

HCL Volt MX Go VMXGo-DEV-100

Lesson 5 – UIのリブランディング



もくじ

はじめに.....	3
前提条件.....	4
コピー可能なコードソース.....	4
Lesson 6 – セグメントのソートとフィルタリング	5
frmLogin のリブランディング目標	5
まとめ	25
法的ステートメント	26
免責事項.....	27

はじめに

HCL Volt MX Go VMXGo-DEV-100 トレーニングコースは、現在のリリースである Volt MX Go v2.0.1 (<https://opensource.hcltechsw.com/voltmxgo-documentation/references/whatisnew.html>) 以降の HCL Volt MX Go ツールを学ぶための開発者向けスタートコースです。

Volt MX Go の開発者が知っておかなければならない 2 つの主要なツールは、1) Design Import と 2) VoltFormula です。Design Import は、Volt MX Go Iris (別名 Volt Iris) のプロジェクト、UI、Domino とデータを交換するロジックをゼロから作成する必要がなく、開発者の時間と労力を大幅に削減します。VoltFormula を使用すると、依存する Domino ロジックを HCL Volt MX プロジェクトで使用できるため、JavaScript でロジックを書き直す時間と労力を節約できます。

Design Import は、Domino バージョン 12.0.2 以降でホストされている HCL Domino アプリケーション (Domino REST API、別名 DRAPI に公開済み) を Volt MX Go の Volt Iris プロジェクトと Volt MX Go の Volt Foundry アプリ (Foundry ミドルウェアサービスのコレクション) にインポートします。Design Import の最終的な出力は、Volt Iris Web アプリで、すぐに機能し、完全に開発され、Domino アプリケーションを表すすべての Volt Iris フォームとウィジェットが含まれ、OAuth2 Identity サービス、Integration サービス、Volt MX Go の Foundry Domino アダプターを使用する Object サービスを持つ Foundry アプリに関連付けられています。

Design Import 後における Volt MX Go アプリ開発の一般的な流れは、組織のブランディングや UI 要件に合わせて UI のリブランド/リファクタリングを行い、Domino 文書のリストにソートやフィルタリング機能を追加することです。

この HCL Volt MX Go VMXGo-DEV-100 トレーニングには、上記を扱う 6 つのレッスンが含まれています。レッスンは以下の通りです。

1. Lesson 1 - Domino REST API 必須情報
2. Lesson 2 - Design Import のセットアップ
3. Lesson 3 - Design のインポート
4. Lesson 4 - VoltFormula
5. Lesson 5 - UI のリブランディング
6. Lesson 6 - セグメントのソートとフィルタリング

このコースでは、HCL Volt MX Go First Touch Recipe Catalog アプリとその資産 (Domino DB (レシピ保存用)、First Touch Recipe Domino REST API スキーマ、スコープ、DRAPI アプリ

UIのリブランディング

(<https://opensource.hcltechsw.com/voltmxgo-documentation/tutorials/firsttouch.html>) を含む) を活用します。DRAPI First Touch Recipe アプリで Design Import を実行し、Volt Iris アプリに VoltFormula を追加し、Iris アプリのログイン画面とダッシュボード画面/フォームをリブランドし、Iris アプリにソートとフィルタリング機能を追加します。

前提条件

このコースを修了するには、HCL Volt MX Go の Volt Foundry (ミドルウェア) と Volt Iris (IDE) に加え、Domino REST API を含む Domino 環境が必要です。Domino と Volt MX Go サーバーのオンプレミスインストールの代わりに、HCL SoFy プラットフォーム (<https://hclsofy.com>) の HCL Volt MX Go サンドボックスを使用できます。HCL SoFy サンドボックスには、Domino、Domino REST API、および Volt MX Go Foundry が含まれます。SoFy が提供するトライアルサンドボックスを使用するには、付録 I を参照してください。

オンプレミス

- HCL Domino server 12.0.2+
- HCL Domino REST API (DRAPI) サービス (タスクとサービスが稼動)
- HCL Domino REST API Console URL
- HCL Domino REST API Admin User Credentials (ユーザーID とパスワード)
- HCL Volt MX Go Foundry v2.0.1
- HCL Volt MX Go Foundry Console URL
- HCL Volt MX Go Foundry Admin User Credentials (ユーザーID とパスワード)
- HCL Volt MX Go Iris v2.0.1

HCL SoFy プラットフォーム (<https://hclsofy.com>)

- HCL Volt MX Go サンドボックス

コピー可能なコードソース

注意：設定やコードスニペットを Volt Foundry や Iris にコピーするよう求められたら、

Lesson_6-Segment_Sorting&Filtering-Copiable_Source.txt ファイルからコピーしてください。このドキュメントからコピーすると、不要な制御文字や隠された制御文字が含まれている可能性があり、Foundry サービスや Iris アプリが正しく動作しなくなります。

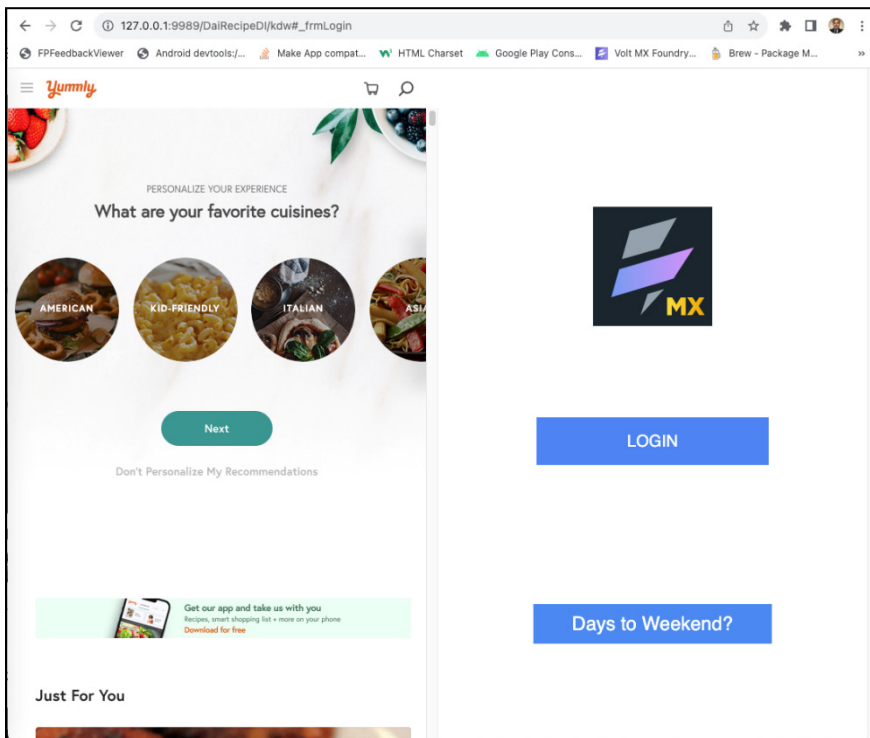
Lesson 5 – UIのリブランディング

このレッスンでは **Design Import** で作成したばかりの Volt Iris First Touch Recipes プロジェクトをリブランディングします。ゴールは、ログイン画面とダッシュボード画面（Volt Iris フォーム）のリブランディングです。

frmLogin のリブランディング目標

まず、ログインフォーム frmLogin のリブランディングを行います。

- ログインロゴの画像を変更します。
- メインフォームを2つに分割し、左半分はブラウザウィジェットを使ってHTMLの **iframe** タグ内に Web サイト <https://www.yummly.com/> を表示し、右半分はログインプロンプトにします。
- フォームが以下の3つのスクリーンブレイクポイントに反応するように調整します。640 ピクセル（モバイル）、1024 ピクセル（タブレット）、1366 ピクセル（デスクトップ）。



UIのリブランディング

ステップ

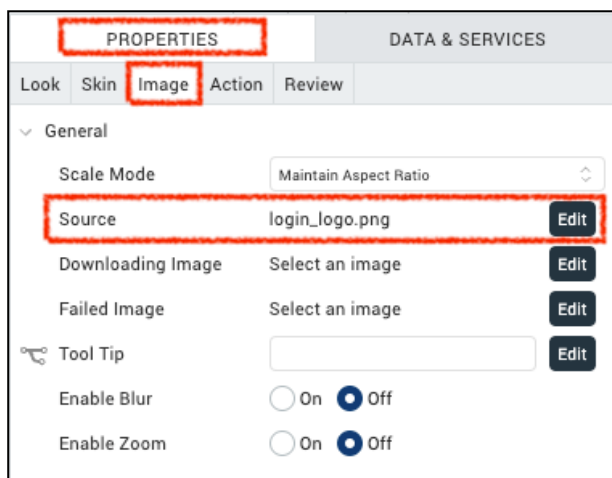
1. Volt Go Iris IDE から、IDE が前のレッスンで **Design Import** によって作成されたレシピプロジェクトですでに開かれていることを確認します。
2. IDE の左上で **Project** タブの下のフォームツリー構造をナビゲートして、フォーム **frmLogin** を開きます。フォームは Iris のセンターキャンバスに表示されているはずです。
3. Iris のセンターキャンバスで、青いロゴを含むトップ画像 **imgLogin** をクリックしてフォーカスを合わせます。



4. ロゴのソースを HCL Volt MX のリモート URL ログ画像に設定してロゴを変更します。ターゲットロゴの URL は次のとおりです。

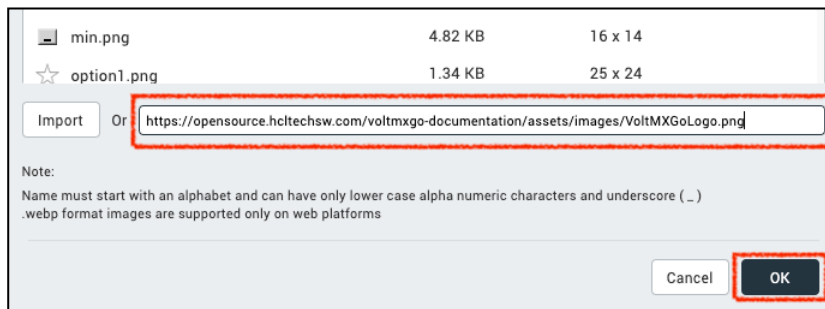
<https://opensource.hcltechsw.com/voltmxgo-documentation/assets/images/VoltMXGoLogo.png>

- **PROPERTIES** パネル → **Image** → **Source** → **Edit** で画像ソースに移動します。

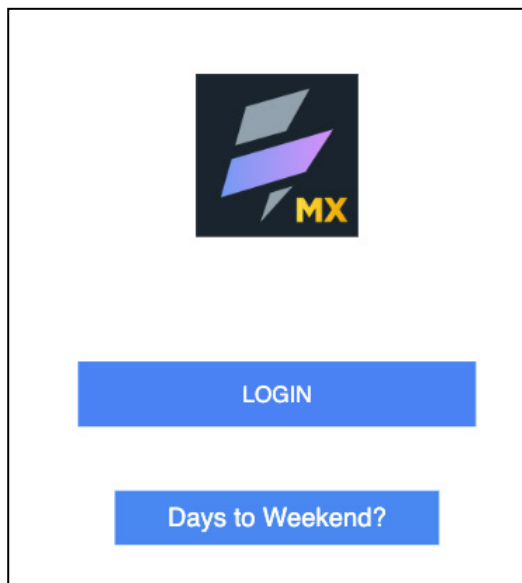


UIのリブランディング

- ポップアップウィンドウ "Source" が表示されます。上記の画像 URL をコピーし、"http image URL" と書かれたテキストボックスに貼り付け OK をクリックします。



ロゴ画像は HCL Volt MX の「稲妻」に変更されます。

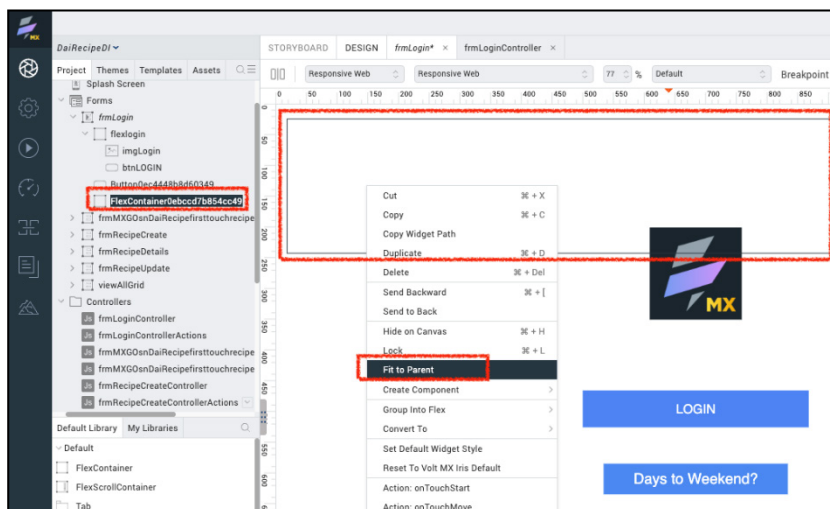


ここでフォーム frmLogin を 2 つの半分に分割します。左半分はブラウザ ウィジェットを使用して HTML の `iframe` タグで Web サイト <https://www.yummly.com/> を表示し、右半分はログイン画面になります。

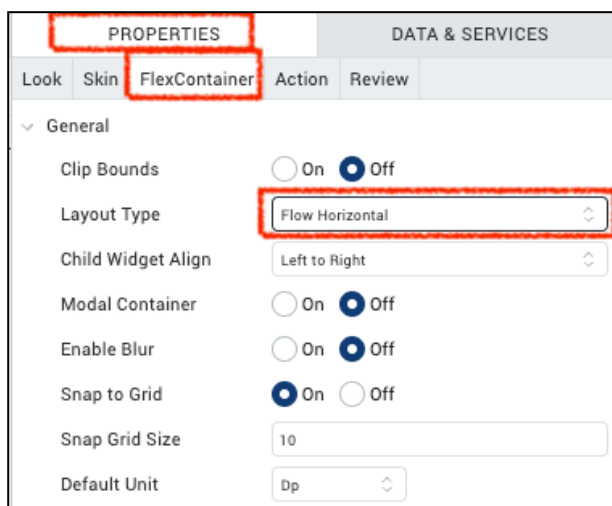
5. frmLogin をクリックしてフォーカスを設定します。
6. frmLogin に FlexContainer ウィジェットを追加します。
7. Default Library から FlexContainer ウィジェットを中央キャンバスの上部にドラッグ & ドロップします。

UIのリブランディング

- 新しい **FlexContainer** を右クリックし、**Fit to Parent** を選択します（ウィジェットは、上と左の位置が 0 に設定された親ディメンションをすべて占めます）。



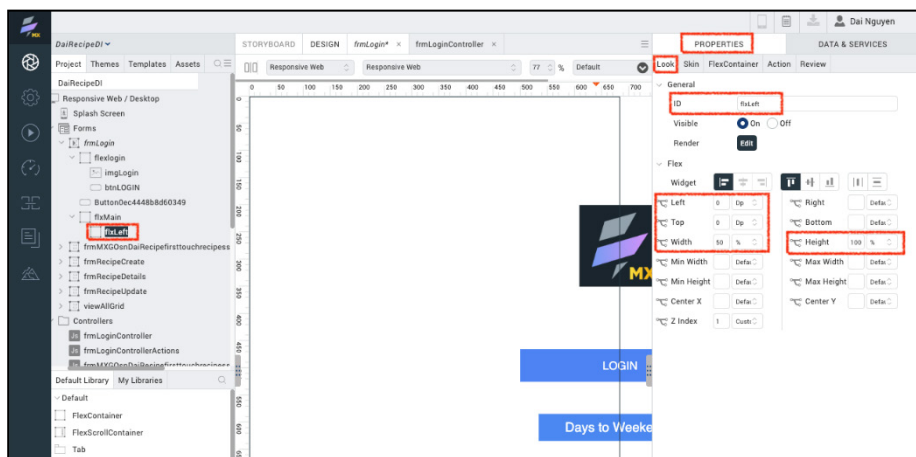
- Project タブの **frmLogin** のツリー構造で新しい **FlexContainer** を右クリックします。
- 新しい **FlexContainer** の名前を **flxMain** に変更します。
- PROPERTIES パネル -> **FlexContainer** -> **Layout Type** に移動し、**flxMain** の **Layout Type** を **Flow Horizontal** に変更します。



9. **flxMain** に新しい **FlexContainer** を挿入し、**flxLeft** という名称のブラウザウィジェットを入れます。
 - Iris のフォーカスが **flxMain** ウィジェットにあることを確認します。

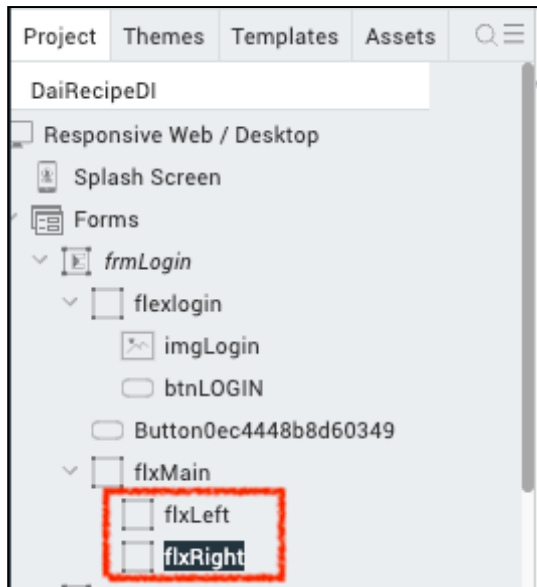
UIのリブランディング

- 新しい FlexContainer ウィジェットを **flxMain** にドラッグ&ドロップします。
- PROPERTIES の Look プロパティを使って設定します。
 - i. ID: flxLeft,
 - ii. Left: 0 Dp,
 - iii. Top: 0 Dp,
 - iv. Width: 50%,
 - v. Height: 100%



10. 次に **flxLeft** を複製し、複製した FlexContainer の名前を **flxRight** に変更します。

- Iris のフォーカスが **flxLeft** にあることを確認します。
- Iris の左側、Project タブの下、**frmLogin** のツリー構造の下にある **flxLeft** を右クリックします。
- **Duplicate** を選択し、複製した FlexContainer の名前を **flxRight** に変更します。
- Iris プロジェクトを保存します。



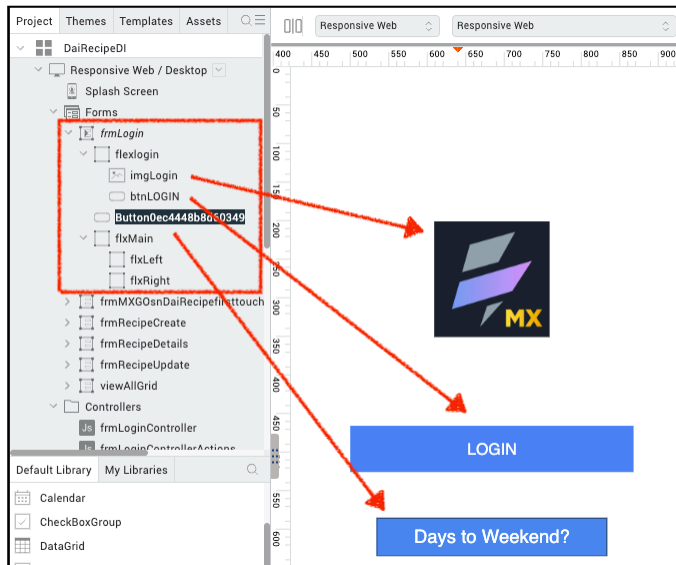
11. ログインボタン、ロゴ、そしてその親 FlexContainer の **flexlogin** を **flxRight** の下に移動します。

- Iris のフォーカスがコンテナ **flxRight** にあることを確認します。
- PROPERTIES -> FlexContainer -> Layout Type に移動します。
- **flxRight** の Layout Type を Flow Vertical に設定します。
- Iris のフォーカスを **flexlogin** に設定します。
- 右クリックし **flexlogin** をカットします。
- Iris のフォーカスを **flxRight** に移します。
- 右クリックで **flexlogin** を **flxRight** に貼り付けます。
- Iris のフォーカスを Days to Weekend?のラベルが付いたボタンに移動させます。
- ボタンをカットします。
- Iris のフォーカスを **flxRight** に設定します。
- Days to Weekend?のボタンを **flxRight** に貼り付けます。
- Iris のフォーカスを "Days to Weekend?"のラベルのボタンに設定します。
- PROPERTIES -> Look -> Top を 0 Dp に変更します。

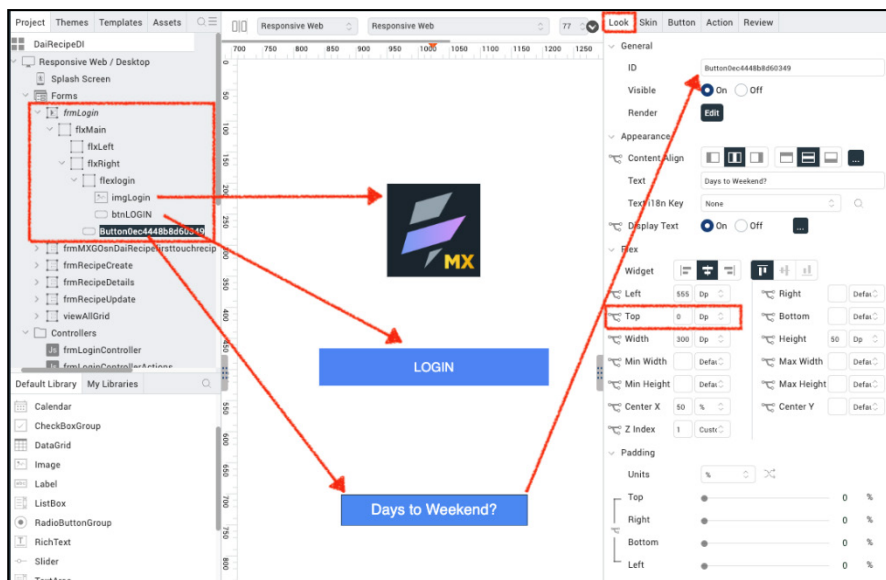
UIのリブランディング

- Iris のフォーカスを flexlogin に設定します。
- PROPERTIES -> FlexContainer -> Look -> Width を 30%から 60%に変更します。
- PROPERTIES -> FlexContainer -> Look -> Center Xを 50%に変更します。
- プロジェクトを保存します。
- 以下は、frmLogin のウィジェットツリー構造のビフォー・アフターです。

Before:



After:



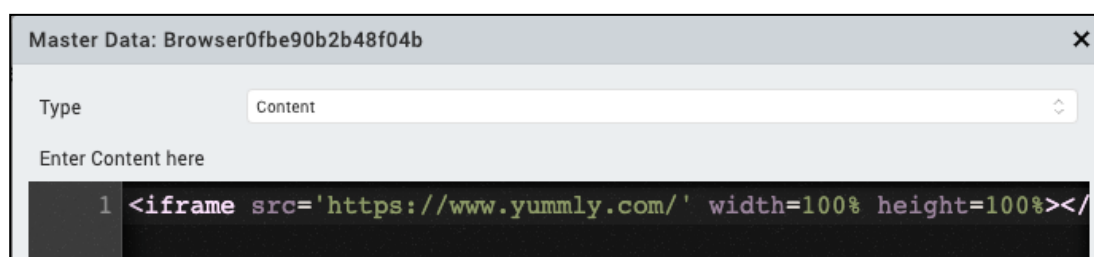
UIのリブランディング

これで frmLogin は一つの FlexContainer **flxMain** を含むようになります。そして **flxMain** は **flxLeft** と **flxRight** に分割され、**flxRight** にはロゴと2つのボタンが入ります。

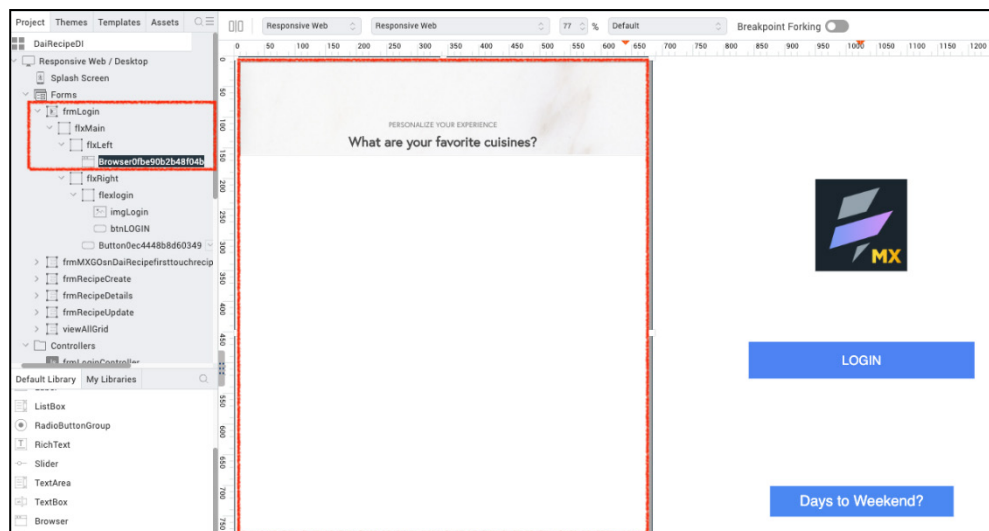
12. flxLeft にブラウザウィジェットを追加します。そして、iframe HTML タグで Web ページ <https://www.yummly.com/> を表示するようにブラウザを設定します。次のようにします。

- Iris のフォーカスを **flxLeft** に設定します。
- **Default Library** から **Browser** ウィジェットを中央のキャンバスにドラッグ&ドロップします。
- 新しい **Browser** ウィジェットを右クリックして **Fit to Parent** を選択します。
- **PROPERTIES** -> **Browser** -> **Master Data** -> **Edit** に移動します。
- Edit をクリックするとブラウザを設定するポップアップウィンドウが表示されます。
- **Browser** ポップアップで **Type** を **Content** に設定します。
- In the editable area below the label “**Enter Content Here**”, erase the word “**Browser**”,
- **Enter Content Here** というラベルの下にある編集可能なエリアで **Browser** という文字を消します。
- 下の HTML iframe タグをコピーし、編集可能領域に貼り付けます。

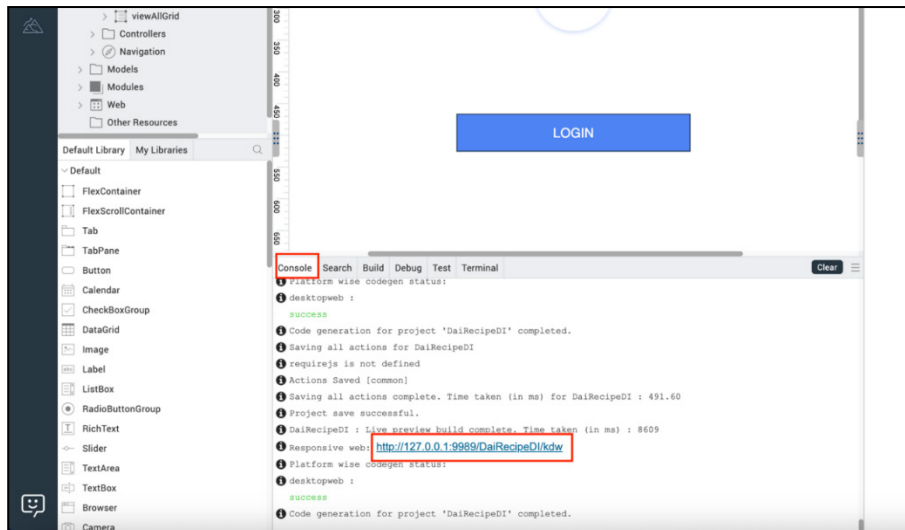
```
<iframe src='https://www.yummly.com/' width=100%
```



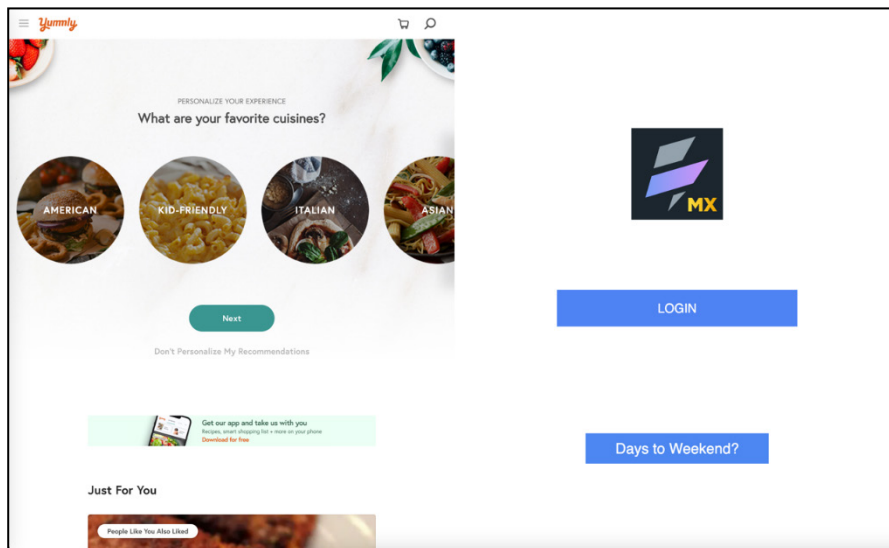
- OK をクリックしてポップアップウィンドウを閉じます。これで、中央のキャンバスのブラウザウィジェットに、ページのコンテンツ www.yummly.com が表示されるはずです。
- Iris プロジェクトを保存します。



13. これで、プロジェクトのプレビュービルドを実行する準備ができました。Iris のトップレベルメニューから **Build -> Live Preview** をクリックします。
14. ライブプレビュー設定で、**Responsive Web** チャンネルのみがチェックされていることを確認します。**Clean Preview** にチェックを入れます。ビルドモードはデバッグです。Foundry 環境が正しいことを確認し、必要であれば **Change** をクリックして設定します。最後に、**Save & Run** をクリックして Web アプリをビルドして実行します。
15. プレビュービルドが完了すると、Volt MX Go Iris Preview デバッグブラウザのポップアップが表示されます。そこでログインして Recipes アプリをテストできます。本来のブラウザからテストするので、デバッグブラウザは閉じてかまいません。
16. 真のブラウザからプレビューアプリをテストするには、次のようにします。
 - Volt Iris IDE の中央下部に、**Console**、**Search**、**Build**、**Debug**、**Test**、**Terminal** のタブがあります。**Console** をクリックします。
 - メッセージウィンドウを一番下までスクロールします。リンクが表示されているはずです。Response web: <http://127.0.0.1:9989/<your Iris project name>/kdw> というリンクが表示されます。



- リンクをクリックすると任意のブラウザで Web アプリが起動します。



ブラウザがモバイルサイズに縮小されたとき、実行中の Web アプリはどのようなでしょうか？

frmLogin をレスポンシブにする

では、frmLogin を 3 つのレスポンシブブレイクポイント、640 ピクセル（モバイル）、1024 ピクセル（タブレット）、1366 ピクセル（デスクトップ）に反応するように、Iris プロジェクトを変更します。それぞれのレスポンシブブレイクポイントに対して、次のようにします。

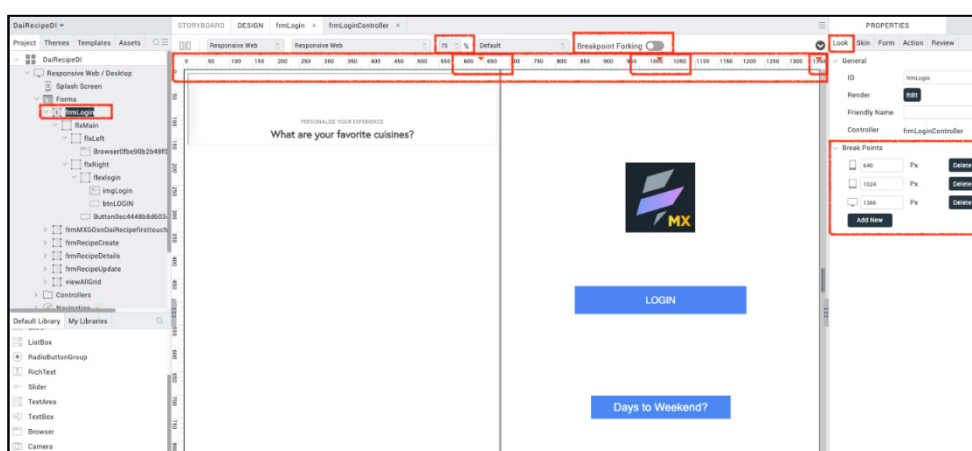
- モバイルでは、FlexContainer の **flxLeft** を非表示にします。ログインプロンプトの **flxRight** だけが表示されます。

UIのリブランディング

- タブレットの場合、**flxLeft** の幅を 50%から 65%に増やし、それに合わせて **flxRight** の幅も調整します。これにより左側の画面面積が広がります。
- デスクトップの場合は変更はありません。

frmLogin をレスポンシブにするには、フォームで定義されているブレイクポイント、ズームファクターが 75%以上のときに中央のキャンバスのルーラーに表示されるブレイクポイント、および **Breakpoint Forking** スイッチを使用します。

Breakpoint Forking を有効にすると、Iris ウィジェットの **PROPERTIES** -> **Look and Skin** の設定は、Web アプリがレスポンシブなディメンションで表示されるときにのみ適用されます。



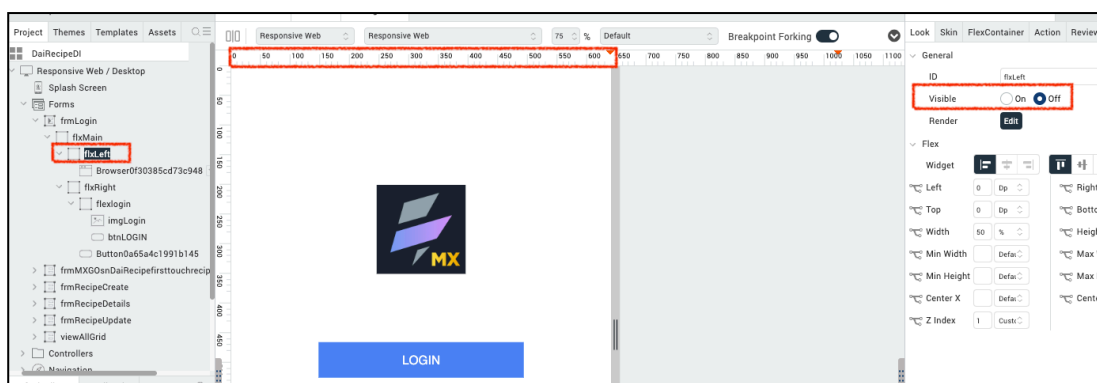
ステップ

17. Iris プロジェクトをモバイルのブレイクポイントに対応させるには、次のようにします。

- Iris の **zoom** ファクターが 75% 以上であることを確認し、中央のキャンバスルーラーのブレイクポイントマーカーが見えるようにします。
- **Breakpoint Forking** を有効にします。
- Click anywhere in the ruler between 0 and 640, the first Break Point, to display your app in a mobile factor. The **frmLogin** should be squeezed to fit this mobile view.
- 最初のブレイクポイントである 0 と 640 の間のルーラー内の任意の場所をクリックして、アプリをモバイル要素で表示します。frmLogin は、このモバイル表示に合わせて縮小されるはずですが。

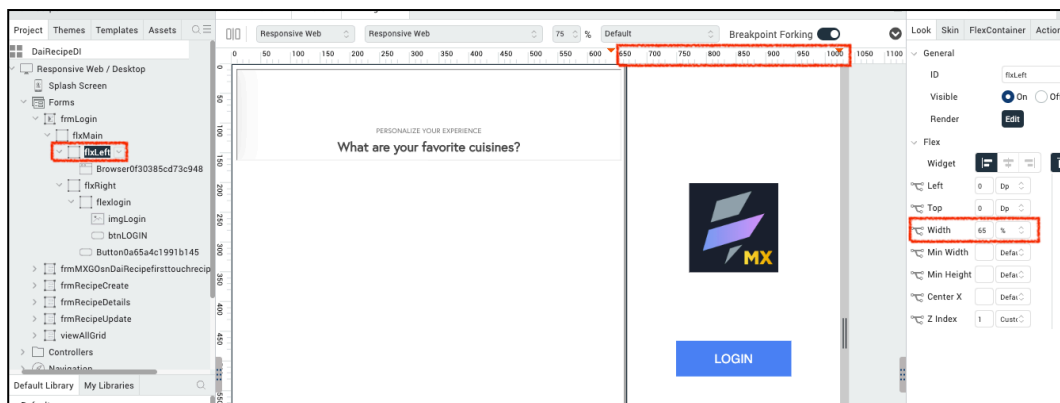
UIのリブランディング

- Iris のフォーカスを **flxLeft** に設定します。
- PROPERTIES → Look → Visible に移動します。
- **flxLeft Visible** をオフに設定します。 **flxRight** とそのウィジェットだけが見えるはずです。
- Iris のフォーカスを **flxRight** に設定します。
- **flxRight** の PROPERTIES → Look → Width を 50% から 100% に調整します。
- Iris プロジェクトを保存します。



18. Iris プロジェクトをタブレットのブレイクポイントに反応するようにするには、次のようにします。

- ルーラーで、2 番目のブレイクポイントである 640 と 1024 の間をどこかをクリックして、アプリをタブレット倍率で表示します。 **frmLogin** をこのタブレット表示に合わせる。
- Iris のフォーカスを **flxLeft** に設定します。
- Change the PROPERTIES → Look → Width from 50% to 65%.
- PROPERTIES → Look → Width を 50% から 65% に変更します。
- Set the Iris focus to **flxRight**.
- Iris のフォーカスを **flxRight** に設定します。
- PROPERTIES → Look → Width を 50% から 35% に変更します。
- Iris プロジェクトを保存します。



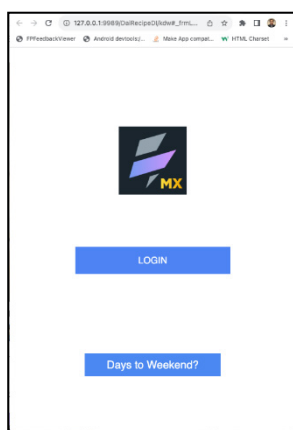
19. Breakpoint Forking を OFF に設定します。

注意：アプリを開発するときに Breakpoint Forking を OFF にすることを忘れないようにすることが重要です。そうすることで、グローバルな変更がひとつのブレークポイント内だけでなく、すべてのブレークポイントディメンジョンに適用されるようになります。

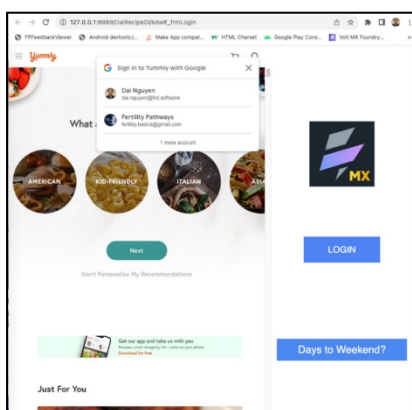
20. Iris プロジェクトを保存します。

21. これで、Recipe アプリをテストするためにプロジェクトのプレビュービルドを実行する準備ができました（上記のステップ 11 を参照）。

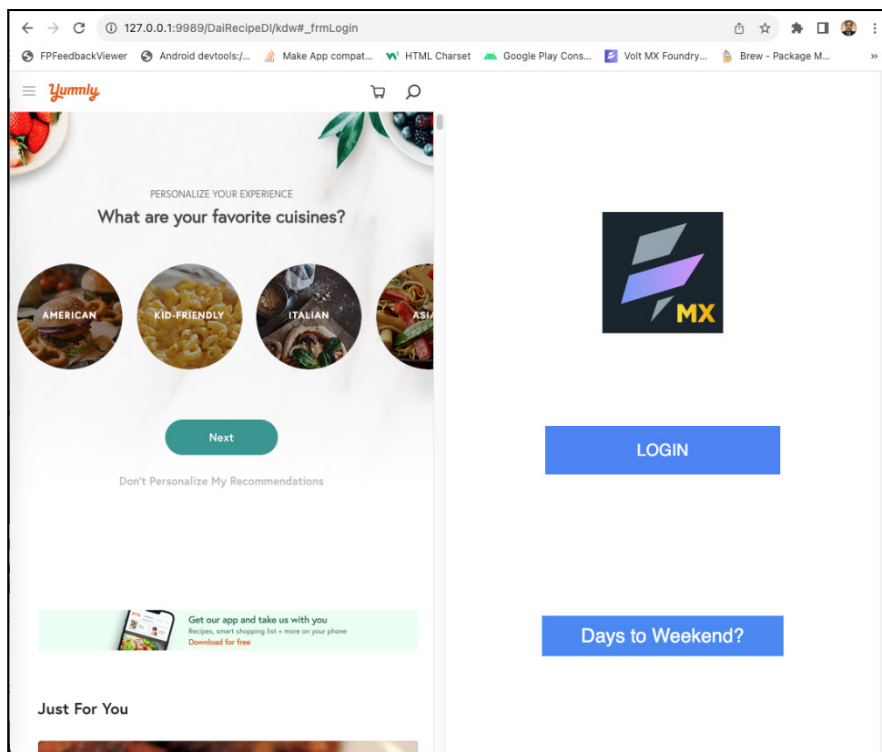
ブラウザの幅をデスクトップ（flxLeft = 50%、flxRight = 50%）、タブレット（flxLeft = 65%、flxRight = 35%）、モバイルサイズ（flxLeft visible off）に変更したとき、frmLogin 画面はそれに応じて調整されましたか？



Mobile



Tablet



Desktop

ダッシュボードフォームのリブランド目標

Dashboard フォームのブランド名を変更します (Design Import したアプリのダッシュボード フォームにはプロジェクトに対して一意の名前が付けられます)。

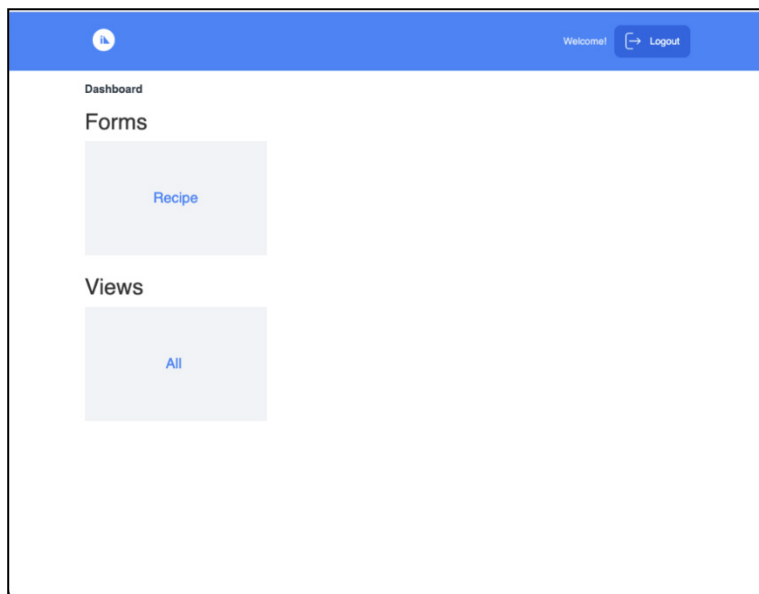
ダッシュボードフォームのリブランディングを行います。

- ヘッダーのロゴ画像を変更します。
- ダッシュボードフォームのボディに背景画像を追加します。

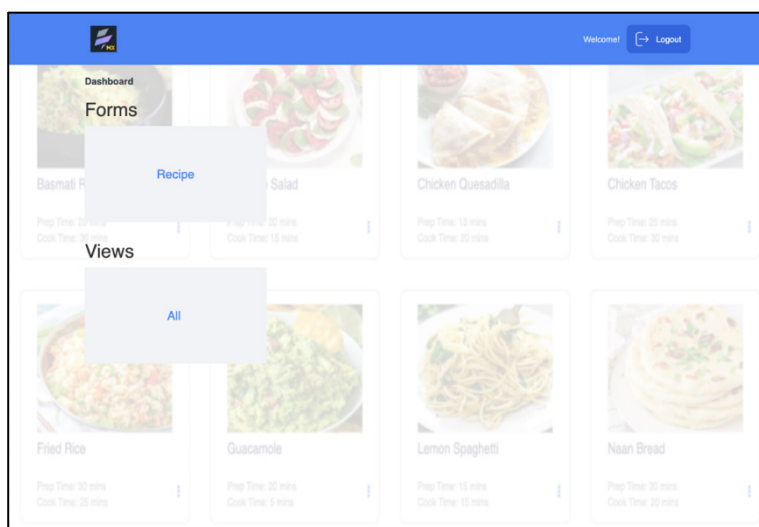
以下はダッシュボードフォームの例のビフォー・アフターです。

UIのリブランディング

変更前



変更後



ステップ

22. Volt Go Iris IDE から、Project -> Forms ツリー構造で Dashboard フォームをクリックし、中央のキャンバスで開きます。

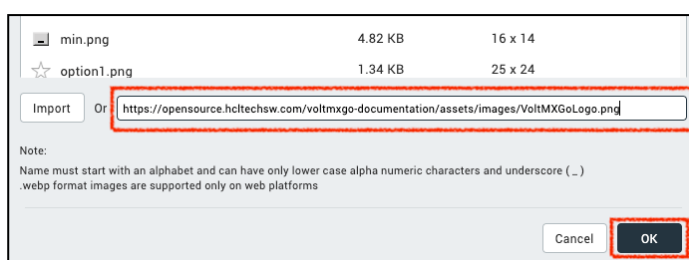
UIのリブランディング

23. ヘッダーロゴを、ステップ4の HCL Volt MX のリモート URL ロゴ画像に変更します。

- Iris フォーカスをロゴ画像に設定します（フォーム Dashboard -> flexHeader -> imgHamburgerMenu）。



- PROPERTIES panel -> Image -> Source -> Edit で画像ソースに移動します。
- Source のポップアップウィンドウが表示されますので、ステップ4でコピーした画像 URL を"http image URL"のテキストボックスに貼り付け、OK をクリックします。



ロゴ画像が HCL Volt MX の稲妻アイコンに変更されます。



24. ダッシュボードフォームに3つの FlexContainer があるはずですが、各 FlexContainer ウィジェットの PROPERTIES -> Look -> Z Index を調整して、これから追加する新しいウィジェットとコンテナの上にそれらが表示されるようにします。手順は以下の通りです。

- Iris のフォーカスを FlexContainer の flexHeader に設定します。
- PROPERTIES -> Look -> Z Index を 3 に設定します。
- Iris のフォーカスを flexBreadcrumb に設定します。
- Set its PROPERTIES -> Look -> Z Index to 3,
- PROPERTIES -> Look -> Z Index を 3 に設定します。
- Iris のフォーカスを flexFieldsContainer に設定します。
- PROPERTIES -> Look -> Z Index を 3 に設定します。

25. 次にフォームの Dashboard に背景画像を追加します。

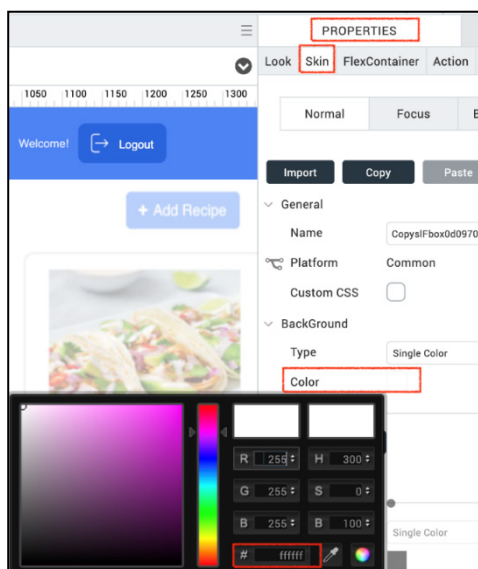
- Iris フォーカスをフォームの **Dashboard** に設定します。
- **Default Library** から **FlexContainer** ウィジェットを中央のキャンバスにドラッグします。
- Right-click on the new FlexContainer and rename it to **flxBGContainer**,
- 新しい FlexContainer を右クリックして flxBGContainer に名前を変更します。
- Right-click on **flxBGContainer** and select **Fit to Parent**,
- flxBGContainer を右クリックし、Fit to Parent を選択します。
- Drag the **Image** widget to the center canvas; the new Image will be the form's full background where **flxBGContainer** will be the Image's parent,
- イメージウィジェットをセンターキャンバスにドラッグします。
- 新しい画像を右クリックし、imgBG に名前を変更します。
- **imgBG** を右クリックし、**Fit to Parent** を選択します。
- **PROPERTIES** パネル -> **Image** -> **Source** -> **Edit** で画像ソースに移動します。
- ポップアップウィンドウの Source で、下のテキストボックスから画像の URL をコピーし、**http image URL** と書かれたテキストボックスに貼り付け、**OK** をクリックします。

<https://opensource.hcltechsw.com/voltmxgo->

- imgBG の **PROPERTIES** -> **Image** -> **Scale Mode** を **Fit to Dimensions** に設定します。
- Set imgBG **PROPERTIES** -> **Look** -> **Z Index** to 1; the Image will be at the bottom of the display hierarchy,
- imgBG **PROPERTIES** -> **Look** -> **Z Index** を 1 に設定します。
- Iris のフォーカスを **flxBGContainer** に設定します。

UIのリブランディング

- FlexContainer ウィジェットを中央のキャンバスにドラッグ&ドロップします。flxBGContainerがその親になります。
- 新しいFlexContainerを右クリックし、名前を flxFade に変更します。バックグラウンドの画像が薄くなります。
- flxFade を右クリックし、Fit to Parent を選択します。
- flxFade の PROPERTIES -> Look -> Z Index を 2 に設定します。flxFade が imgBG のトップ (on top) に配置されます。
- flxFade の PROPERTIES -> Skin -> BackGround Color に移動します。
- Color のボックスをクリックし、RGB カラー値を fffff (白) に設定します。

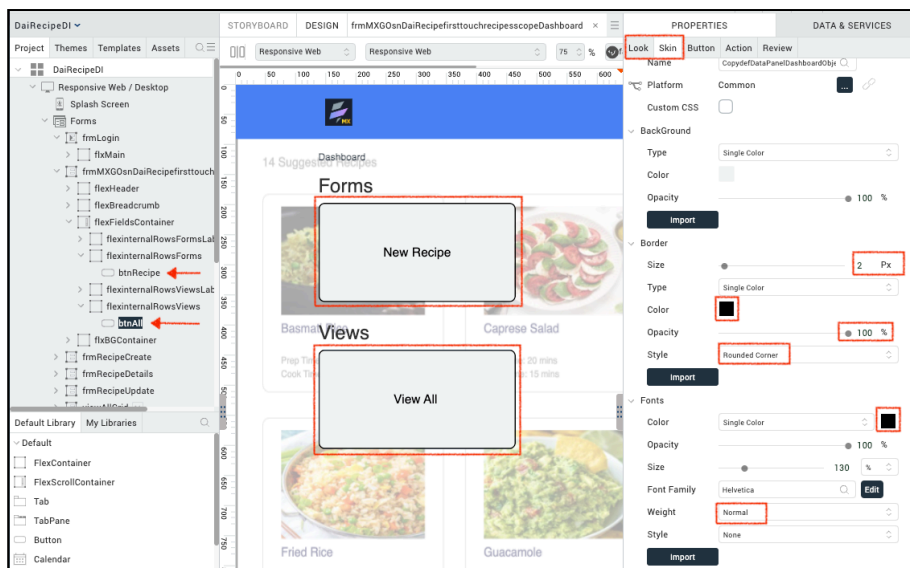


- flxFade の PROPERTIES -> Skin -> BackGround Opacity に移動します。
- Opacity の値を 60%に変更します。
- Iris プロジェクトを保存します。

26. 2つのボタン (Recipe と All) を、近くのラベルと同じように見えるようにします。黒い丸い縁取りを追加し、フォントの色を変更し、ボタンのテキストを更新します。

- 中央のキャンバスにある Recipe ボタンをクリックして Iris のフォーカスを設定します。
- PROPERTIES -> Look に移動します。

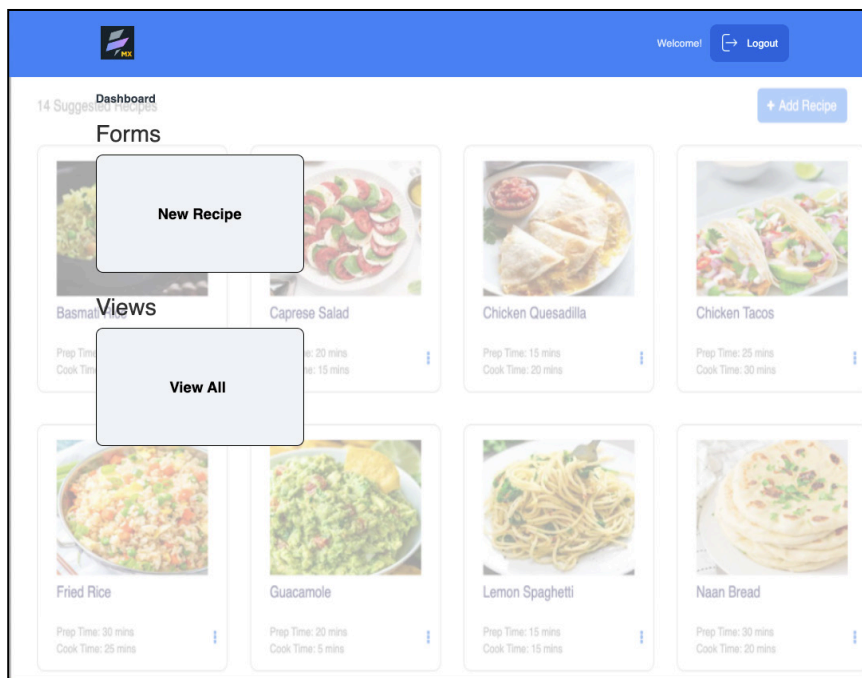
- ボタンの **Text** を Recipe から **New Recipe** に変更します。
- **PROPERTIES** -> **Skin** を開きます。
- 以下のボタンのスキンプロパティを変更または追加します。
 - i. Border -> **Size: 2 Px**,
 - ii. Border -> **Color: #000000**,
 - iii. Border -> **Opacity: 100%**,
 - iv. Border -> **Style: Rounded Corner**
 - v. Fonts -> **Color: #000000**
 - vi. Fonts -> **Weight: Bold**
- 中央のキャンバスにある All ボタンをクリックして、Iris のフォーカスを設定します。
- **PROPERTIES** -> **Look** に移動します。 .
- Change the button's **Text** from All to View All.
- ボタンのテキストを All から View All に変更します。
- **PROPERTIES** -> **Skin** に移動します。
- 以下のボタンスキンのプロパティを変更または追加します。
 - i. Border -> **Size: 2 Px**,
 - ii. Border -> **Color: #000000**,
 - iii. Border -> **Opacity: 100%**,
 - iv. Border -> **Style: Rounded Corner**,
 - v. Fonts -> **Color: #000000**.
 - vi. Fonts -> **Weight: Bold**.



27. Iris プロジェクトを保存します。

28. これで、プロジェクトのプレビュービルドを実行して Recipe アプリをテストする準備ができました（上記のステップ 11 を参照）。

ダッシュボードに新しいロゴ画像、レシピを表示する色あせた背景画像、黒の丸い枠線と黒のフォントの 2 つのボタンが表示されているはずです。



まとめ

このレッスンを完了しました。この時点であなたの Design Import アプリのログインとダッシュボードのフォームはリブランディングされました。frmLogin はレスポンシブで左側にレシピの Web ページを表示し、右側にログインを表示します。ダッシュボードには新しいロゴと薄くなった背景画像があります。

法的ステートメント

このエディションは、HCL Volt MX Go のリリース 2.0.1、および新しいエディションで別段の記載がない限り、それ以降のすべてのリリースおよび変更に適用されます。

あなたが HCL Technologies Ltd.に情報を送信する場合、あなたは HCL Technologies Ltd.に、あなたに対していかなる義務を負うことなく、適切と思われる方法で情報を使用または配布する非独占的な権利を付与します。

©2023 Copyright HCL Technologies Ltd and others. 無断複写・転載を禁じます。

米国政府ユーザーへの注意 - 制限された権利に関連する文書 - 使用、複製、または開示は、HCL Technologies Ltd.との GSA ADP スケジュール契約に規定された制限に従うものとします。

免責事項

本レポートは、HCL 利用規約 (<https://www.hcl.com/terms-of-use>) および以下の免責事項の対象となります。

本レポートに含まれる情報は、情報提供のみを目的としています。本レポートに含まれる情報は、情報提供のみを目的として提供されるものであり、本書に含まれる情報の完全性および正確性を確認するよう努めたが、商品性、非侵害性、特定目的への適合性の黙示保証を含むがこれに限定されない、明示または黙示を問わずいかなる保証もなく、現状のまま提供されるものである。また、本情報は、HCL 社の現在の製品計画および戦略に基づいており、HCL 社により予告なく変更される場合があります。HCL は、本レポートまたはその他の資料の使用またはその他の関連から生じる直接的、間接的、偶発的、結果的、特別またはその他の損害について責任を負わないものとします。本書に含まれるいかなる内容も、HCL 社またはその供給業者やライセンサーによる保証や表明を意図するものではなく、またそのような効果をもたらすものでもありません。

本レポートにおける HCL の製品、プログラム、サービスへの言及は、HCL が事業を展開するすべての国でそれらが利用可能になることを意味するものではありません。本プレゼンテーションで言及されている製品のリリース日や機能は、市場機会やその他の要因に基づき、HCL の独自の裁量で随時変更される可能性があり、将来の製品や機能の提供を約束するものではありません。これらのレポートをサポートするために使用される基礎データベースは、毎週更新されます。この Web ツールを使用して生成されたレポートと他の HCL ドキュメンテーションソースの間に見られる不一致は、このツールと他のソースの公開および更新サイクルが異なることに起因する場合も、そうでない場合もあります。本レポートに含まれるいかなる内容も、あなたが行った活動が特定の売上、収益の増加、節約、またはその他の結果をもたらすことを意図したものではなく、またそのような効果を持つものでもありません。利用者は、本レポートの結果として利用者が得た結果または利用者が行った決定について、単独で責任を負うものとします。HCL 利用規約 (<https://www.hcl.com/terms-of-use>)にかかわらず、本サイトの利用者は、本ツールから生成されたレポートを利用者自身の内部業務目的のためにコピーおよび保存することが許可されています。それ以外の使用は許可されません。