your Content Template, set the Operation Security Level to Anonymous App Users and set the Target URL to /contents/\$contentID.

1	+ 🖸 🕫 🖄 🖬 Q	Service Definition Operations List getProductItem *	
₽₽	All Services	Name*	Operation Security Level (?)
			· · · · · · · · · · · · · · · · · · ·
A	OXContentService (1.0)	getProductItem	Anonymous App Use 👻
	getProductItem	Target URL	Target HTTP Method ?
([])	searchWebContentBy		
\smile	, ···	https://dx.: play.hclsofy.com/dx/api/core/v1/dxrest/we /contents/\$contentID	GET 🚽

 Then find the UUID for your default content, e.g. Product 01 under Woodburn Studio Content – Products: 4282f9fd-799d-48fb-a49b-65c9598016cd. Hover over Product 01 to get the pop up to copy the UUID.

Published Site 💌				\square \triangle : x_A English + HCL ()
Content Composer			Crea	te ▼ Q Search Title
B Woodburn Studio Content >	Content >	Blogs	•	Product 07 • Details
Woodburn Studio Content French	Authoring Templates	Exhibitions	•	E Product 01 Last modified: 2/8/2021 9:25 AM
Woodburn Studio Content Spanish		R Products	>	I Type: Content
Woodburn Studio Design		Our Projects	•	Status: Published UUID: 4282/9(4-799d-48/b-a49b-65c9598016cd Copy UUID 12
Woodburn Studio Extras		About-Us	•	ELocation: / Woodburn Studio Content / Products
hclsoftware		Contact Us	•	Created: 8/8/2019 2:41 PM
iodagroup		Home	•	Product 01 : Product 01
Items per page 100 * (i) < >		Items per page 100 👻	< > 10	ems per page 100 * (j) < > Display Title Product 01

17. Then add the request parameter **contentID** using **Add Parameter** and set the **Default Value** to the UUID of your content item, in this case **4282f9fd-799d-48fb-a49b-65c9598016cd**. Click **SAVE AND FETCH RESPONSE**.

+ Add Parameter	🖸 Сору	Paste	Delete	*	♦		Requ	est Template 🕐	Show	Hide
										Q
NAME	÷	VALUE ?				TEST VALUE	DEFAULT VALUE	DATA TYPE	ENC	DESCRIP
contentID		request					4282f9fd-799d-48fb-a49b-65c9598016	d string		
Default value will be used if	Test value is e	mpty.					m10000(SAVE AND	D FETCH RE	SPONSE
								CANCEL	SAVE OPI	ERATION

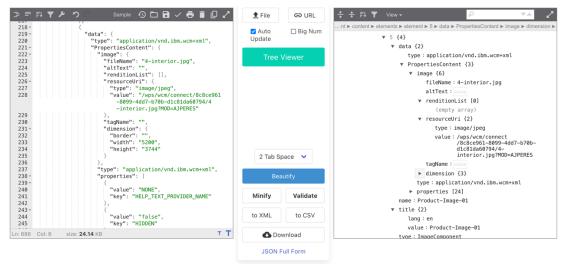
18. Now have a look at the JSON payload. Open the source view.



19. Then you may want to expand it or copy it to paste and see it in your preferred JSON beautifier/viewer, like https://codebeautify.org/jsonviewer.



20. To enable your operation to return all the **Product Title**, **Designer Name**, **Description** and **images** belonging to this Woodburn Studio product content, you need to understand the JSON format. For example, you see that the Product-Image-01 element is the element number 5 and has the name and data nodes.



21. To understand the structure of this JSON, use the right panel of

https://opensource.hcltechsw.com/experience-api-documentation/ring-

<u>api/#operation/webContentReadContent</u>, where you can expand the path to see that the elements are under a **content – content - elements node** and each **element** has a **name**, **title**, **displayTitle**, **type** and **data** node. Depending on the element type, the data will look different. Here you see the sample structure of an image with its renditions expanded, using renditionList.



22. You may use a JSON Path Tester, like <u>https://codebeautify.org/jsonpath-tester</u>, to find the paths for each of your elements. You may do a lookup for each element using its name in the following path **\$.content.content.elements.element[?(@.name=='<element name>')]**, e.g. for Product-Name this would be \$.content.content.elements.element[?(@.name=='Product-Name')]. In this way, you may find the paths to product title, designer name, description, and images.

Code Beautify	JSON Formatter Hex Color Codes HMAC Generator My Ip Search Recent Links More * Sign in
JSON Path Tester☆	Save & Share
JSON Path TesterX	Result : Generated JsonPath Cart 1 - [{ "data": { "data": { ""cype": "text/plain", ""ropoprtiesContent"; { ""ropoate: "product One", ""ropoate: "product One", ""key": "HELP_TEXT_PROVIDER_NAME" "key": "HELP_TEXT_PROVIDER_NAME" "key": "HELP_TEXT_PROVIDER_NAME" * "key": "HELP_TEXT_PROVIDER_NAME" * "key": "HELP_TEXT_PROVIDER_NAME" * *
\$.content.content.element/@.name=='Product-Name'] TEST JSON PATE RESET	27 28 "name": "Product-Name", 29 ttte: t 30 "lang: "en", 31 "vale: "Product-Name" 32 }, 33 "tupe": "ShortTextComponent"

23. The images are a bit special, as they have a source image, accessible with \$.content.content.elements.element[?(@.name=='<Image Element Name>')].data.PropertiesContent.image.resourceUri.value and different renditions that may be retrieved easily as well using \$.content.content.elements.element[?(<Image Element Name>)].

data.PropertiesContent.image.renditionList[?(@.name=='<rendition>')].resourceUri.value. See the table with element names and corresponding paths.

Element Name	Path						
Product-Name	\$.content.content.elements.element[?(@.name=='Product-Name')].data.PropertiesContent.value						
Designer-Name	\$.content.content.elements.element[?(@.name=='Designer-						
	Name')].data.PropertiesContent.value						
Description	\$.content.content.elements.element[?(@.name=='Description')].data.PropertiesContent.value						
Product-Image-01	Source: \$.content.content.elements.element[?(@.name=='Product-Image-						
	01')].data.PropertiesContent.image.resourceUri.value						
	Desktop : \$.content.content.elements.element[?(@.name=='Product-Image-						
	01')].data.PropertiesContent.image.renditionList[?(@.name=='desktop')].resourceUri.value						
	Tablet : \$.content.content.elements.element[?(@.name=='Product-Image-						
	01')].data.PropertiesContent.image.renditionList[?(@.name=='tablet')].resourceUri.value						
	Smartphone : \$.content.content.elements.element[?(@.name=='Product-Image-						
	01')].data.PropertiesContent.image.renditionList[?(@.name=='smartphone')].resourceUri.value						
Product-Image-02	Source: \$.content.content.elements.element[?(@.name=='Product-Image-						
	02')].data.PropertiesContent.image.resourceUri.value						
	Desktop : \$.content.content.elements.element[?(@.name=='Product-Image-						
	02')].data.PropertiesContent.image.renditionList[?(@.name=='desktop')].resourceUri.value						
	Tablet : \$.content.content.elements.element[?(@.name=='Product-Image-						
	02')].data.PropertiesContent.image.renditionList[?(@.name=='tablet')].resourceUri.value						
	Smartphone : \$.content.content.elements.element[?(@.name=='Product-Image-						
	02')].data.PropertiesContent.image.renditionList[?(@.name=='smartphone')].resourceUri.value						
Product-Image-03	Source: \$.content.content.elements.element[?(@.name=='Product-Image-						
	03')].data.PropertiesContent.image.resourceUri.value						
	Desktop : \$.content.content.elements.element[?(@.name=='Product-Image-						
	03')].data.PropertiesContent.image.renditionList[?(@.name=='desktop')].resourceUri.value						
	Tablet : \$.content.content.elements.element[?(@.name=='Product-Image-						
	03')].data.PropertiesContent.image.renditionList[?(@.name=='tablet')].resourceUri.value						
	Smartphone : \$.content.content.elements.element[?(@.name=='Product-Image-						
	03')].data.PropertiesContent.image.renditionList[?(@.name=='smartphone')].resourceUri.value						

24. Go to your **Response Output** and start creating the **productName**, **designerName** and **description** parameters by using the path defined above for the images, you may manage them into a collection of elements (images). Create a collection of elements (images) that has a data.ProperiesContent.image defined, using

\$.content.content.elements.element[?(!!@.data.PropertiesContent.image)]. And then, for each image, add the name with name, entries of the sourceUri values for source using data.PropertiesContent.image.resourceUri.value, and the desktop, tablet and smartphone renditions (if they exist, which is not the case by default for the Woodburn Studio Products) for each image, using

data.PropertiesContent.image.renditionList[?(@.name=='<rendition>')].resourceUri.value. The next time you test the operation you will see the list of image and other elements with values in the Output Response. To re-test, click the green **play** icon located left Test Log panel.

*	+ 0 0 5 1 Q ⊗ All Services ⊕ 1 DXAssetsService (1.0) ⊖ 1 DXContentService (1.0)		dd Parameter				Ena	ble pass-through:	Output ③
	getProductItem								
Θ	searchWebContentBy		NAME \$	РАТН			SCOPE	DATA TYPE	COLLECTION
\bigcirc			productName S.content.content.elements.element[?(@.name=='Product-Name')].data.PropertiesContent.value				response	string	
₽₽₽		designerName \$.content.content.elements.element[?(@.name=='Designer-Name')].data.PropertiesContent.value					response	string	
0-0-1 1	description \$.content.elements.element[?(@.name=='Description')].data.PropertiesContent.value n						response	string	
×	images \$.content.elements.element[?(II@.data.PropertiesContent.image)]				response	collection			
R	name name				response	string	images		
1 6	data.PropertiesContent.image.resourceUri.value				response	string	images		
		data.PropertiesContent.image.renditionList[?(@.name=='desktop')].resourceUri.value					response	string	images
			tablet	data.PropertiesContent.image.renditionList	?(@.name=='tablet')].resourceUri.value		response	string	images
			smartphone	data.PropertiesContent.image.renditionList	?(@.name=='smartphone')].resourceUri.	value	response	string	images
	Test Log	Default	value will be used if Test value is er	mpty.		m10000062	5002 -	SAVE AND	FETCH RESPONSE
	Request Payload		6	Backend Response 🔚 🕫	Outpu	t Result			
	{ *contentID*; *4282f9fd-799d-48fb-a49b-4 }	65c9598	016cd*	summary Lang creator for anne Select any node to generate JSON Path.	() () () () () () () () () ()	"smartj "name": "source	t": "", pp": "", phone": "", : "Product-I =": "/wps/wc		© (2) ce961-8099-4dd7- AJPERES"

25. Then, like with the DXAssets, you need to add the host to allow you create a full URL to your images. Click **Advanced**, select **JavaScript** and paste this code, updated with your own host URL. Click **SAVE AND FETCH RESPONSE** and check it works.

var tmpDXHost = "https://dx.sbx0000.play.hclsofy.com";

result.addParam("DXHost", tmpDXHost);

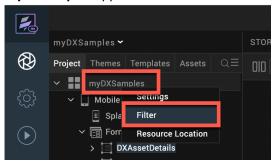
	+ C C> S C Q All Services B G {} DXAssetsService (1.0) G getAsset G {} DXContentService (1.0) G getProducttem G searchWebContentBy	✓ Advanced Custom Code Invocation Stub Backend Response Properties Front End API Preprocessor Java JavaScript Rules Class Select a Class ✓	Pass-through Cookies Server Events Postprocessor ⑦ Jav Java JavaScript Rules var tmpDXHost = "https://dx. play.hclsofy.com"; result.addParam("DXHost"; tmpDXHost;
) * *		No class Available	
	Test Log Request Payload	Description Image: Constraint of the second secon	x Output Result
	{ "contentID": "4282f9fd-799d-48fb-a49b }	->-65c9598016cd*	 Consume the second secon

26. Then republish your Foundry application **DXAssetsContentIntegrationSample**. Click the **Publish** tab, select your runtime and click the **Publish** button.

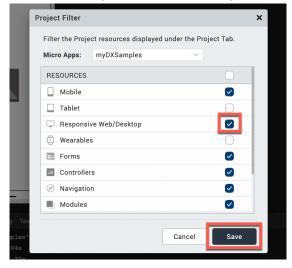
	DXAssetsContentIntegrationSam	nple 🖉 1.0 🗸			
F	Configure Services Manage Client App A	ssets	Publish		
API	Service & Web Client Vative Client				
	m10000		DEFAULT VERSION		A A A A
	CONSOLES	SERVICE CONFIG	APP STATUS Published 05 Apr 2023 07:41 UTC	ENVIRONMENT STATUS (?) Available	HISTORY & ROLLBACK
X			U	NPUBLISH CONFIGURE &	PUBLISH PUBLISH

27. You may also use the HCL Digital Experiene Content Adaptor. This includes all endpoints of the WCM REST V2 APIs. You can download this from the Volt MX Marketplace <u>https://marketplace.hclvoltmx.com/items/hcl-dx-content-adapter</u>. Details on how to download, import and start using are described here <u>https://opensource.hcltechsw.com/digital-experience/latest/extend_dx/integration/mx/example/dx_apis_in_foundry/</u>.

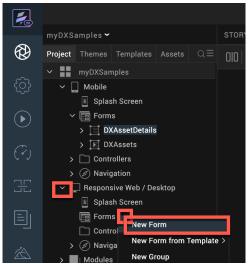
28. Now switch to your Volt Iris IDE to create your web application. In your Iris project **myDXSamples**, you need to unhide the Responsive Web /Desktop first, as this help you creating the right resources more easily for your web application. For that, right click on your **myDXSamples** application and then **Filter**.



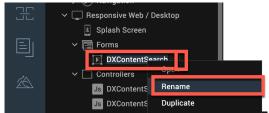
29. Then check Responsive Web/Desktop and click Save.



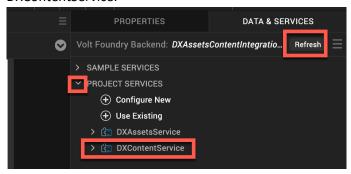
30. Then expand the **Responsive Web / Desktop**, right click on **Forms** and click **New Form** to create a form that allows you to search the right content first.



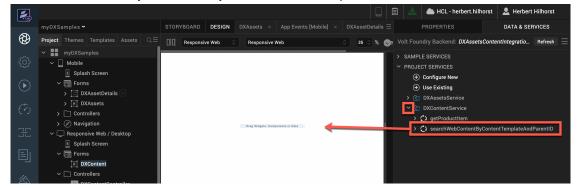
31. Then rename it, e.g. to **DXContentSearch**.



32. This application is already associated with the Foundry services you have created before. As you have updated the Foundry services, you need to refresh the Foundry association. Under DATA & SERVICES, click **Refresh** and expand **PROJECT SERVICES** to see your DXContentService.



33. Expand DXContentService to see your operations and drag & drop searchWebContentByContentTemplateAndParentID operation into the form.



34. Select **Entry form for Request** when asked and **OK**, as this allows you to select the Content Template and Parent ID of the content you want to show.

				> 💭 getProductItem		
) Na iespo Sp		What would you	like to generate?		×	Temp
For Co Js Js Na Na Iodu						
ntain rollCu	List & Details Using Response	Grid Using Response	Details Using Response	Entry form for Request		
ne ar		rorm (TV)spatel		Сапсе		

35. This adds a flex container with your input parameters and a button. Then update this button. Click your button, then PROPERTIES, and update under Look – Appearance the text, e.g. to Search Web Content and the width, e.g. to 300 Dp. Feel free to make other changes as well.

1				🗒 📩 📥 нс	L - herbert.hilhorst 👤 Herbert Hilhorst
		STORYBOARD DESIGN (Assets × App Events [Mobile] ×	DXAssetDetails × DXContentSearch* ×		DATA & SERVICES
\otimes	Project Themes Templates Assets Q≡	Responsive Web 🗘 Responsive Web	ି 😽 ି % Breakpoint Forking 🔍 🕻	Look Skin Butto	n Action Review
~	> Controllers	0	Wilcome, User name	✓ General	
(j)	> Ø Navigation		Wecces, user rans		btnSquared
~	Responsive Web / Desktop	ContentTemplateID		Visible	💿 On 🜑 Off
	Splash Screen			Render	Edit
	✓ E Forms ✓ DXContentSearch ✓	ParettD		_	EUIT
C)	→ FlexHeader			 Appearance 	
	imgHamburgerMenu			സ്റ്റ് Content Align	
38	> flexRight				Search Web Content
	✓ ☐ flexFieldsContainer			ଂଲ୍ଙ Display Text	● On ● Off
	> [] flexcontentTemplateID			✓ Flex	
	> 🗌 flexparentID				
×	✓ ☐ flexFooter			Widget 🗦	
222	✓ ☐ flexSquaredBorderedButton			സ് Left 1	s ⊖ °C° Right Defac⊖
	btnSquared			ాధి Тор	Defat ○ °℃ Bottom 2 % ○
	✓ ☐ Controllers			℃ Width 300	Dp 🔿 👓 Height 65 % 🔿
	Js DXContentSearchController			: 🕆 Min Width 27	
	Default Library My Libraries Q	Search Web Content			
	Buf with			ന്ന Height	Defat⊖ °℃° Max Height Defat⊖

36. Then you want to show the results on the same form. Select the form, click DATA & SERVICES and drag your searchWebContentTemplateAndParentID operation to any place in the form, except for your existing flex container.

1					📥 HCL - herbert.hilhorst	💄 Herbert Hilhorst
		STORYBOARD DESIGN (Assets ×)		DXContentSearch* \times \equiv	PROPERTIES	DATA & SERVICES
⊗	Project Themes Templates Assets $Q\equiv$	00 Responsive Web C Responsive	e Web 0 44 0 % B	reakpoint Forking 🕥 🦁 Volt Fo	oundry Backend: DXAssetsConter	tIntegratio Refresh 📃
ŝ	Controllers Ø Navigation Responsive Web / Desktop	0	Webore, Der rame	· Logoxt · PRO	IPLE SERVICES	
∢	 Splash Screen Em Forms 	ContentTemplateD			 Configure New Use Existing DXAssetsService 	
$(\vec{\gamma})$	✓ ► DXContentSearch ○ ✓ ☐ flexHeader			~ (DXContentService O getProductItem	
H	 ➢ imgHamburgerMenu > ☐ flexRight > ☐ flexFieldsContainer 		-		> 🗘 searchWebContentByConten	TemplateA View Mapping
Ej	A secondaria A secondaria					
źA.	✓ ☐ flexFooter ✓ ☐ flexSquaredBorderedButton					
	btnSquared Controllers					
	Controller Default Library My Libraries	Search Web Content				

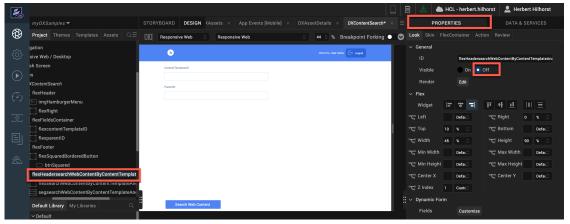
37. Now select a Grid Using Response and click OK.

					🔜 💭 getProduc	ctitem
nt ht Co ite		What would you	like to generate?			×
en Jar Sqi			==			
ear es						
	List & Details Using Response	Grid Using Response	Details Using Response	Entry form f	or Request	
				Cance	ок	
	Conning Form IDVA	ant al				

38. Configure this flex container to show correctly on the right of the form. Click PROPERTIES and update the Flex parameters, Pin to right for the Widget setting, Left to Default, Right to 0%, Top to 10%, Width to 45% and Height to 90%.

1						🗒 📥 🖶	CL - herbert.hil	horst 💄 Herbert Hilhorst
		STORYBOARD DESIGN (Assets × App Even		AssetDetails × DXContentSearch	* × Ξ	PROPER	TIES	DATA & SERVICES
\otimes	Project Themes Templates Assets $Q\equiv$	00 Responsive Web 🔅 Responsive Web		ି 😽 🎸 🖇 Breakpoint Forking	• ©	Look Skin Flex	Container Acti	ion Review
~~	avigation			Wetcome, User name		✓ General		
- Cộ	ionsive Web / Desktop		_	wecome, user name Err Logout	- 1		flexHeaderse	archWebContentByContentTemplateAn
\sim	plash Screen	ContentTemplateID	TEle	1d		Visible	💽 On 💮 🔘	Dff
	orms		Title	ld Hd	- 10	Render	Edit	
	DXContentSearch	ParentiD	Title	Ni di			Edit	
(~)	flexHeader		Tão Tite	id M	- 11	✓ Flex		
	ImgHamburgerMenu IngHamburgerMenu IngHamburgerMenu		Tão	id id		Widget	: 🛨 🖃	표 제 표 비 표
Æ	flexFieldsContainer		Tite	N N	- 10	℃ Left	Defat	°C; Right 0 % ்
	> flexcontentTemplateID		Tite	N .	1	°℃ Top 10	8.0	°℃ Bottom Defac
Ξı	> flexparentID		Tão	N			8 0	
	flexFooter							
A	→ ☐ flexSquaredBorderedButton					°് Min Width	Defat	ଂପ୍ତ୍ର Max Width Defatି
	🗆 btnSquared					°℃ Min Height	Defac	ଂପ୍ତ୍ର Max Height Defaเ
	flexHeadersearchWebContentByContentTemp	1				°८° Center X	DefatO	ଂପ୍ତ Center Y Defatି
	flexsearchWebContentByContentTemplate					°℃° Z Index 1	Custc	
	> segsearchWebContentByContentTemplate.							
	Default Library My Libraries Q	Search Web Content						
	∽ Default					Fields	Customize	

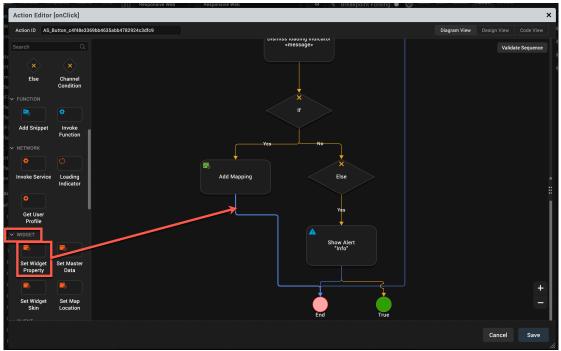
39. Then hide this flex container, until you have done a search. Ensure you have selected the container and then under **PROPERTIES**, switch **Visible** to **Off** and notice it disappears.



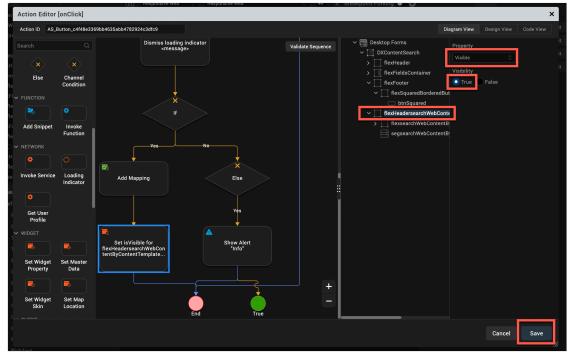
40. Then change the button to show the flex container when you start the search. Select the **button** and then under **PROPERTIES – Action**, click **Edit** for the **onClick** event.

1				📋 📩 🌰 HCL - herbert.hilhorst	👤 Herbert Hilhorst
		STORYBOARD DESIGN (Assets × App Events [Mobile] >	DXAssetDetails × DXContentSearch* ×		DATA & SERVICES
Ø	Project Themes Templates Assets $Q\equiv$	Responsive Web	🗧 🕴 🖇 Breakpoint Forking 🔵	🛇 Look Skin Button Action Review	
~~	gation		Wetcome, User name	✓ General	
ŝ	sive Web / Desktop		Watcone, User name	onClick AS_Button_c4f48e3369	o Edit 🖉
\sim	sh Screen	ContentTemplateID		onTouchStart Click Edit to add actions	Edit O
(\mathbf{b})	is XContentSearch			onTouchMove Click Edit to add actions	Edit P
~	flexHeader	ParentD		onTouchEnd Click Edit to add actions	
(?)	ingHamburgerMenu			on rouchend Check ear to add actions	
	flexRight				
35	flexFieldsContainer				
	flexcontentTemplateID				
	flexparentID				
_	flexFooter				
×	flexSquaredBorderedButton				
	btnSquared				
	flexHeadersearchWebContentByContentTemplat				
	flexsearchWebContentByContentTemplateAnc				
	segsearchWebContentByContentTemplateAnc				
	Default Library My Libraries	Search Web Content			

41. Scroll down to see the Add Mapping node and then drag the **Set Widget Property** node under **Widget** category to the **blue connector** between **Add Mapping** and **End** nodes.



42. Then, select **flexHeadersearchWebContentByContentTemplateAndParentID** flex container and ensure the **Property** is set to **Visible** and **Visibility** to **True**. Click **Save**.



43. Then add a form for the details of each product. Right click **Forms** and **New Form**.



44. And rename it to **DXProductDetails**.



45. Then create a global variable to pass the Content UUID and have easy access to its details in the DXProductDetails form and one to access the DXHost. Click **Edit** and then **Global Variables**.

é '	Volt MX Iris Proje	ct Edit Build Debug Forge	Window He
••		Undo	жz
		Redo	<u> </u>
	myDXSamples 🗸	Cut	ж Х 🔀
	Desired Therese T	Сору	жс
N S	Project Themes Te	Paste	ж V 🗘 🗍
~~~	∽ 🔄 DXConten	tSe Delete	
ද්ටා	∽ ่ flexHea ™ img	Deletast tult totationte	τQ
$(\mathbf{b})$	> [] flex	Caarab	ራ <del>ዘ</del> F
	✓ ☐ flexFie > ☐ flex	Select All	жA
(~)	> flex		
	✓ ☐ flexFoo ✓ ☐ flex	Manage Cordova Plugins	
		Manage Native Function API(s)	
E	⊂ b ∨ ∏ flexHea	Manage Web Libraries	
	> [] flex		
7.4	> seg	Start Dictation	.0,

46. Then add two the variables **gbIDXContentID** and **gbIDXHost**, and **close** it.

Variables		×
Type: Simple	\$	
+ × ↑ ↓		
Variable	Default Value	
gbIDXAssetURL		
gbIDXContentID		
gbIDXHost		

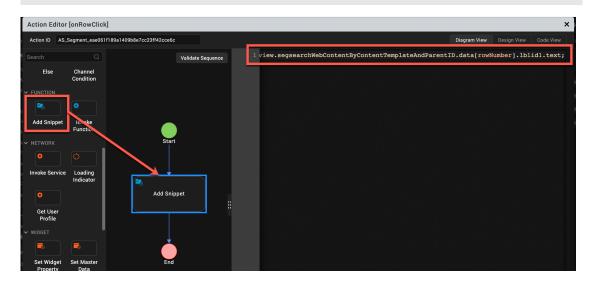
47. Then set this variable when you click a row in your search list of content result. Under the search result form, select the segment

segsearchWebContentByContentTemplateAndParentID, click PROPERTIES and Edit for the onRowClick action.

1						🗄 🗻	📥 HCL - herbert.hilhorst	👤 Herbert Hilhorst
		STORYBOARD DESIGN ×		DXAssetDetails × DXContentSea	rch ×	≡ Р	ROPERTIES	DATA & SERVICES
$\otimes$	Project Themes Templates Assets $Q\equiv$	00 Responsive Web	Responsive Web	⊖ 41 ☉ % Breakp	oint Forki	Dook Skin	Segment Action Review	
~~~	✓					✓ General		
ණ	✓ ☐ flexHeader	۵		Welcome, User name		onRowCli	ck Click Edit to add actions	Edit
~	🦮 imgHamburgerMenu	Content Templatel D				onTouchs	Start Click Edit to add actions	Edit
$\mathbf{\bullet}$	> flexRight						Nove Click Edit to add actions	
	→ ☐ flexFieldsContainer	ParentD						
(?)	flexcontentTemplateID flexparentID					onTouchE	nd Click Edit to add actions	Edit 🖉
~ ~	FilesFooter							
æ	→ flexSquaredBorderedButton							
	btnSquared 🖓							
Ej	✓ ☐ flexHeadersearchWebContentByC							
	→ flexsearchWebContentByConte							
×	> segsearchWebContentByConte							
	DXProductDetails							
	✓ □ Controllers							

48. Use a snippet to set the new global variable to the content ID in the row. Drag the **FUNCTION – Add Snippet** between **Start** and **End** and add the follow code:

gblDXContentID=self.view.segsearchWebContentByContentTemplateAndParentID.data[ro wNumber].lblid1.text;



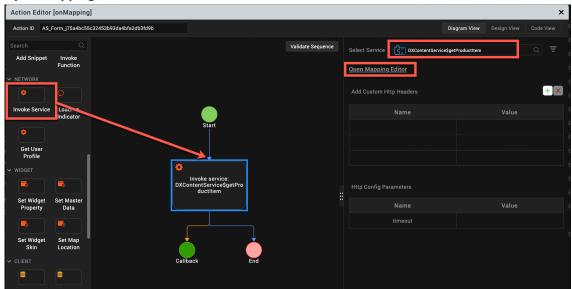
49. Then add a navigation to your new form. Add a **GENERAL** – **Navigate to From** widget between your **Add Snippet** and **End** and set it to your new **DXProductDetails** form. Then click **Save**.



50. Now create the mapping for the form. Select your new form **DXProductDetails** and click **Edit** under **PROPERTIES – Action – onMapping**.

1							Ē	≟ 🌢	HCL - herbert.hilhorst	👤 Herbert Hilhorst	
		STORYBOARD	DESIGN bile] ×			DXProductDetails ×	=	PROP	ERTIES		
$^{\odot}$	Project Themes Templates Assets $Q\equiv$	CONTRACTOR	ve Web 🔅 🛛 Re	sponsive Web	 ↓ 42 	% Breakpoint Forking			orn Action Review		
~~~	btnSquared						~ G	General			
ŝ	✓ ☐ flexHeadersearchWebContentByConte								Click Edit to add actions	Edit	0
$\bigcirc$	flexsearchWebContentByContentTe     segsearchWebContentByContentTe							nMapping	Click Edit to add actions	Edit	Ð
$(\mathbf{b})$	DXProductDetails						pr	reShow	Click Edit to add actions	Edit	0
<i>C</i> .	Controllers						р	ostShow	Click Edit to add actions	Edit	0

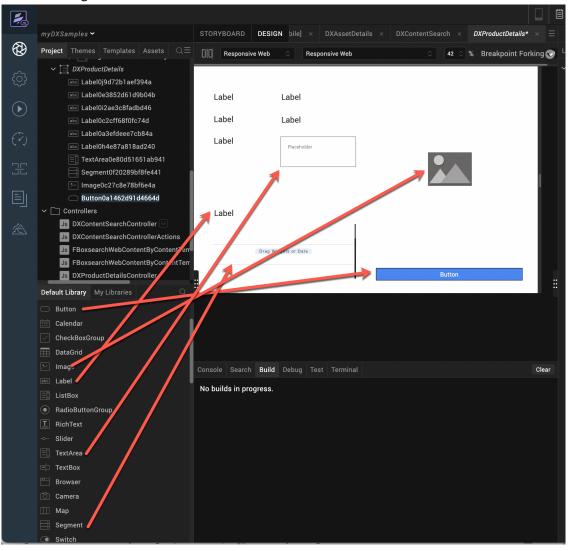
51. Drag the NETWORK – Invoke Service between Start and End and set it to DXContentService\$getProductItem operation to get the details of your content. Then click Open Mapping Editor.



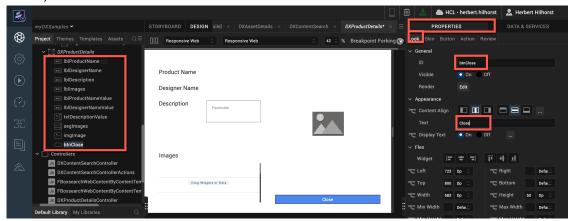
52. Then select your Variable - gblDXContentID and link it to the Services – Service – DXContentServices\$getProductItem – contentID. Click Save.

Actio	on Editor	[onMapping	1]											×
Actio	n ID AS_	Form_j75a4bc5	5c32453b93da4bfa2db3fd9b								Diagram View			
Searc						Validate Seque	ence							
Add	Snippet	Invoke Function						> 🗐 I				> 🖪 Fi		
✓ NET	NOBK			Star					/ariables			> 🖬 V		
0									□ gbIDXAsse □ gbIDXCon1	_		~ [] s		DXContentS
	nvoke	Loading					-	> 🗐 🤅		entid (stri		_		ntID {string}
	ervice	Indicator		<b>\$</b>					Services					(enny)
•				Invoke se DXContentServ	rvice:									
	et User			ductite	em		:							
١	Profile													
				, <u>↓</u>	<u> </u>									
	Widget operty	Set Master Data					+							
'				Callback	End		-							
, C	Widget	Cat Man								_	_			
												Canc	el :	Save

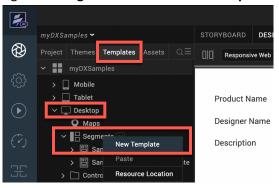
53. Then build your form using the widgets from the palette. To render a collection, you created to manage your image, you may use a segment. You may create something like using **Labels**, a **TextArea**, a **Segment**, an **Image** and a **Button**. You will use the labels to display the text elements, a text area for the description (as this may be multiple lines), a segment for the list of images (source for now, but you may also add renditions later), an image to render the selected image and a button to navigate back to the search page. Drag them to your form from the Widgets Palette.



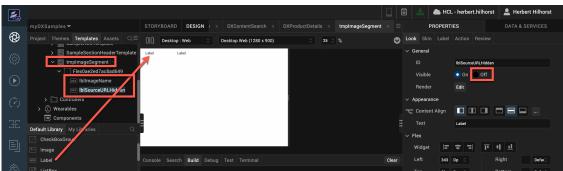
54. Then update the IDs from the autogenerated ones to make them more context specific as well as the text value of the labels, when they have a constant value. Use the prefixes "Ibl" for labels, "txt" for the text area, "seg" for the segment and "btn" for your button. For the values, clear the text. It should look like this:



55. Then define a template for the segment for the images that may render the title of the image and set the right URL. Do this for the desktop now. Click **Templates**, open **Desktop**, right click **Segments** and click **New Template**.



56. Rename the Template to something more easily readable, like **tmpImageSegment**. This template will store the image name (visible) and the image URL (hidden). Then, add a label for each of them, **IbIImageName** and **IbISourceURLHidden**. Clear the text of **IbImageName** and hide IbISourceURLHidden, by clicking **Visible Off**. When this is set, your template looks empty again. In this case, you will only show the source image. You are making the change now for the desktop. You may repeat these steps for the Tablet and Smartphone renditions as well.



57. Then update your form with this new template. Select your **DXProductDetails** form and then your **segImages** segment and under **Segment** select your new **tmpImageSegment Row Template**.

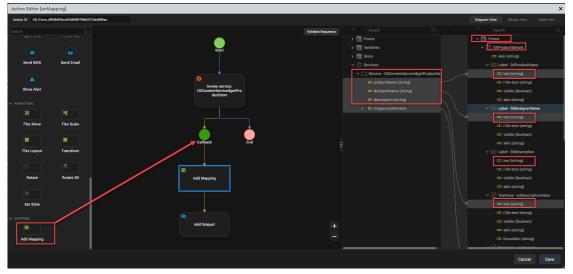
1					📥  📥 HCL - herber	rt.hilhorst 💄 Herbert Hilh	iorst
		STORYBOARD DESIGN bile] × DXAssetDetails >	DXContentSearch × DXProductDet	ails* $\times$ $\equiv$	PROPERTIES		
$\otimes$	Project Themes Templates Assets $Q\equiv$	Responsive Web	ି ଓ ଓ ଅନେକ୍ଷ Sreakpoin	t Forking 🜏 🛛	.ook Skin Segment Acti		
ŝ	Contentary Sease archWebContentByContent     DXProductDetails      Intent IbiProductName			`````````	<ul> <li>General</li> <li>Master Data</li> </ul>	Edit	
$(\mathbf{b})$	IblDesignerName	Product Name			Row Template	tmpImageSegment 🔅	Edit
$\mathbf{\Theta}$	IblDescription	Designer Name			Section Header Template	None 🗘	Edit
(?)	बग्द Iblimages 🗹 बग्द IblProductNameValue	Description			Swipe Config	Edit	
~ ~	abe: IbIDesignerNameValue	C TRANSPORT			Group Cells	🔵 On 💿 Off	
Æ	E txtDescriptionValue				Retain Selection	🔵 On 💿 Off	
	segImages				Separator	💿 On 🔵 Off	
	imgImage ◯ btnClose	Images			Separator Thickness	•	
	✓ ☐ Controllers				Separator Color		
Ŕ	Default Library My Libraries				Separator Transparency	•	
	CheckBoxGroup				Show Scrollbars	🔵 On 💿 Off	
	📰 DataGrid		Close		View Type	Table	
	Image						

58. Now update your form mapping. Click **DXProductDetails** form, **PROPERTIES – Action** and the **onMapping Edit**.

1					📥 HCL - herbert.hilhorst	👤 Herbert Hilhorst
		STORYBOARD DESIGN bile] × DXAss		DXProductDetails $\times$ $\equiv$	PROPERTIES	DATA & SERVICES
⊗	Project Themes Templates Assets Q ≡ > flexsearchWebContentByConten	Image: Construction of the second	eb 🗘 39 🗧		ok Skin Form Action Review	
ŝ	segsearchWebContentByConten     Time DXProductDetails				General init Click Edit to add actions	B Edit P
ightarrow	at IblProductName	Product Name			onMapping AS_Form_j75a4bc55c32	24 Edit 🖉
	nter IbIDesignerName	Designer Name			preShow Click Edit to add actions	Edit 🖉

59. Then add a MAPPING – Add Mapping node on top of the Callback. It will show below. And then map the first text fields under Forms – DXProductDetails - productName, designerName and description under Services – Service –
DXCentration Cast Descharts the service and is a text of the value labels under service.

DXContentService\$getProductItem to the corresponding text of the value labels under Forms – DXProductDetails.



60. Map **the images** collection to the **Segment – masterdatamap – rowData** collection first and then map the images **name** and **source** to the corresponding text of the labels.

Action Editor	[onMapping]					×
Action ID AS_F	orm_j75a4bc55c324	53b93da4bfa2db3fd9b			Diagram View	Design View Code View
Search		Start	Validate Sequence			= Search Q
<b>,89</b> ,	89.			> 📰 Forms		E TextArea - txtDescriptionVa
Flex Move	Flex Scale			> 📰 Variables		Segment - segImages
		•		> 🧰 Skins		✓ □□ masterdatamap
	<b>X</b>	Invoke service:		✓ ☐ Services	7	✓
Flex Layout	Transform	DXContentServiceSgetPro ductitem		✓  ☐ Service - DXContentServ		✓ 🔤 Label - IblimageNa
		uutitein .		🚥 productName {string}		💷 text {string}
C				🚥 designerName {string		🚥 i18n-text (string
Rotate	Rotate 3D			description {string}		🚥 visible (boolean
notate	notate ob	↓ ↓		✓		🚥 skin (string)
				🚥 name {string}		✓ abc Label - IblSourceUI
		Callback End		🚥 source (string)		🚥 text {string}
Set Style				🚥 desktop (string)		🚥 i18n-text (string
✓ MAPPING				🚥 tablet {string}		🚥 visible (boolean
<b>2</b>		<b>E</b>	+	🚥 smartphone {strinç		🚥 skin (string)
		Add Mapping	-			> 🚥 metalnfo
Add Mapping			_			visible (boolean)
						Cancel Save

61. Then add an **Add Snippet** widget to set the global variable DXHost. Drag it to **Add Mapping** and enter:

Then	click <b>Save</b>								
Action Ed	itor [onMapping]								×
Action ID	AS_Form_j75a4bc55c324	53b93da4bfa2db3fd9b				_	Diagram View	Design View	
Search IT Else	Channel Condition		LALOMENTSERVICE-SGEIFFO ductitem Caliback	End	Validate Sequence		gblDXHost=getProductItem.DXRost;		
Add Snip	Functio		Add Snippet						
Get Use Profile			Add Mapping		+				
								Cancel	Save

gblDXHost = getProductItem. DXHost;

62. Then update the segment to render the image in the form, once one of the images are clicked. Select your **images** segment, click **PROPERTIES** – **Action** and then **Edit** on the **onRowClick** event.

1						Archart HCL - herbert.hilhorst	💄 Herbert Hilhorst	
		STORYBOARD DESIGN App Eve		DXContentSearch × DXProductDetails		PROPERTIES		
$\otimes$	Project Themes Templates Assets $Q\equiv$	Responsive Web 🔅 Res	esponsive Web	39 🛇 % Breakpoint Forking 🔵	S Look	Skin Segment Action Review		
~~	✓				∨ Gen	eral		
ŝ	iblProductName				onRo	wClick Click Edit to add actions	Edit	
_	IblDesignerName	Product Name			onTo	uchStart Click Edit to add actions	Edit	
	inc IblDescription	Designer Name			onTo	uchMove Click Edit to add actions		
	IblImages     IblProductNameValue							
(?)	Interfolductivamevalue	Description Placebolder			onic	uchEnd Click Edit to add actions	Edit	
	TxtDescriptionValue							
H	segimages S							
	💹 imgImage	Images						
	🗀 btnClose							
_	✓ ☐ Controllers							
×	Ja DXContentSearchController							
	Default Library My Libraries							
	CheckBoxGroup		Close					
	DataGrid							

63. Then set the image URL to the one selected in your segment. Add a **FUNCTION – Add Snippet,** set the value using the code below and click **Save**.

self.view.imgImage.src=gbIDXHost+self.view.segImages.data[rowNumber].lbISourceURLHid
den.text;

Action	n Editor [onRow	vClick]						×
Action	ID AS_Segment_i	b4572e56040f40d2932e2b2f35195f84				Diagram View	Design View	Code View
Search				Validate Sequence	<pre>1 .ew.segImages.data[rowN</pre>	umber].lb]	lSourceURLH	lidden.text;
✓ NETWO	Snippet	tion ing	Start		2 3 4			
Pr V WIDGE	t User rofile	aster	End	# -				
							Cancel	Save

64. Finally, update your button to clear the image and go back to the search form. Select your button and click **onClick Edit**.

2									i 🕹	📥 HCL - herbert.hilhorst	💄 Herbert Hilhorst	
			DESIGN				DXProductDetails			PROPERTIES		
$\otimes$	Project Themes Templates Assets $Q\equiv$	[][] Responsi	ve Web 🛛	Responsive Web		ා 30 ී % Breakp	oint Forking 🔵	•		n Button Action Review		
	IblProductName								✓ General			
(ý)	IbiDesignerName								onClick	Click Edit to add actior	s Edit Ø	
.0.	abd IbIDescription	Product Name										
$\mathbf{\bullet}$	abi Iblimages	Designer Name							onTouch	Start Click Edit to add action	s Edit Ő	۴.
$\odot$	IblProductNameValue	Description	aceholder						onTouch	Move Click Edit to add action	s Edit 🖉	
~	IbiDesignerNameValue								onTouch	End Click Edit to add action	s Edit ර	
(?)	txtDescriptionValue											
	📑 segImages 🖂	Images										
36	🤊 imglmage											
	btnClose							1				
E	Default Library My Libraries					_						
	CheckBoxGroup				Close							
×	📰 DataGrid					-						

65. First clear the image URL. Drag a FUNCTION – Add Snippet and enter:

self.view.imgImage.src="";

Action	Editor [onClick]					×
Action ID	AS_Button_ee9146a5	1f924b55bb7165edb6143024			Diagram View	
Search	Q Condition			Validate Sequence	<pre>l self.view.imgImage.src=""; </pre>	
Y FUNCTION	•		Start			
V NETWOR	•		Add Snippet			
Get U Prot	file		End			

66. Then add the navigation. Drag a **GENERAL** – **Navigate to Form** node, set it to **DXContentSearch** and click **Save**.

Action Ec	ditor [onClick]			×
Action ID	AS_Button_ee9146a51f924b55bb7165edb6143024			Diagram View Design View Code View
Search	٩	Start	Validate Sequence	Pass data with navigation 🕕
Add Vari	Variabie	Add Snippet	, DXProductDetails	
String Operatio Exit Ap	P g Try-Catch	Navigate to Form DXContentSearch	+	
Olear Tim	eeout	<u> </u>	-	Cancel Save

67. Now test it. In the menu click **Build – Live Preview Settings**.

É	Volt MX Iris	Project	Edit	Build	Debug	Forge	Window
• •	•			Run Live Preview			жR
				Live Pre	eview Set	tings	
<b>/</b>				Custom	ize Volt N	ИХ Арр	
	mvDXSampl	es 💙					- N

68. Ensure the **Responsive Web** is the only channel selected and select the **Clean Preview** checkbox. Ensure the Foundry runtime matches the Foundry runtime environment, which you published Foundry application. Then click **Save & Run**.

Live Preview Settings	DVI	×
Platform and Channels		
	NATIVE	ADAPTIVE WEB 🕕
<b>é</b> 100		
n Android	$\Box$	
Android TABLET		
é ios		0
🛱 Android	$\Box$	0
Environment m1000 /9.5 Change Preview Mode Debug V () Clean Preview Get HCL Volt MX ()		
Project Settings Cancel	Save	Save & Run

69. Enter your Content Template ID **36a33863-6ea3-4d44-8430-aa5287682aa2** and Parent ID **0407455c-68ed-455b-84c6-f916cedff649** and click **Search Web Content**.

•••	Volt MX Iris Preview
$\langle \rangle$	3 🙊
	Responsive         Image:
	•
	ContentTemplateID
I	36a33863-6ea3-4d44-8430-aa5287682aa2
	0407455c-68ed-455b-84c6-f916cedff649
	Search Web Content

70. It shows the list of content items of that Content Template and ParentID (Site Area). Click one of the products to show, e.g. **Product 01**.

		Volt MX Iris F	Preview		
< > C *					
	Responsive	\$ 1280	<b>x</b> 1769	70 %	
•				Welcome, User name	[→ Logout
ContentTemplateID			Title	ld	
36a33863-6ea3-4d44-8430-aa5287682	2aa2		Product 05	6c5abb	2b-bdbd-461c-97ba-43c5994ed3f9
ParentID			Product 08	f4437e8	b-6bf2-4e5f-939f-9e841781c0d7
			Product 04	bbaaf30	lf-8895-4c8f-99ca-60cfafd934c3
0407455c-68ed-455b-84c6-f916cedff6	649		Product 01	4282f9f	d-799d-48fb-a49b-65c9598016cd
			Product 03	5e8961	1c-b55f-47e4-a82d-16e8985cffe4
			Product 02	0929ca	af-1ead-4fb4-b919-ae000e00e1e6
			Product 07	1fd886c	11-e655-48da-a354-471adc11bcc8
			Product 06	374a89	e4-ebde-4f5d-8c39-3f897c863705

71. It shows the details on that content item. Then select one of the images, e.g. **Product-Image-01** and it should show the image to the right.

• • •	VOIT MX IFIS P	review	
< > C	*		
	Responsive 🗘 1280	x 1769 70	%
Product Name	Product One		
Designer Name	Mariana Bowes		
Description	Lorem Ipsum is simply durany text of the printing and typesetting industry. Lorem Ipsum has been the Industry's standard durany text ever since the 1500s, scanshed it in our local states and the state of the state survived not only five centre, but also the large into electronic typesetting, remaining essentially unchanged. It was oppularied in the 1960s with the release of Letraset bestev containing Lorem Ipsum		
Images			
Product-Im	age-01	11/11/11	
Product-Ima	age-02		
Product-Ima	age-03		

72. Feel free to make the UI look better, add the other image renditions, add a loading icon when you load the image, etc.

You have successfully learned how to integrate the DX content into your HCL Volt MX application.

# Part 3: Use Volt MX Foundry to Integrate External Data Sources in DX

In this part, you will learn how use Digital Data Connector to integrate data sources from HCL Volt MX Foundry (integrated external data sources) into HCL Digital Experience.

To learn how more on the Digital Data Connector, use the DX for Developers (Beginners) course <u>https://hclsw.co/hdx-dev-100</u> with the Digital Data Connector lesson: <u>https://hclsoftwareu.hcltechsw.com/courses/lesson/?id=1451</u>.

You may use Volt MX Foundry for API management, as documented in <u>https://opensource.hcltechsw.com/volt-mx-</u> <u>docs/docs/documentation/Foundry/voltmx_foundry_user_guide/Content/API_Management.html</u>. Foundry helps to integrate easily with many different data sources using endpoint adapters. See details on <u>https://opensource.hcltechsw.com/volt-mx-</u> docs/docs/documentation/Foundry/voltmx_foundry_user_guide/Content/Services.html.

Then, the Help Center has the following document *Connecting to HCL Volt MX Foundry through Digital Data Connector (DDC) :* <u>https://opensource.hcltechsw.com/digital-</u> <u>experience/latest/extend_dx/ddc/integrating_voltmx_foundry/</u>. Follow these steps and you will get this result.

<	Published Site 👻 Edit Mode: CON			<u></u>	:	[☆] A English ▼	HCL	· (i)
	← Back to home	¢	wpsadmin   Actions   Log Out   ⊘	Search	Site	0		
	Site Manager							
	🖺 🖻 🛟 🛟	6	Web Content Viewer (Contextual)			E* D*		
	Volt MX DDC -		• Leanne Graham • Bret • Sincare@april.biz <ui></ui>					
	Volt MX DDC							

# Part 4: Integrate Volt MX Foundry Web Applications in DX

In this part, you learn how to integrate Foundry web applications into HCL Digital Experience.

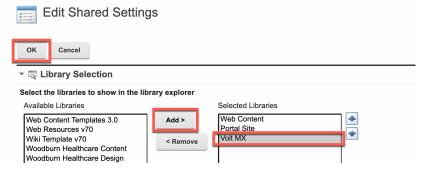
Today, an easy way to integrate an existing Foundry web application, is using an iFrame. This may change in the future. This works only for Foundry web applications that are suitable to be displayed in an iFrame.

This has been used for the integration you have seen in the DX integration with HCL Volt MX for Business Users.

1. Go to the Web Content Authoring and add the Volt MX web content library. Open the Applications Menu;, click **Web Content – Authoring** and **Edit Shared Settings**.

Published Site 🔻			⊡ 🔂	: 🎗 English 🕶 HCL (j
$\leftarrow$ Back Q : X			Home	
Web Content	- Library Explorer		Web Content	Search -
	Item Views	Libraries	Digital Assets	
Content Composer	Project Views	New -	Leap	Delete Display:
Q Authoring	▹ Group By Views	Previe	Site Templates	
Preview	Personal Views		Themes	
Web Content Libraries	(?) Help	Туре	Personalization	
	Preferences	▶ 🛃	Administration	
Syndicators	Configure	_	Help Center	First Previous Next Last
്റ്റ Subscribers	Edit Shared Settings Advanced Search	onow. re	Support	
Feed Configurations	Profile Users		Sabbour	

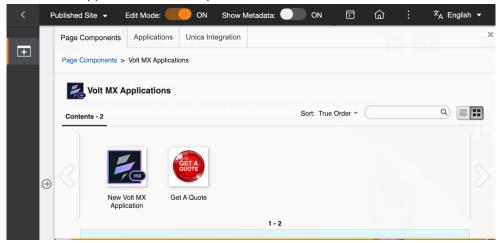
2. Then add the Volt MX web content library to the right and click OK.



You may want to set this to work with your own Volt MX server. In this case, edit the Volt MX (web content library) – Content – Global Config (site area) – Global Config content and update the VoltMX Server URL with your own server.

Gl Stat	Volt MX > Content > <b>obal Config</b> us - Published   Last eated by Harry Pappu	modified 5/17			Harry Pappus
Edit	Add To Project	Preview	Link	More -	Close
Name: Global Con Display titl	fig e:				
Global Con Descriptio None	•				
✓ VoltMX VoltMX Set	Server URL				
	mx.woodburn.digital:	8443/apps/			

4. Then have a look at the page components that are exposed in the toolbar for your business users, to allow them to easily add any Volt MX application into a DX page. There is a folder for Volt MX Applications, that may look like this.



5. You may look at how this **New Volt MX Application** that is preconfigured to show a default application (here GetAQuote) and optional form with a dedicated width and height. You may also update this to your needs. Feel free to reach out to our HCL DX services team to get production support on this implementation.

ibraries >	Volt MX > Content >	<ul> <li>Volt MX Appli</li> </ul>	cations > N	ew Volt MX Ap	plication	
N N	ew Volt MX A	pplicatio	on 😭			
St	atus - Published   Last	modified 6/22	/22, 9:58:46	AM GMT by	wpsadmin   C	reated by Harry Pappus
Edit	Add To Project	Preview	Link	More -	Close	
Conten	t Properties					
Name:	MX Application					
Display t						
Descripti						
This is an			nt that is not	yet configure	d. Add it to t	ne page and set the application name and optionally form to configure it with your
debiolied	voit wix application of	choice.				
✓ Volt N	IX Application					
	Application					
	name of the VoltMX a	pplication. E.g	. GetAQuot	9		
GetAQue	Sie					
- Volt N	IX Form					
Volt MX F	Form					
Optionally	, enter the name od th	ne specific Voli	MX form. E	.g. frm_Quot	91	
- Width	1					
Width						
	width of the compone	nt				
700						
- Heigh	ıt					
Height						
	height of the compone	ənt				
700						
- provi	ew-image					
previo	-mage					

6. And have a look at the use Content and Presentation Template for these content items. Feel free to copy and change them. You may involve HCL services to make them production ready.

You have successfully learned how to integrate HCL Volt MX Foundry web applications into DX.

# Part 5: Use Volt MX Iris to turn a DX Site into a Native Mobile Application

In this part, you learn how you may use Volt MX Iris to build a native mobile application from any DX site and deploy this to any store, like Apple AppStore and Google Play.

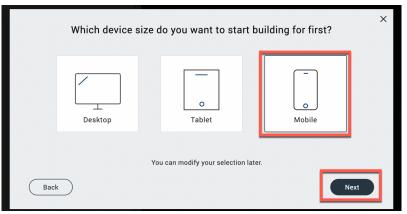
1. Create a new application in Iris. Create a new project. In the menu, open **Project** and click **New Project**.

É	Volt MX Iris	Project	Edit	Build	Debug	Forge	Window	Help
••	•	New Proje	ect			νN		Volt MX Iris
		Open				>		
<b>#</b>		Open File						
	0 1 - 14						DECION	

#### 2. Select Native App and click Next.

What do you want to start with now?							
Image: Web App     Image: Constraint of the second se							
Start with building either a web app or native app. Alternatively, you can also choose from one of our sa app projects to load and navigate around. Not sure what to start with? Help me decide	imple						
You can later add or modify your selection.							
	ext						

3. Then select the Mobile devices size, as it will directly create you the form for Mobile. Click **Next**.



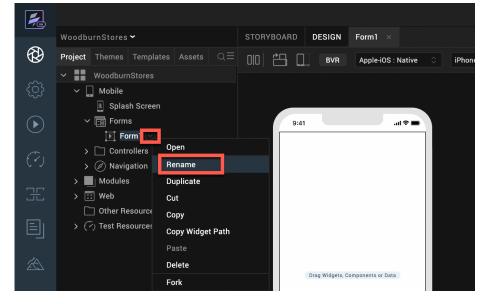
4. Name your Iris project, for example **WoodburnStores** and click **Create**.

You started with a <b>Native App</b> .						
		Project Name				
	WoodburnStores		)			
		Advanced Project Settings				
Back			Create			

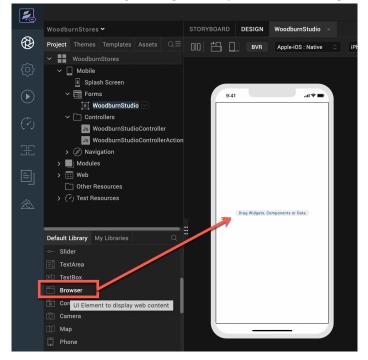
5. Then switch to the design mode. Click **DESIGN**.

	WoodburnStores 🛩	STORYBOARD DESIGN
	Default Library My Libraries	Mobile 0
	✓ Buttons	
ŝ	Button Button 1 Button 2	
	BorderButtonRound BorderButtonRound	Form1
$\bigcirc$		
Æ	BorderButtonRound BorderButtonRoundT	
(~)	Https://www.abcalifg.com	
	HyperLinkButton PaddedButtonRound	
£	Buton 2 Buton 2 Buton 2	<b>∺</b>
	PaddedButtonBound PaddedButtonBound	

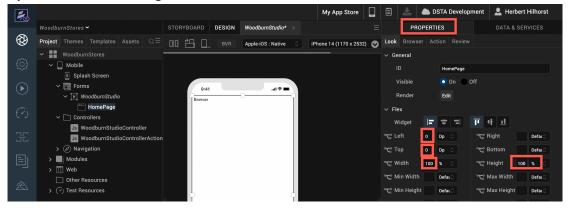
6. Then rename the form that was created, e.g. to the name of your site, like Woodburn Stores.



7. Add a browser widget. Drag and drop the Browser widget to your form.



8. Ensure it takes all space of the form. Click **PROPERTIES**, under Flex make **Left 0 Dp**, **Right 0 Dp**, Width 100% and **Height 100%**.

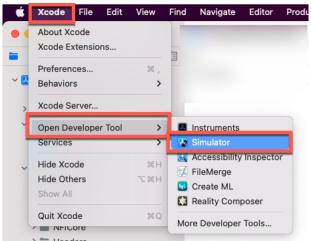


Set it to the URL of your site. Under PROPERTIES, switch to Browser and click Edit of Master Data. Then switch the Type to URL and enter the URL of your site, e.g. the one of Woodburn Stores (update the URL to your instance) :

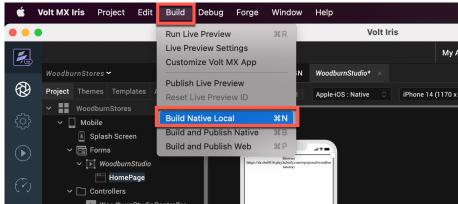
1		Master Data:	HomePage		×	📩 🌰 DSTA De	velopment 💄 Herbert Hilhorst
	WoodburnStores 🛩 STOF	Turne	URL		0	PROPERTIES	DATA & SERVICES
$\bigotimes$	<b>Project</b> Themes Templates Assets $Q \equiv$	Туре	-			ok <b>Browser</b> Action R	eview
	VoodburnStores	URL	https://dxplay	hclsofy.com/wps/portal/woodburnstores.	×	General	
- - - -	✓ 🛄 Mobile	Method	get			Master Data	Edit
_	Splash Screen					Detect Phone Number	• 0n • Off
$\mathbf{b}$	✓	Key		Value		Zoom	On • Off
-	HomePage					Native Communication	On Off
$(\mathcal{P})$	✓ ☐ Controllers						
	Js WoodburnStudioController					iPhone	
35	Js WoodburnStudioControllerActio					Base URL	
_	> Ø Navigation					Bounces	💿 On 🕘 Off
E	Default Library My Libraries					Browser Type	WK Web View
	Slider					Windows	
×A.	TextArea					Enable Cache	🔵 On 💿 Off
	🖃 TextBox					SPA	
	🛅 Browser					Set as content	On 💽 Off
	CordovaBrowser						
	🖸 Camera						
	Ш Мар						
	Consc			_			
	PickerView	Add	Delete	Cancel	ок		
U U U	Segment No b						

https://dx.sbx0000.play.hclsofy.com/wps/myportal/woodburnstores.

10. Then build your new native application and test it locally. You need to have set up a local emulator. This could be with Xcode <u>https://developer.apple.com/xcode/</u> if you have a Mac or Android Studio <u>https://developer.android.com/studio</u>. For example, on Xcode, start the Simulator. In the Xcode menu, under **Open Developer Tool**, click **Simulator**.

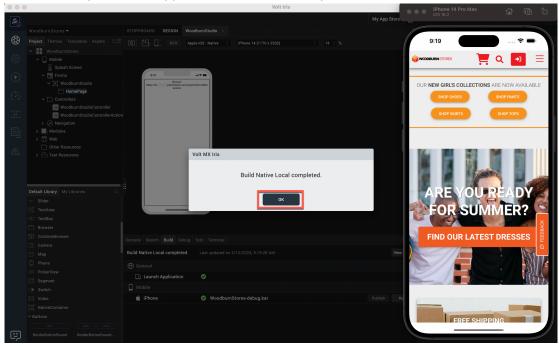


#### 11. Then run Build – Build Native Local.



12. Select the mobile emulator, e.g. iPhone 14 Pro Max (16.2). Select Clean Build and click Build.

Build Native Local			×						
Post Build Action	tun on my Device,	/Emulator 🗸							
Device" action installs th you to view your app on	The Post Build Action is initiated after the Build is complete. The "Run on my Device" action installs the application to your connected device and enables you to view your app on your device. As part of this action, backend services are also published to Foundry.								
Configured Devices & E	mulators	Clear History	Refresh						
🔲 Mobile									
📫 iPhone de Herb	ert (not connecte	d)							
G Mobile Emulators	0		_						
📫 iPhone 14 Pro N	Max (16.2)								
Tablet									
No devices conn	ected, connect a d	evice and click REFR	ESH						
Environment DSTA A	cademy v9.5 Ch	ange							
Build Mode Debug	· 0	Clean Build							
		Strict Mode	0						
Project Settings		Cancel	Build						



13. It then builds your native application and shows it in your mobile emulator. Click **OK**.

Congratulations. You have successfully turned your DX site into a native mobile application and may get it published on the different stores.

# Conclusion

Using this lab tutorial, you have learned how easy it is to use HCL Digital Experience assets like images, videos and files and structured content in your HCL Volt MX applications, and how to use Volt MX to easily integrate its data into DX, using Digital Data Connector and to integrate Foundry web applications into DX sites. And you learned how you may HCL Volt MX Iris to turn your existing DX sites into native mobile applications.

### Resources

Refer to the following resources to learn more:

HCL Digital Experience Home - <u>https://hclsw.co/dx</u>

HCL Volt MX Home - https://www.hcltechsw.com/volt-mx

HCL Digital Experience on HCL SoFy - <a href="https://hclsofy.com/">https://hclsofy.com/</a>

HCL Software - https://hclsw.co/software

HCL Product Support - <u>https://hclsw.co/product-support</u>

HCL DX Product Documentation - https://hclsw.co/dx-product-documentation

HCL Volt MX Product Documentation - <u>https://opensource.hcltechsw.com/volt-mx-</u> docs/docs/documentation/index.html

HCL DX Support Q&A Forum - https://hclsw.co/dx-support-forum

HCL DX Video Playlist on YouTube - https://hclsw.co/dx-video-playlist

HCL DX Product Ideas - <u>https://hclsw.co/dx-ideas</u>

HCL DX Product Demos - https://hclsw.co/dx-product-demo

HCL DX Did You Know? Videos - https://hclsw.co/dx-dyk-videos

HCL DX GitHub - https://hclsw.co/dx-github

## Legal statements

# This edition applies to version 9.5, release 216 of HCL Digital Experience and HCL Volt MX V9 and to all subsequent releases and modifications until otherwise indicated in new editions.

When you send information to HCL Technologies Ltd., you grant HCL Technologies Ltd. a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

#### ©2024 Copyright HCL Technologies Ltd and others. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with HCL Technologies Ltd.

## **Disclaimers**

# This report is subject to the HCL Terms of Use (<u>https://www.hcl.com/terms-of-use</u>) and the following disclaimers:

The information contained in this report is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied, including but not limited to the implied warranties of merchantability, non-infringement, and fitness for a particular purpose. In addition, this information is based on HCL's current product plans and strategy, which are subject to change by HCL without notice. HCL shall not be responsible for any direct, indirect, incidental, consequential, special or other damages arising out of the use of, or otherwise related to, this report or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from HCL or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of HCL software.

References in this report to HCL products, programs, or services do not imply that they will be available in all countries in which HCL operates. Product release dates and/or capabilities referenced in this presentation may change at any time at HCL's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. The underlying database used to support these reports is refreshed on a weekly basis. Discrepancies found between reports generated using this web tool and other HCL documentation sources may or may not be attributed to different publish and refresh cycles for this tool and other sources. Nothing contained in this report is intended to, nor shall have the effect of, stating.

or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results. You assume sole responsibility for any results you obtain or decisions you make as a result of this report. Notwithstanding the HCL Terms of Use <u>(https://www.hcl.com/terms-of-use)</u>, users of this site are permitted to copy and save the reports generated from this tool for such users own internal business purpose. No other use shall be permitted.