HCL Volt MX Go
VMXGo-DEV-100

Lesson 4 – VoltFormula

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 4 – VoltFormula.....5

まとめ.....20

法的ステートメント.....21

免責事項.....22

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

（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 (手順は Appendix I を参照)

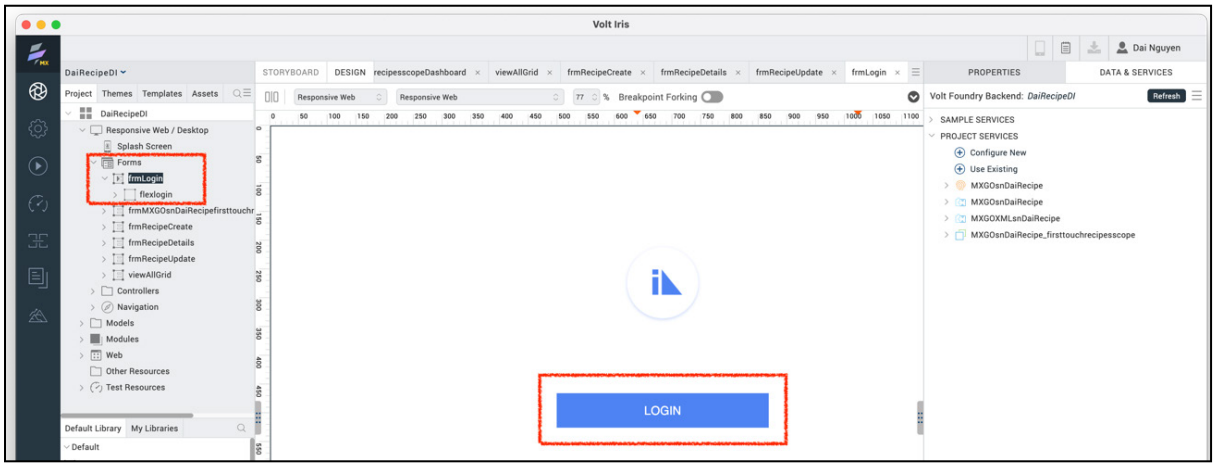
コピー可能なコードソース

> 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_4-Volt_Formula-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 4 – VoltFormula

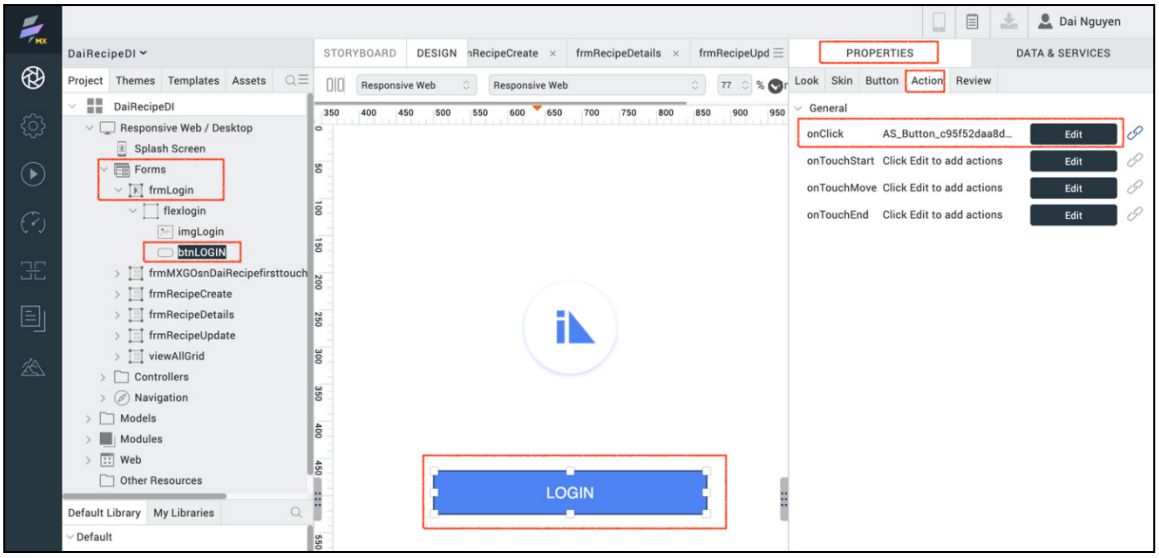
Volt MX Go Iris の VoltFormula では、OpenFormula と Notes Formula Language をソースに入力すると JavaScript コードに自動的に変換できます。Volt Iris で生成されたアプリはこのコードを後で実行できます。このレッスンでは、ローコードのアクションスクリプトエディタを使って VoltFormula を追加する方法と、それを Iris フォームコントローラに追加する方法を学びます。

Design Import で作成したばかりの Volt Iris First Touch Recipes プロジェクトを使用します。このパートの目標は、フォーム Login、frmLogin、LOGIN ボタンを修正して、VoltFormula を追加することです。



ステップ

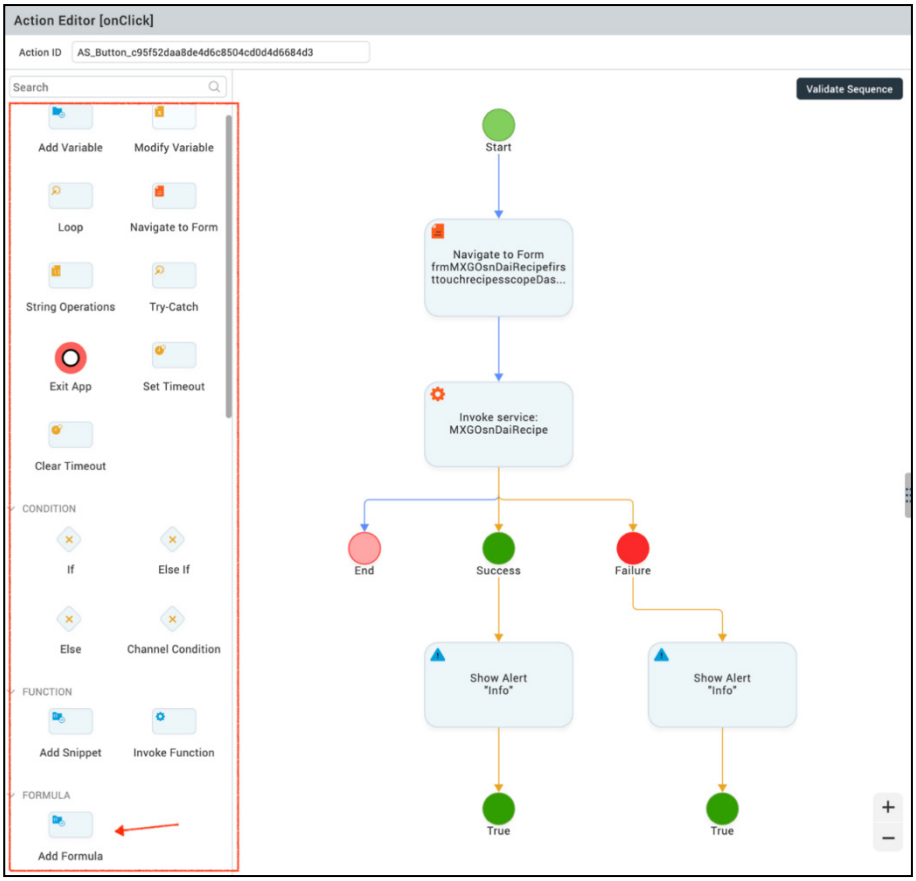
1. Volt Go Iris IDE から、**Lesson 3 - Design Import Part 2** で Design Import によって作成された Recipe プロジェクトがすでに IDE で開かれていることを確認します。
2. IDE の左上で、**Project** タブの下にあるフォームツリー構造を移動して、フォーム **frmLogin** を開きます。
3. Iris 中央のキャンバスで LOGIN ボタンをクリックしてフォーカスを合わせます。
4. Iris の右上で、**PROPERTIES** -> **Action** をクリックしてボタンウィジェットで利用可能なイベントを確認します。**onClick** イベントにはすでにアクションスクリプトがあり、その名前は AS_Button_で始まるもののはずです。
5. **Edit** ボタンをクリックして onClick イベントを修正します。



6. ローコードのアクションスクリプトエディタが開きます。

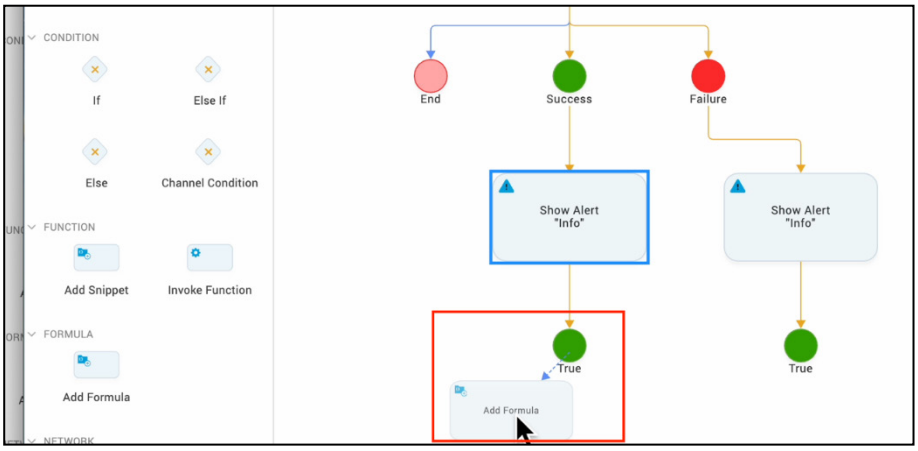
LOGIN ボタンのイベントハンドラのロジックスクリプトが表示されます。スクリプトは以下の通りです。1) ダッシュボードの Iris フォームに移動する、2) Foundry ログインサービスを呼び出す、3) ログインに失敗したら、エラーアラートを表示してスクリプトを終了する、4) ログインに成功したら、logged in アラートを表示してスクリプトを終了する。各ノードをクリックすると、その詳細が表示されます。

アクションスクリプトエディタの左側には、使用可能なローコードアクションノードが表示されたパレットがあり、ドラッグ＆ドロップしてロジックアクションスクリプトを構築できます。パレットウィンドウの一番下にある Add Formula ノードに注目してください。

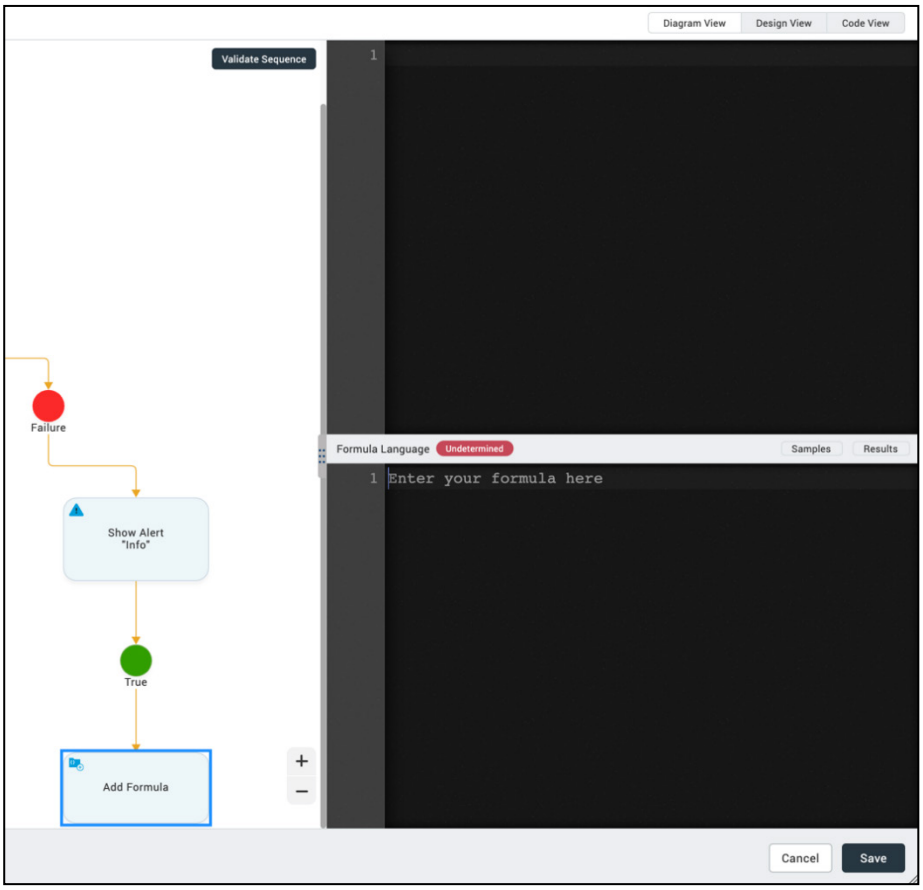


7. Success パスの Show Alert を変更して VoltFormula を追加します。この Formula は週末までの日数を計算し、ユーザーがログインに成功したときにウィンドウをポップアップします。

アクションパレットから、Add Formula ノードを Success -> Show Alert "Info" -> True ノードにドラッグします。True ノードと Add Formula ノードを結ぶ青い矢印線が表示されます。矢印の線が見えたら、ノードをドロップします。



8. ノードをドロップすると右側の VoltFormula エディタペインが表示されます。ペインは 2 つの部分に分かれており、下部には Domino Notes の Formula または OpenFormula を入力します。上の部分には JavaScript コードに変換された Formula が表示されます。JavaScript コードは、アクションスクリプトの一部としてノードの実行時に実行されます。

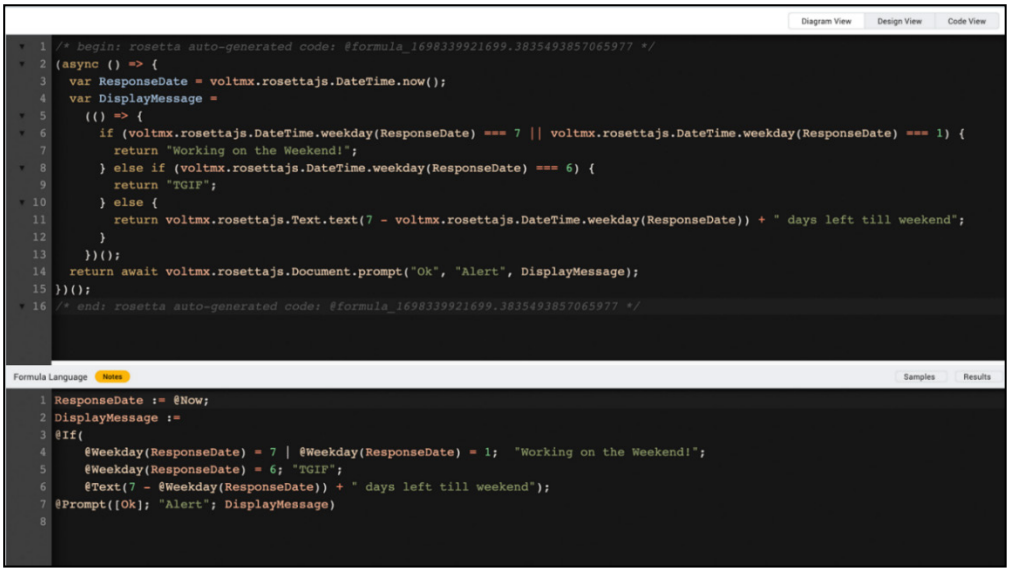


9. 以下のサンプルコードを使用して下のセクションに Formula として挿入します。

```
ResponseDate := @Now;
DisplayMessage :=
@If(
    @Weekday(ResponseDate) = 7 |
@Weekday(ResponseDate) = 1;  "Working on the
Weekend!";
    @Weekday(ResponseDate) = 6; "TGIF";
    @Text(7 - @Weekday(ResponseDate)) + " days left
till weekend");
@Prompt([Ok]; "Alert"; DisplayMessage)
```

Formula は週末までの日数を計算します。今日が平日であれば週末までの日数が表示され、そうでなければポップアップウィンドウで Working on the Weekend!と表示されます。

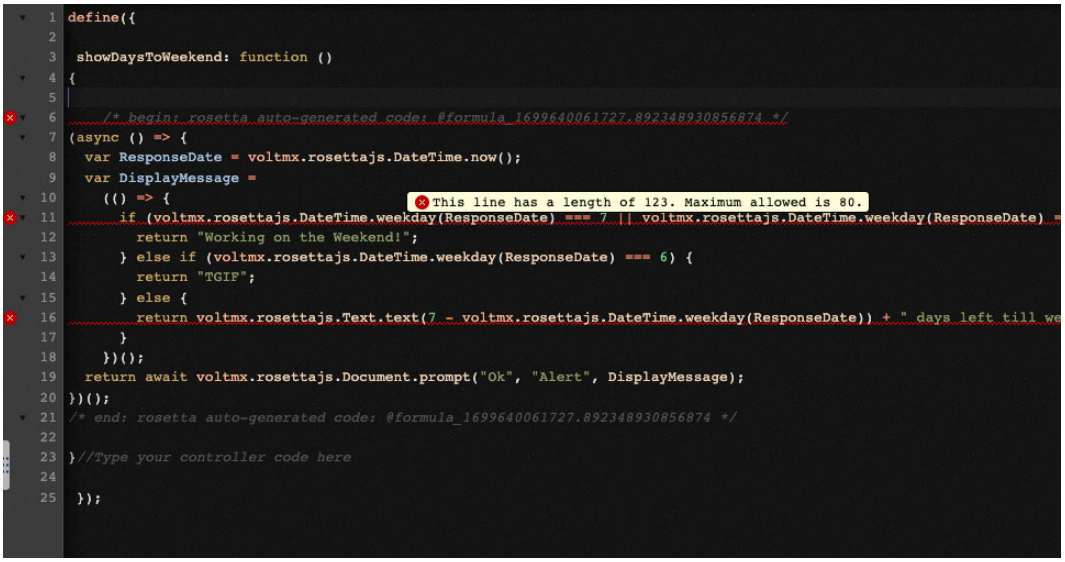
10. エディタペインには、Formula コードが下部に、変換された JavaScript コードが上部に表示されます。



11. Save をクリックして、更新したローコードのアクションスクリプトを保存します。

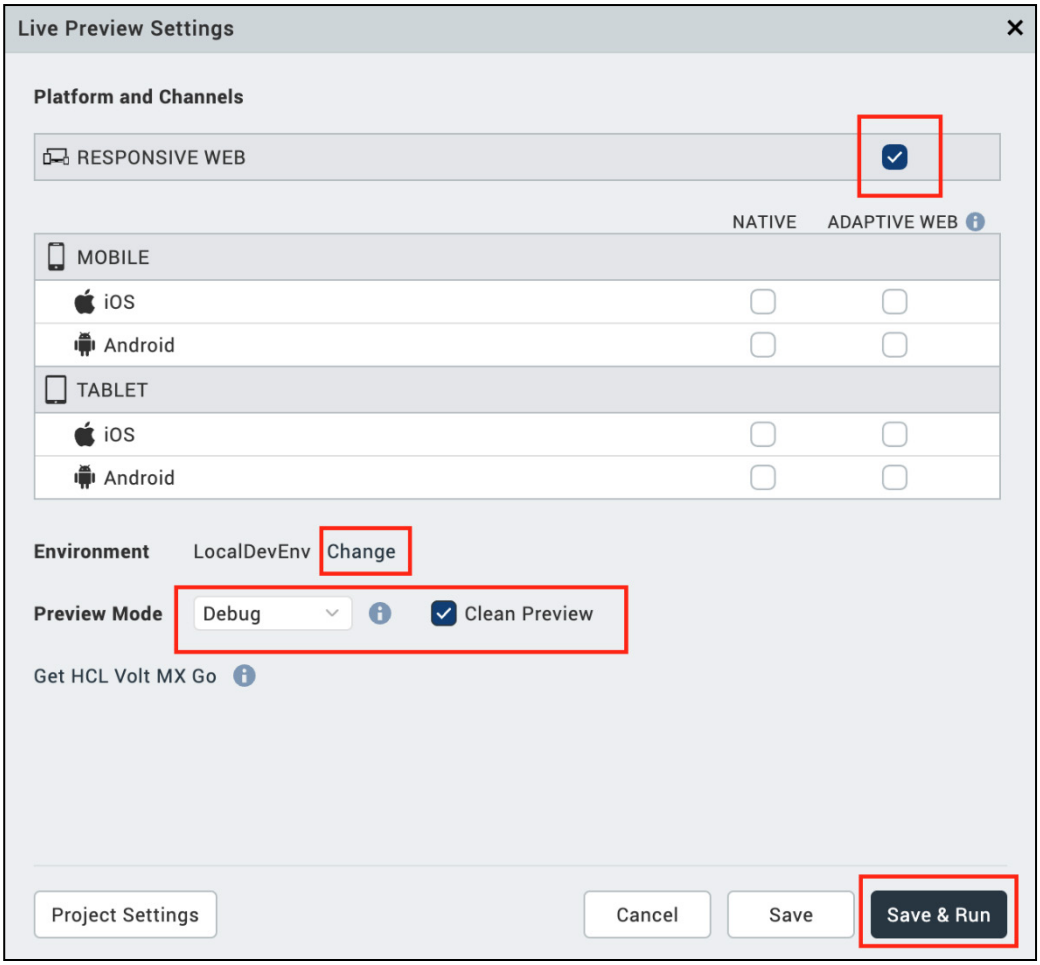
12. Iris のトップレベルメニューから Project -> Save、または Save All をクリックして Iris プロジェクトを保存します。

注意：HCL Volt Iris JavaScript コードエディタは JavaScript のコード行が長すぎるなどのエラーを表示することがあります。この種のリンティングエラーはコードの実行を妨げるエラーではありません。これらのエラーは簡潔で質の高いコードを開発するのに役立ちます。これらは Iris IDE に適用する制限やコードスタイルであり、設定が可能です。Iris -> トップレベルメニュー Help -> JS Linting Configurations を確認してください。下は Iris エディタのスナップショットの例で、リンティングエラーが表示されています。



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 設定をクリックします。

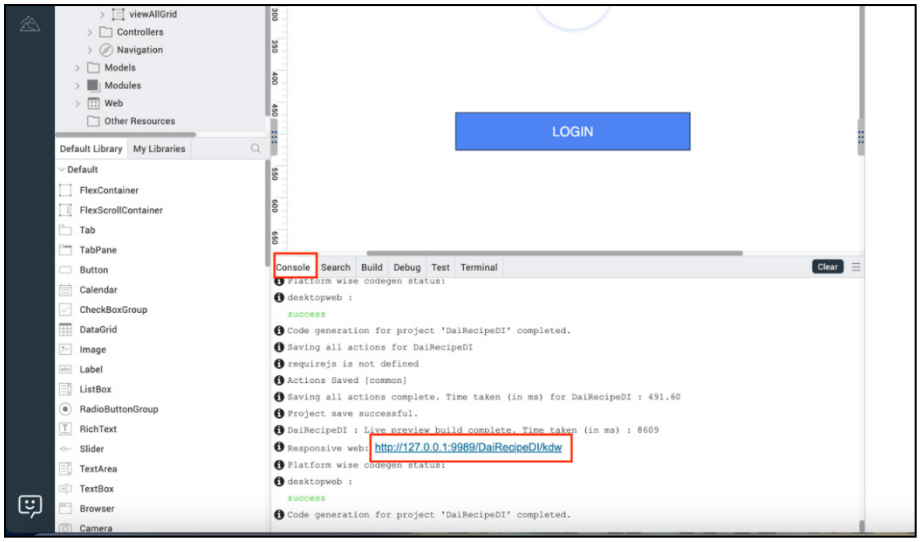
14. ライブプレビュー設定で Responsive Web チャンネルのみにチェックが入っていることを確認します。Clean Preview にチェックを入れます。ビルドモードは Debug です。Foundry 環境が正しいことを確認し、必要であれば Change をクリックして設定します。最後に Save & Run をクリックして、Web プレビューアプリをビルドして実行します。



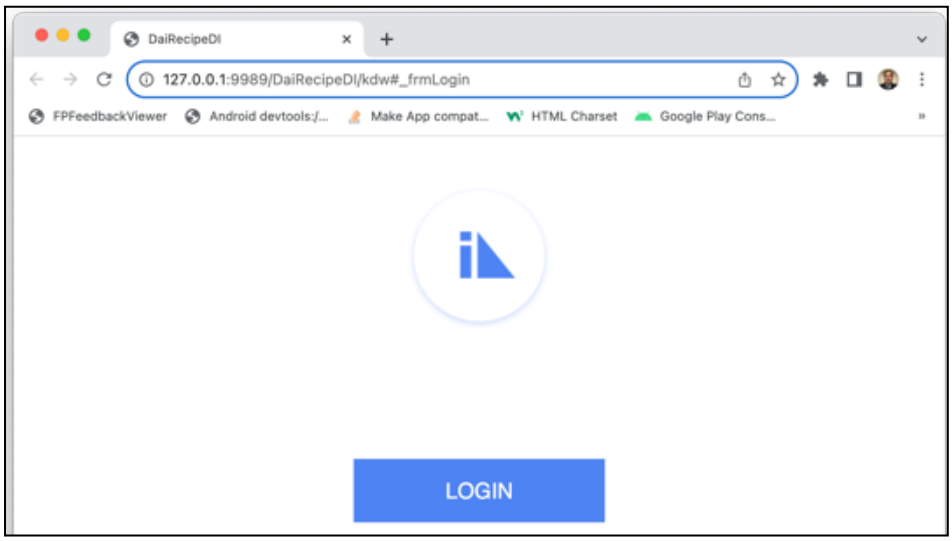
15. プレビュービルドが完了すると Volt MX Go Iris Preview デバッグブラウザのポップアップが表示されます。そこでログインして Recipes アプリをテストできます。デバッグブラウザは閉じてかまいません。

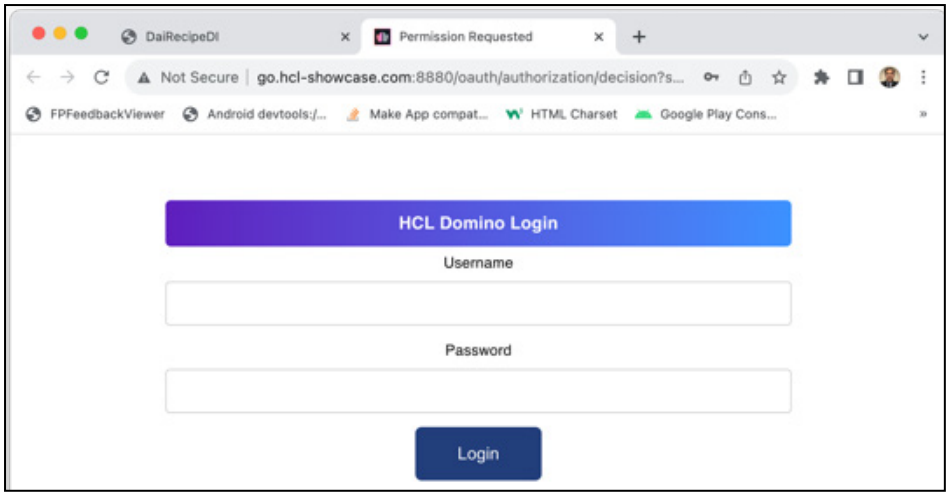
16. 本来のブラウザからプレビューアプリをテストするには次のようにします：

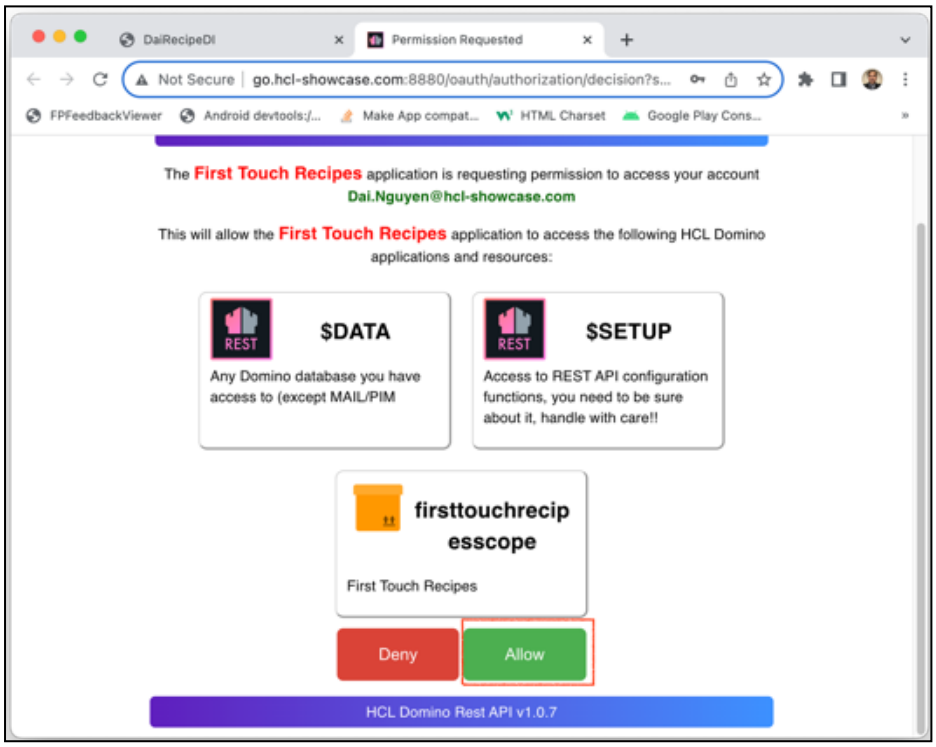
    - Volt Iris IDE の中央下部に Console、Search、Build、Debug、Test、Terminal のタブがあります。**Console** をクリックします。
    - メッセージウィンドウを一番下までスクロールします。リンクが表示されているはずです。Response web: http://127.0.0.1:9989/<your Iris project name>/kdw というリンクがあるはずです。

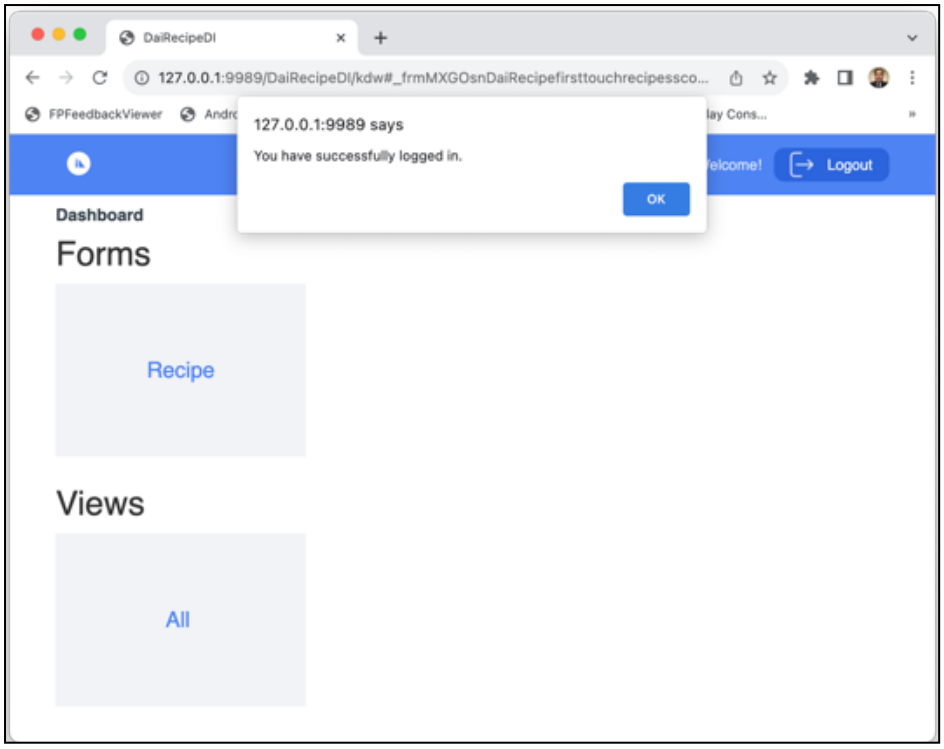


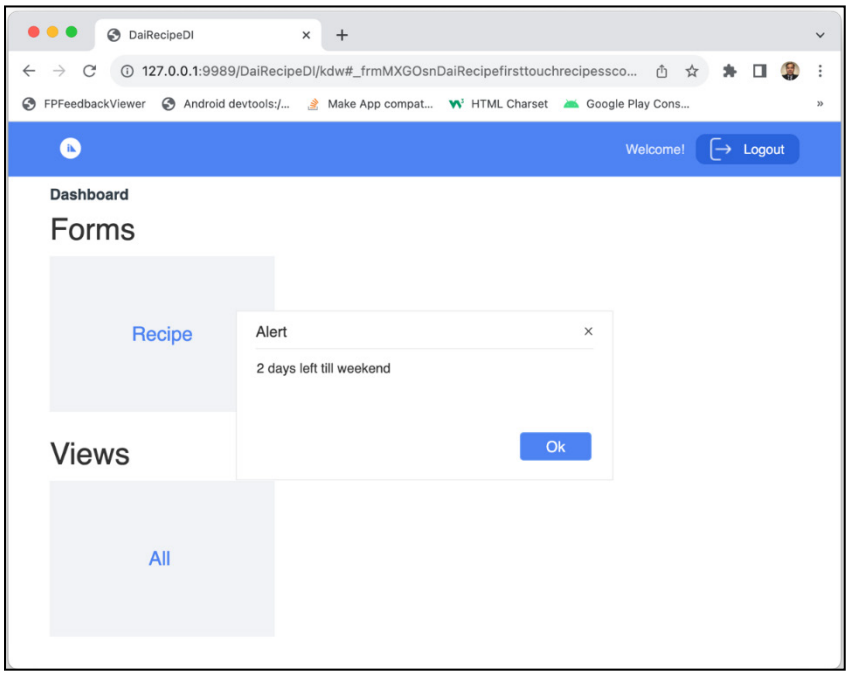
- リンクをクリックして任意のブラウザでウェブアプリを起動します。
- レシピアプリをテストするには、LOGIN をクリックします -> Domino クレデンシャルを入力して OAuth2 認証を通過し、Login をクリックします -> 緑の Allow ボタンをクリックして必要な DRAPI スコープへのアクセスを許可します。
- You have successfully logged in というポップアップアラートが表示されます。OK をクリックします。
- 週末までの日数が VoltFormula のアラートが表示されます。OK をクリックします。









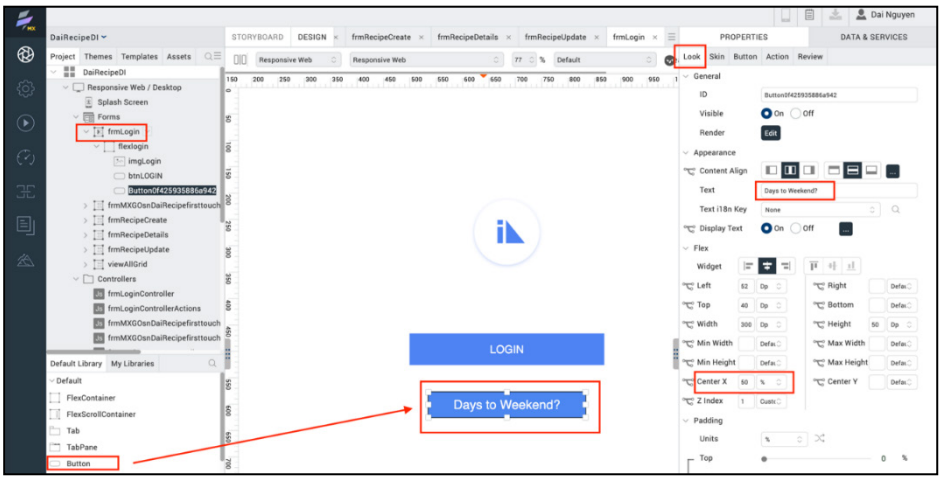


ローコードアクションスクリプトエディタを使って VoltFormula を追加するところまで、このレッスンは終了しました。

VoltFormula

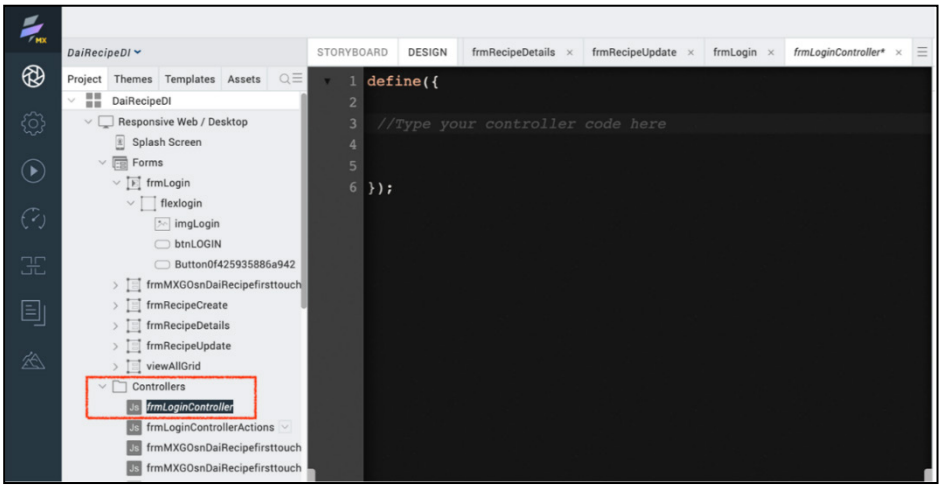
次に、Recipes プロジェクトを修正して Iris フォームコントローラに VoltFormula を追加します。フォームコントローラは Volt Iris のフォームに直接結びついた JavaScript コードモジュールです。MVC (Model-View-Controller) アーキテクチャのコントローラ部分です。Volt Iris で作成するすべてのフォームには、対応するコントローラが作成されます。ここでは、フォーム **frmLogin** を修正してボタンを追加し、その **onClick** イベントで **frmLoginController** モジュール内の関数を実行するようにします。

17. Volt Go Iris IDE から、Lesson 3 - **Design Import** 2 Design Import で作成された Recipe プロジェクトを開いてください。
18. 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 **frmLogin** フォームを開きます。フォーム frmLogin にフォーカスが当たっていることを確認してください。
19. frmLogin にボタンを追加します。ウィジェットの **Default Library** から Button ウィジェットを LOGIN ボタンのすぐ下にドラッグして新しいボタンをドロップします。



20. 新しいボタンが常に水平方向の中央に配置されるようにするには、**PROPERTIES** -> **LOOK** をクリックします。
21. **Center X** プロパティを 50%に設定します。
22. ボタンの **Text** を Days to Weekend?に変更します。
23. プロジェクトを保存します。
24. Iris IDE の左側でツリー構造 **Controllers** の下にある **frmLoginController** をクリックします。

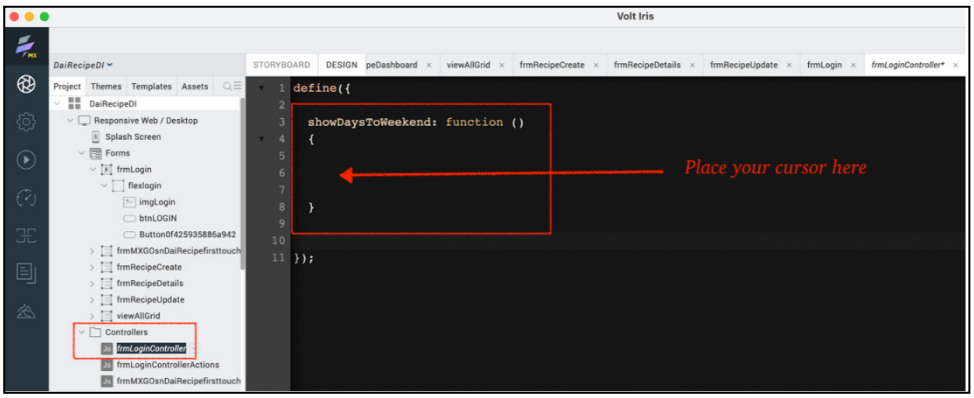
25. frmLoginController が Iris の中央のキャンバスに開きます。コントローラは主に空で、一番上の行に "define({" が、一番下の行に "});" が表示されているはずです。その間に関数を追加します。



26. この空の showDaysToWeekend JavaScript 関数をコピーしてコントローラに挿入します。

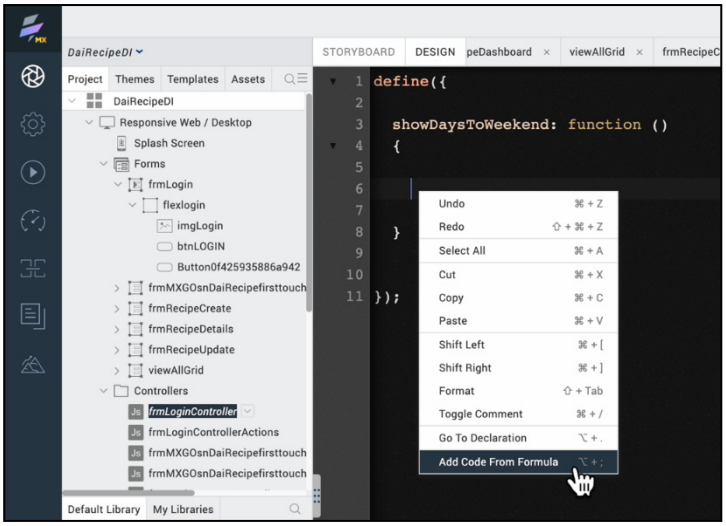
```
   showDaysToWeekend: function ()
   {

}
```



27. showDaysToWeekend 関数の中央にカーソルを置き、マウスを右クリックします。

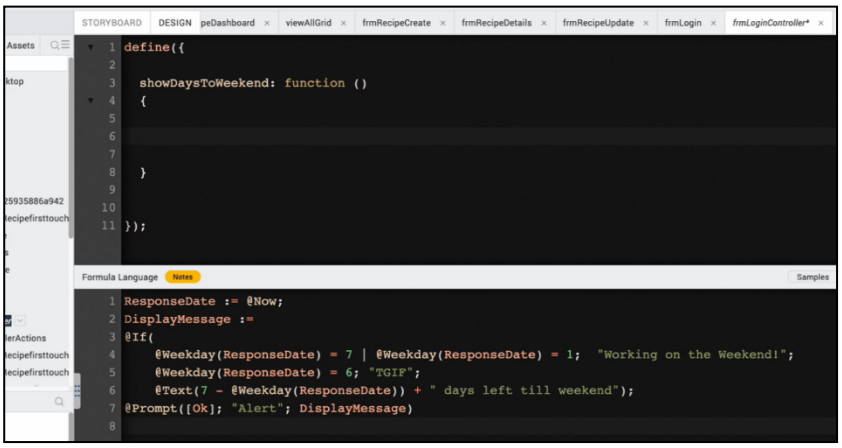
28. ポップアップウィンドウが表示され、下部に Add Code From Formula が表示されます。それを選択します。

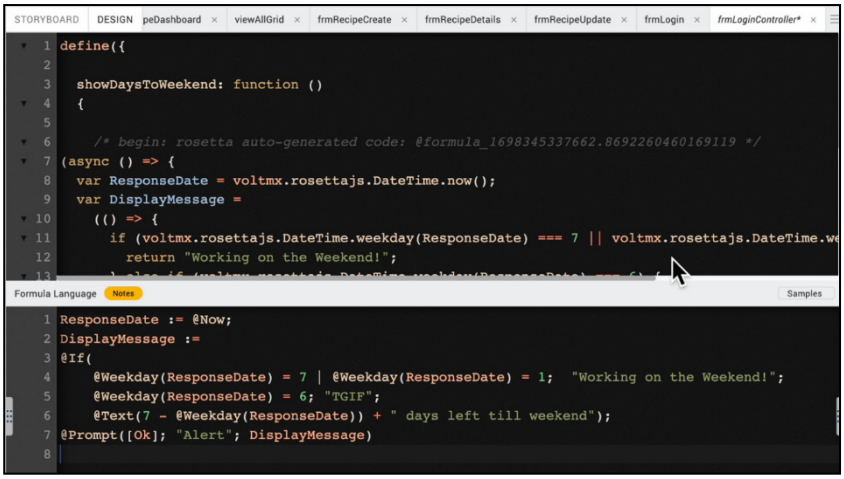


29. 現在のエディタが 2 つの部分に分割されます。上部は JavaScript コントローラのコードになります。Formula エディタは下の部分になります。Formula エディタは、ローコードのアクションスクリプトで見たのと同じ動作をします。

30. 先ほどのローコードアクションスクリプトと同じフォーミュラコードを使用します。上記のステップ 9 からこの Formula をコピーします。

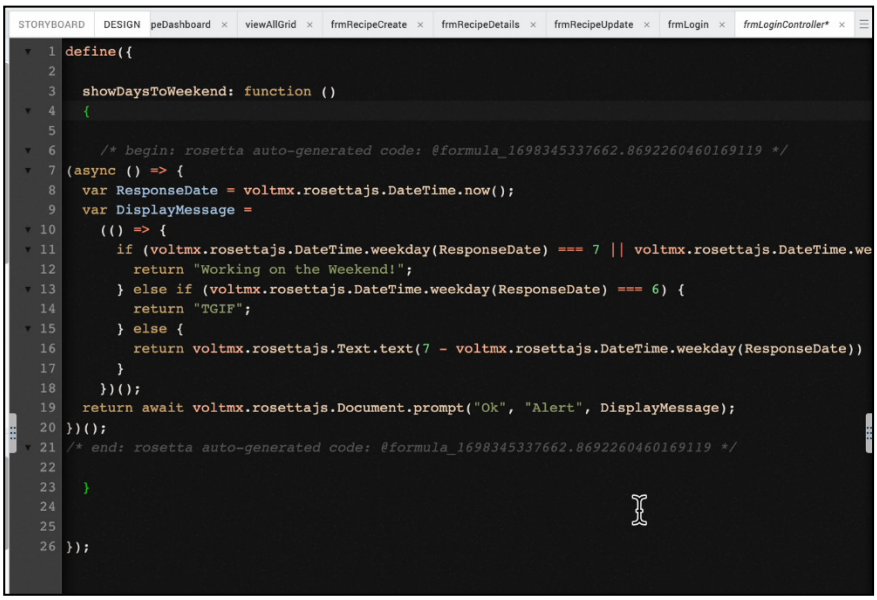
31. コピーした Formula を一番下の Formula エディタペインに挿入します。Formula は JavaScript に変換され、上のエディタペインに表示されます。以下のスナップショットをご覧ください。





32. 上の JavaScript エディタペインをクリックすると下の Formula エディタペインが表示されなくなります。

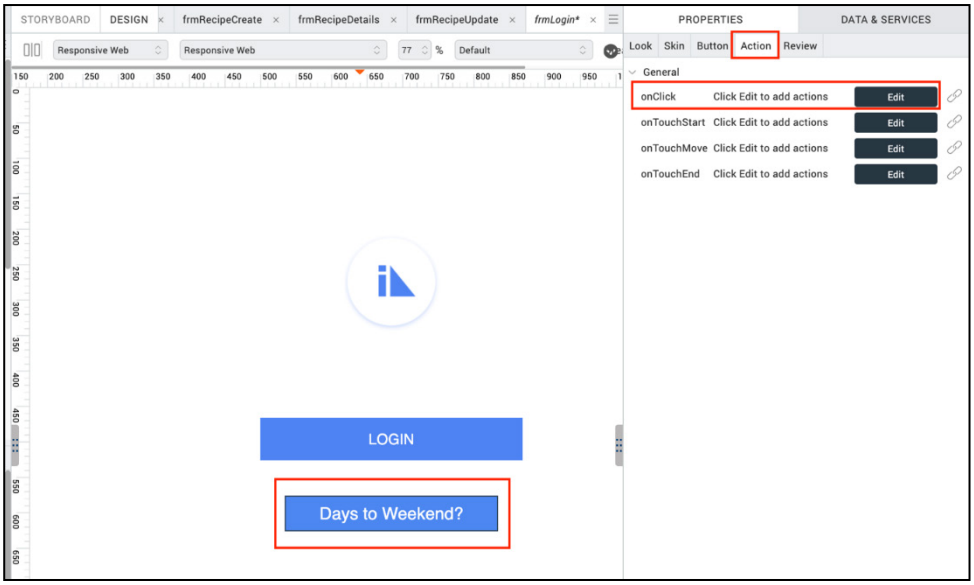
33. これでコントローラ内に変換された JavaScript コードが表示されるはずです。新しいコードは関数 showDaysToWeekend のボディにあるはずです。これで、frmLogin フォームからこの関数をコールできるようになります。



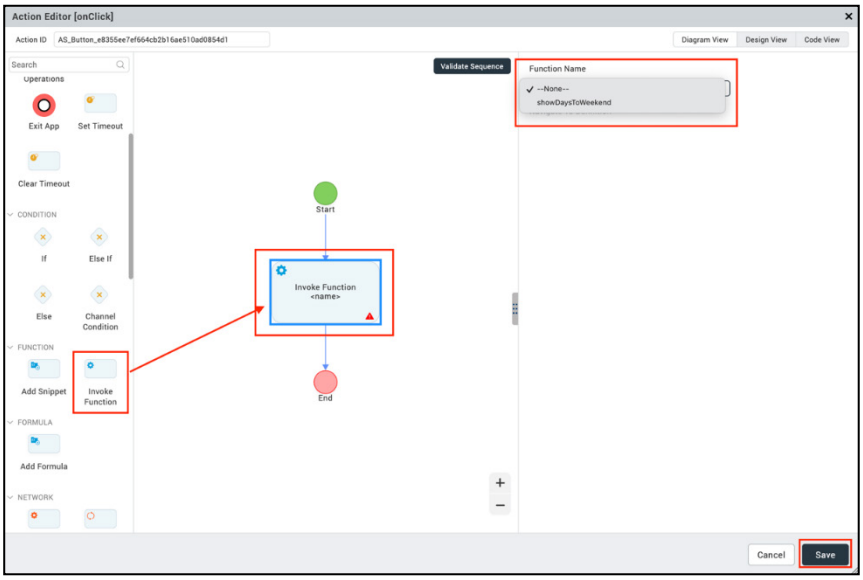
34. Iris プロジェクトを保存します。

35. frmLogin フォームに戻ります。ボタン Days to Weekend?" をクリックしてフォーカスを設定します。

36. PROPERTIES -> Action -> onClick event -> Edit をクリックします。



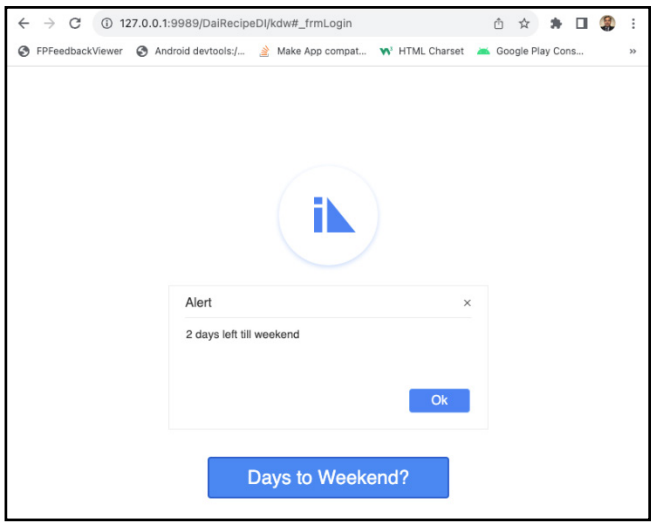
37. 左側のパレットから Invoke Function アクションノードをスクリプトにドラッグします。
38. Function Name の下のプルダウン コントロールを使用して showDaysToWeekend 関数を選択します。
39. Save をクリックして新しいアクション スクリプトを保存します。



40. Iris プロジェクトを保存します。
41. 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 Settings をクリックします。

42. ライブプレビュー設定で **Responsive Web** チャンネルのみにチェックが入っ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 **Debug** 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 プレビューアプリをビルドして実行します。

アプリをテストします。**Days to Weekend** ボタンをクリックすると、ポップアップウィンドウが表示されるはずです。



Days to Weekend ボタンをクリックするたびにポップアップウィンドウが表示されたら成功です。これで、コントローラエディタを使用して VoltFormula を追加するパートのレッスンが完了しました。

まとめ

Volt Iris プロジェクトに VoltFormula を追加するこのレッスンを完了しました。

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。

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。