



# HCL Volt MX

## Volt MX Jumpstart Exercise – Part 2

このドキュメントは、「Volt MX ジャンプスタートトレーニング」の以下のクラス演習を対象としています。

パート 1では、画面開発をカバーします。

パート 2 では、アクションについて説明します。

この演習が終了すると、ログインして自動車保険に加入するための4画面アプリケーションが完成します。

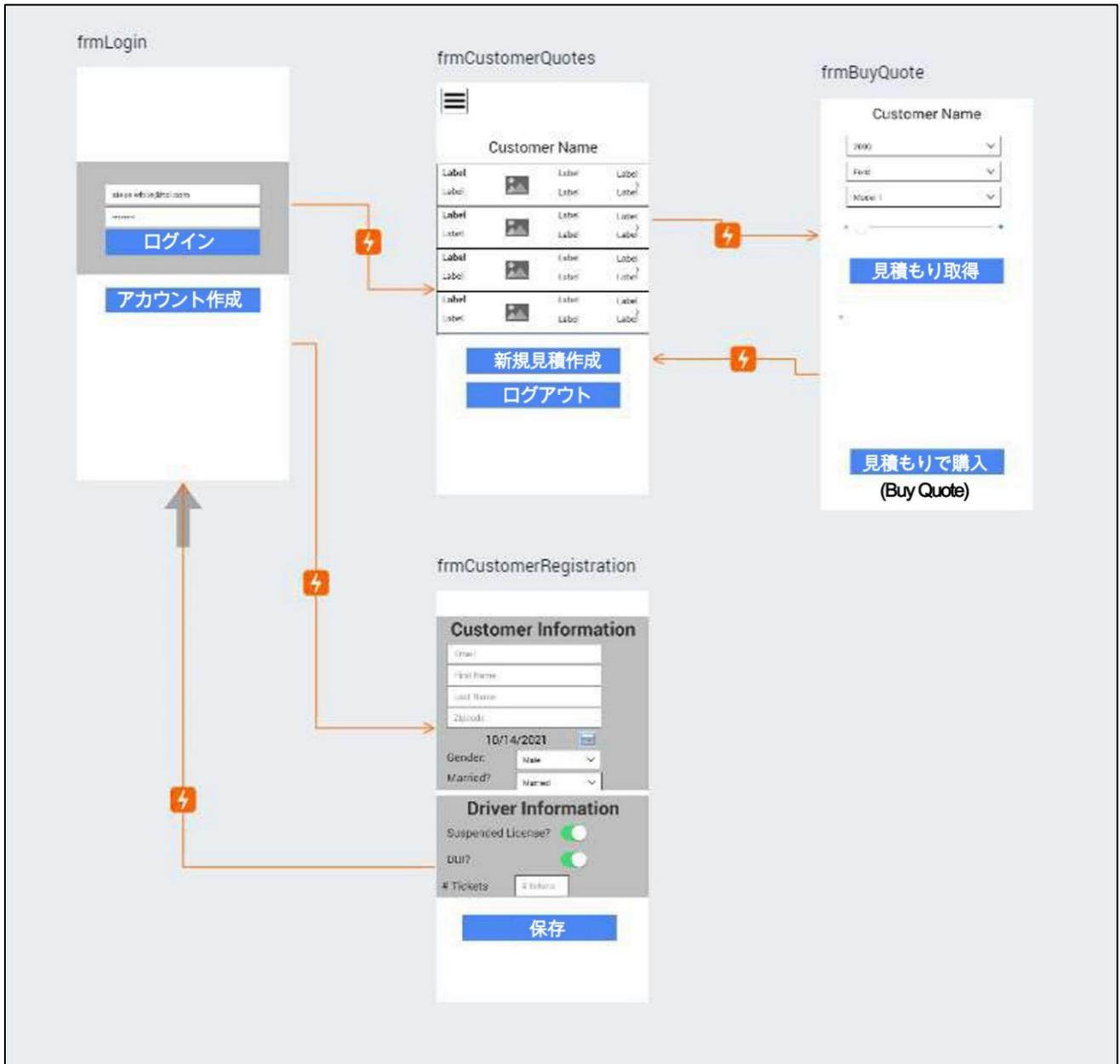
.

## Contents

Volt MX Jumpstart Exercise – Part 2 .....	1
概要 .....	3
Foundry のトレーニングアプリ .....	3
連携・統合サービス .....	4
InsuranceLogin .....	4
multiquotesForInsurance .....	4
getMultiQuotes .....	4
オブジェクトサービス .....	5
InsuranceDataObj.....	5
画面設計 .....	6
ログイン画面.....	6
顧客情報フォーム .....	7
顧客見積もり画面 .....	8
顧客情報を編集するハンバーガーマニュー .....	9
テスト .....	10
バックエンドの要件 .....	10
提供されるアセット .....	10

## 概要

以下のアプリを作成します。これは、Web、モバイルのいずれのアプリにすることができます。



このアプリケーションはLive Previewで動作します。公開する必要はありません。

クラウド（トライアルクラウドで十分）またはローカルのオンプレミスインストールで完成させられます。

## Foundry のトレーニングアプリ

インポートとパブリッシュが可能なのは、IrisTrainingV9 Foundryアプリです。

**サンプルのトレーニングアプリをまだインポートおよび公開していない場合は、Foundryアプリケーションをインポートおよび公開してください。**

この演習では、トレーニング用Foundry サービスの作成方法は含まれていません。プロジェクトで使用される Identity、Integration、Object サービスが演習に含まれています。

## 連携・統合サービス

### InsuranceLogin

これは、ユーザー名とパスワードを受け取り、成功または失敗の結果を提供するシンプルなユーザーリポジトリです。

サンプルユーザー（steve.white@hcl.com、パスワードは Volt@123）があらかじめ用意されています。このユーザーを編集したり、テスト用に新しいユーザーを作成することができます。

\*\*\*ユーザーは開発コンソールからのみ追加可能です。

このサービスには2つの入力パラメーター（ユーザーとパスワード）があります。この演習の中で、ログインフォームのフィールドをIdentityサービスの入力パラメーターにマッピングします。

### multiquotesForInsurance

multiquotesForInsuranceは、単一オペレーションを持つモック・データ・サービスのサンプルです。

### getMultiQuotes

#### インプットパラメーター

```
carYear
carMake
carModel
carMileage
carTrim
```

#### サンプル出力

```
{
  "premium_adder_comments": "year > 2018", "insurance_type":
  "Auto",
  "approval_comments": "Approval Comments > 2018",
  "approval_status": "Approved",
  "car_trim_level": "+AWD +Turbo",
  "opstatus": 0,
  "AutoInsuranceQuotes": [
    {
      "premium_base": "400",
      "premium_adder": "1200", "id":
      "Progressive",
      "ins_type": "Auto",
      "provider_logo":
      "https://www.progressive.com/content/images/domainprogressive/wh3/base/
      icons/logo-progressive.svg"
    }...
  ],
  "term_length": "6",
  "httpStatusCode": 200
}
```

## オブジェクトサービス

### InsuranceDataObj

このオブジェクトサービスには2つのオブジェクトが含まれます。

#### Customer

Customerオブジェクトは、Customer Screen上で顧客情報の保存と更新を行うためのものです。

#### フィールド

```
customerId  
emailAddress  
lastName  
firstName  
DOB  
Gender  
ticketCount  
zipCode  
userName  
maritalStatus  
isSuspendedLic  
hasDUI  
CreatedBy  
LastUpdatedBy
```

#### Methods

```
Get  
Post (Create)  
Put (Update)
```

#### Quote

```
Fields  
quoteId  
customerId  
carMake  
carModel  
carYear  
carTrimLevel  
termLength  
premium_base  
premium_adder  
premium_adder_comments  
approvalStatus  
approvalComments  
insuranceType  
insuranceProvider  
insuranceProviderLogo  
CreatedBy  
LastUpdatedBy
```

#### Methods

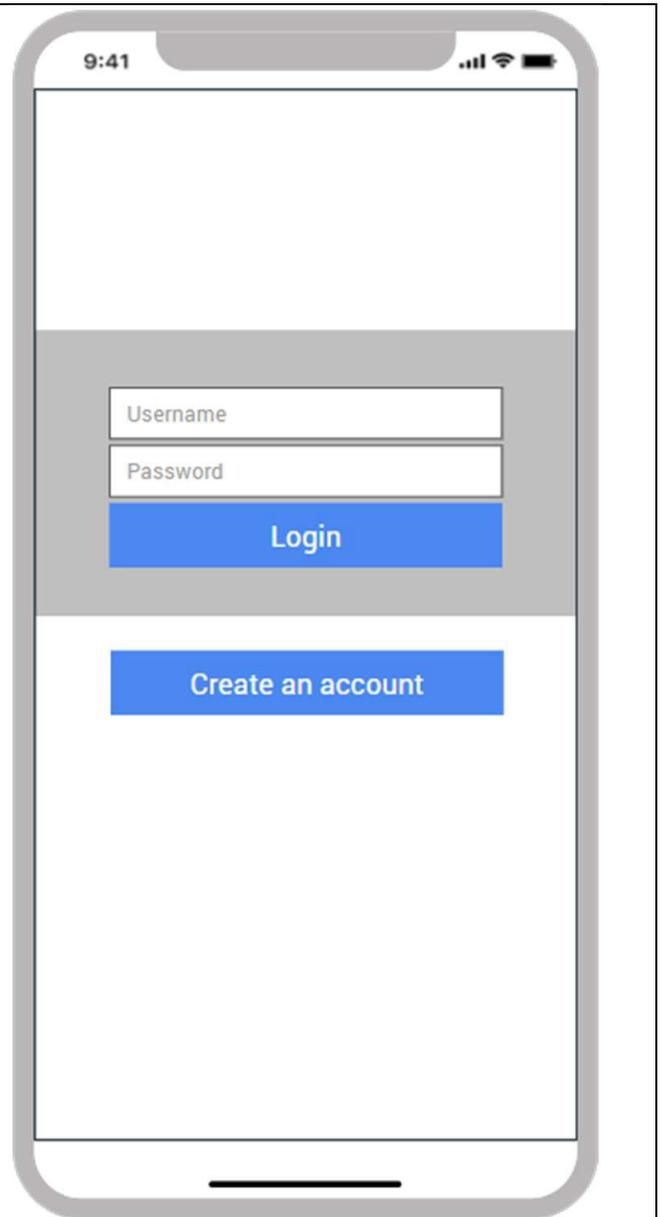
```
Get  
Post (Create)
```

## 画面設計

### ログイン画面

Fields: UserName/Password

- ・ ログインボタンにアクションを追加します。
- ・ ユーザー名とパスワードをサービスにマップさせます。
- ・ 結果をチェックし、ログインに成功した場合はCustomer Quote Screenへ遷移します。
- ・ ログインに失敗した場合、エラーを表示します。
- ・ Create an Account (アカウント作成)ボタンにアクションを追加し、Customer Screenへ遷移できるようにします。



## 顧客情報フォーム

- Saveボタンにアクションを追加する。
- 見積書(quote)オブジェクトサービスのcreate操作で、顧客情報を保存する。
- 保存に成功したら、ログイン画面に戻る。

9:41

### Customer Information

Email

First Name

Last Name

Zipcode

10/14/2021

Gender: Male

Married? Married

### Driver Information

Suspended License?

DUI?

# Tickets # tickets

Save

## 顧客見積もり画面

- ・ 見積書オブジェクトのサービスのgetメソッドを使用して、その顧客のすべての見積書をロードします。
- ・ 顧客オブジェクトのquoteサービスのgetメソッドを使用して、現在の顧客情報をロードします。
- ・ 顧客情報を表示(expose)するためのアニメーションアクションをハンバーガーメニューに追加します。
- ・ 新規見積もり(New Quote)]ボタンにアクションを追加し、[新規]に移動するようにします。

### 注記

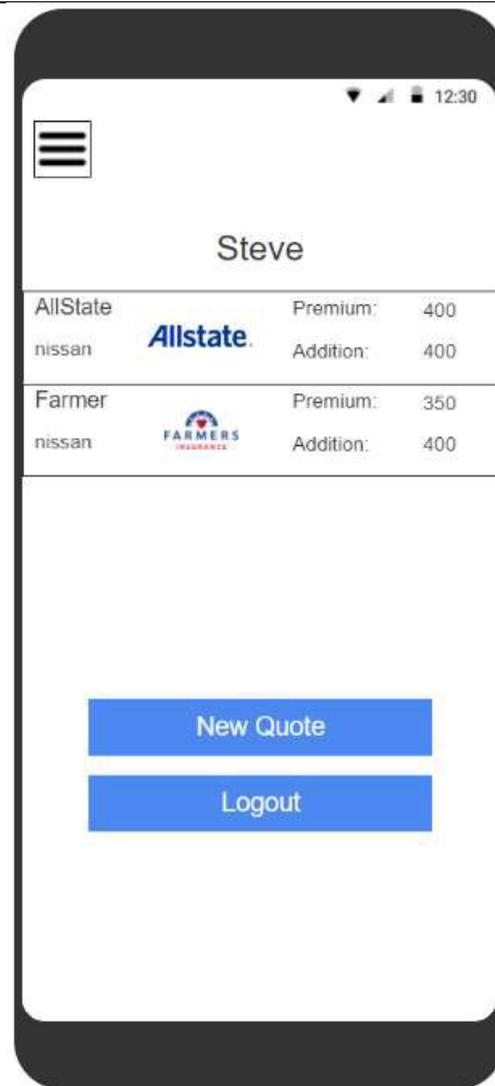
- ・ どこから遷移してきたかを確認します。
- ・ この画面は、ログイン画面または見積もり購入(by quote)画面から移動してきます。どちらの場合も、サービスにマッピングするために必要な顧客情報を渡されていることを確認します。

### 既知の問題

- ・ AppPreviewではハンバーガーメニューの画像が表示されません。

### 自分でやること

- ・ さまざまなテンプレートを作成し、テストすること。



## 顧客情報を編集するハンバーガーメニュー

- ハンバーガーメニューが "open" の位置にあるとき、顧客情報を編集できます。
- Saveにアクションを追加して、ウィジェットのデータをマッピングし、Customerオブジェクトのサービスのupdateメソッドを呼び出すようにします。
- アニメーションアクションを追加し、ハンバーガーメニューが選択されると、現在のビューにある顧客の見積もり情報を保持した状態で画面が元のレイアウトに戻るようになります。



The screenshot shows a mobile application interface with a hamburger menu icon in the top right corner. The main content is divided into two sections: "Customer Information" and "Driver Information".

**Customer Information**

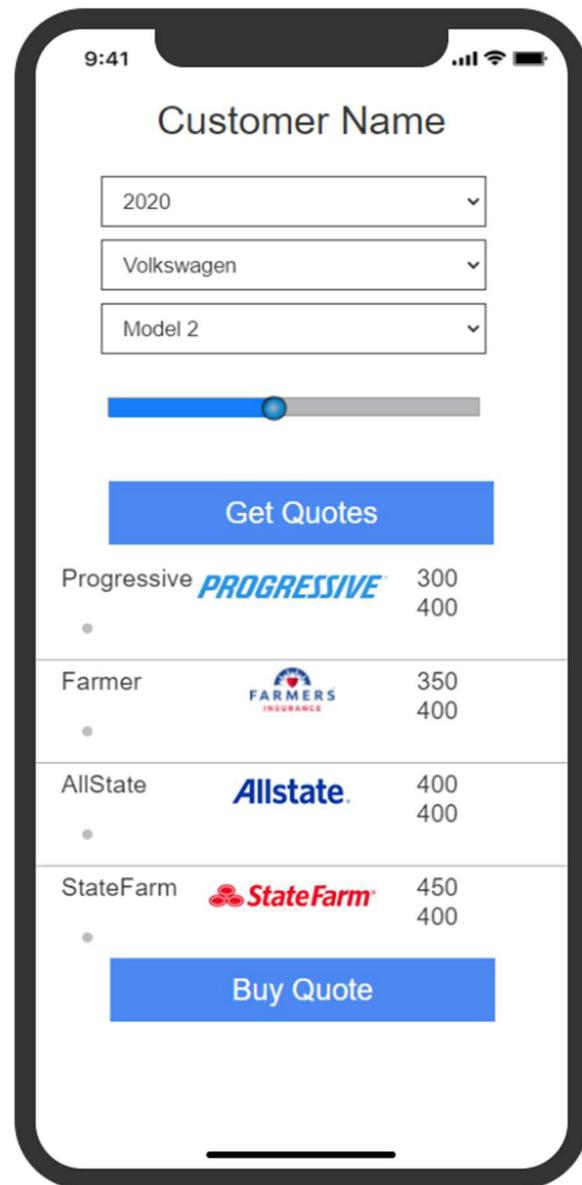
- Input field: steve.white@hcl.com
- Input field: Steve
- Input field: White
- Input field: 10058
- Date field: 03/01/1986
- Gender dropdown: Male
- Marital status dropdown: Married

**Driver Information**

- Toggle switch: Suspended License? (off)
- Toggle switch: DUI? (off)
- Input field: # Tickets (0)

At the bottom, there is a large blue button labeled "Update".

- GetQuotesボタンにアクションを追加し、ウィジェットの選択内容をmock date integrationサービスに送信します。
- 結果をセグメントにマッピングします。
- セグメントのクリックイベントにアクションを追加し、選択した情報を見積もりオブジェクトサービス/作成メソッドにマッピングし、見積もりを保存します。
- 成功したら、顧客見積もり画面に移動します。



## テスト

この演習のすべての機能は、ライブプレビューでテストできます。

## バックエンドの要件

この演習を完了するには、サンプルのFoundryアプリをインポートして公開する必要があります。Foundryコンソールで新しいユーザーをシステムに追加する必要があります。この演習では、自己登録はサポートされていません。

## 提供されるアセット

1. IrisTrainingApp (Foundry)
2. TrainingApp (Iris)