Updating the Dreamhouse App with external datasource - Lab 1

The app currently uses postgres db via Heroku. In this section, we will learn how to update the property data using an external datasource, without taking the app down via Anypoint platform.

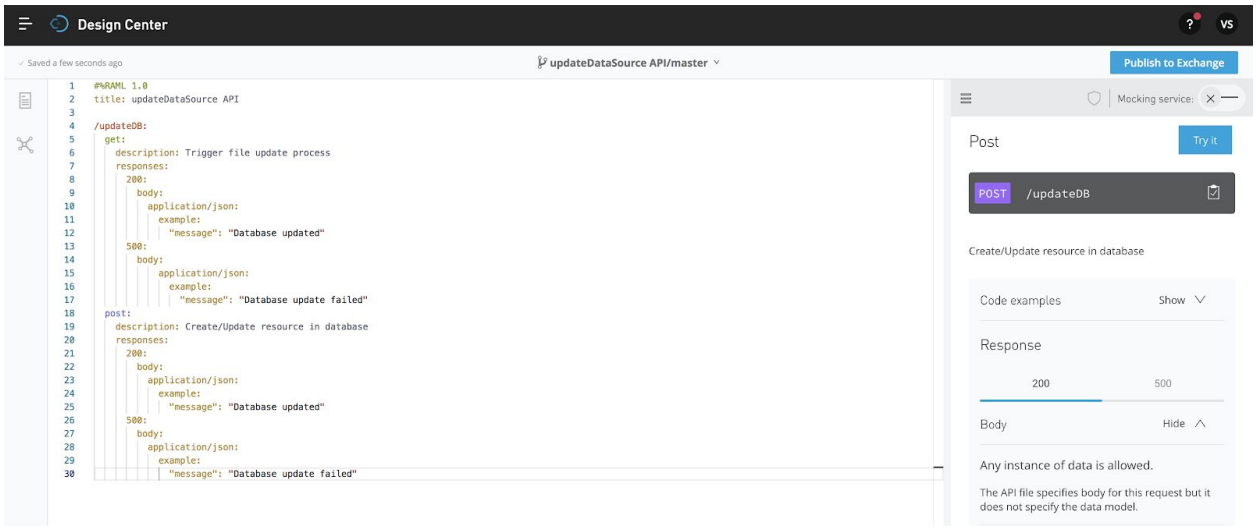
Objective:
In this use case, we will leverage an external data-source and use Anypoint platform to read, transform and update the Dreamhouse Application data. External datasource, could be a Saas application, database or a simple flat file. In our today's walkthrough, we will leverage a CSV file with Property data and update the Postgres Database

Prerequisite:

1. Navigate to the github link and download the .csv datasource file
2. Download the updateDataSource API.raml as well

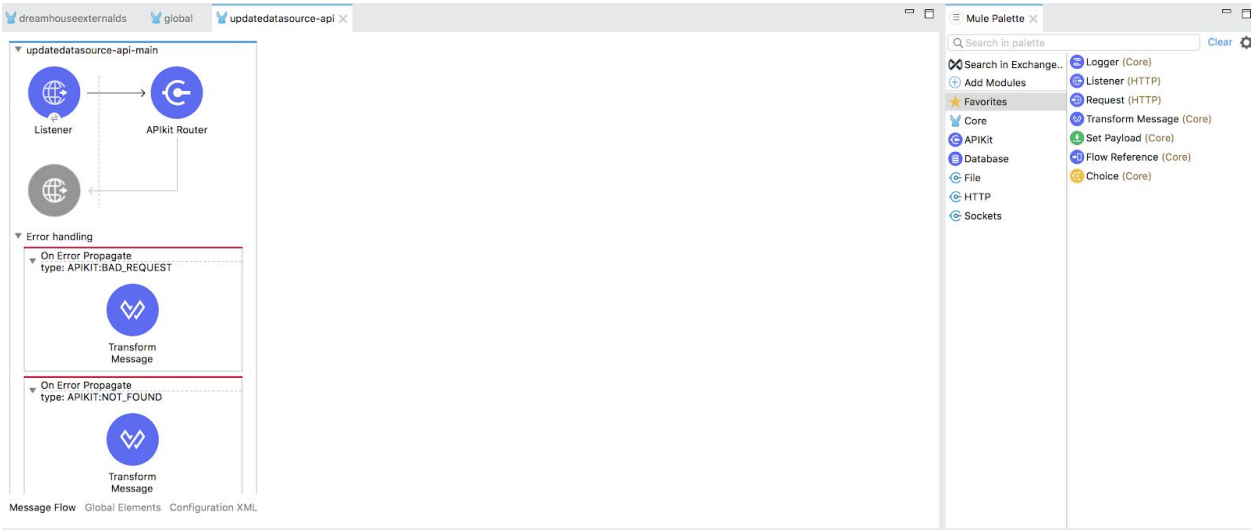
Let's start with API design first approach:

1) Log in to anypoint.mulesoft.com
2) Navigate to Design Center
3) Hit 'Create' and choose API Specification
4) You could either choose the manual way by creating a spec in RAML format or use the visual designer tool



Now, we will move to the implementation:

1) Open Studio Application
2) Create ‘New Mule Project’ and fill in the project name
3) Before clicking ‘OK’, check the ‘API definition’ and choose ‘Design Center’ as the source
4) Look for the API Spec that you created Design Center, select it and hit okay
5) Review the pre-populated flows (diagram below) that were created based on the API Contract



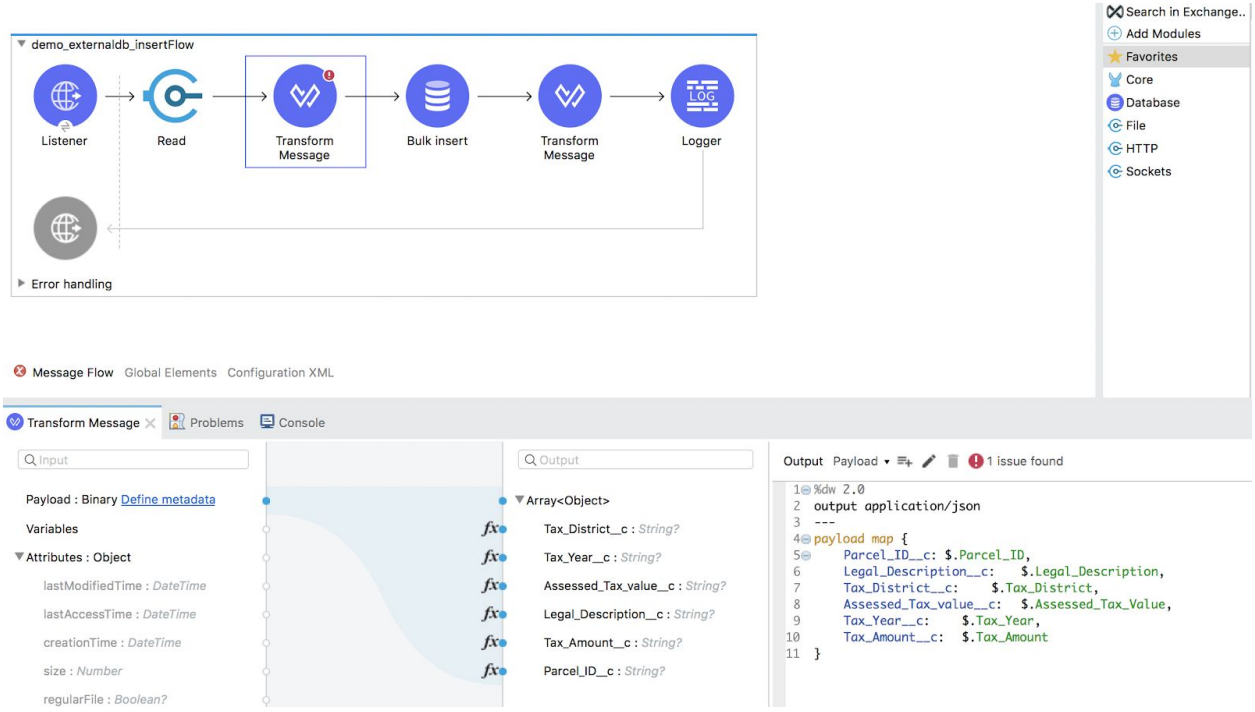
6) Create another ‘mule configuration’ and name it as ‘implementation’

7) We will now implement the logic under ‘implementation’, by leveraging the file and database connector from Mule Palette & reference it in the master XML file (pre-populated by API Spec)
    a) Under implementation, drag ‘n’ drop the file (read) connector to the canvas
    b) Create a folder ‘file’ under /src/main/resources and copy the .csv file
    c) Select the file connector and under ‘File Path’, add the csv location as : **${app.home}/file/ <file name>**. (*$app.home represents the project file structure path)*
    d) Add the ‘Transform’ connector & convert the payload to JSON:

```
%dw 2.0
output application/json
---
payload map {
    Parcel_ID__c: $.Parcel_ID,
    Legal_Description__c:    $.Legal_Description,
    Tax_District__c:     $.Tax_District,
    Assessed_Tax_value__c:   $.Assessed_Tax_Value,
    Tax_Year__c:     $.Tax_Year,
    Tax_Amount__c:   $.Tax_Amount
}
```

e) Add the database connector (bulk insert). By doing so, you are able to add all records in 1 shot. In the query section copy the below query:

```
INSERT INTO Property__c(Parcel_ID__c, Legal_Description__c, Tax_District__c,
Assessed_Tax_value__c, Tax_Year__c, Tax_Amount__c)
VALUES (:Parcel_ID__c, :Legal_Description__c, :Tax_District__c,
:Assessed_Tax_value__c, :Tax_Year__c, :Tax_Amount__c)
```



Note: Test the logic using API Console/REST Client and review the updated data in the Dreamhouse App

Creating a mobile notification alert based on a custom preference in Dreamhouse App - Lab 2

The biggest advantage of using Anypoint platform is the end-to-end solution infrastructure and the availability of Anypoint Exchange. The exchange offers core system connectors, that can easily be configured without the need of any complex code. It includes a wide range of SaaS systems, complex systems like mainframes and pre-built system connectors (built by MuleSoft and offered in design tools). These connectors are leveraged during implementation while using Anypoint Studio or Anypoint Flow designer (online design tool)

Objective

In this use case, we will be creating an API to trigger SMS notifications based on a custom preference. The custom preference dataset can be queried from Salesforce and transformed as a message body for the notification. For this exercise, lets create a dataset of *property data* based on a certain zip code, and have the results displayed in the form of an SMS message.

We have already abstracted a notification API, that currently uses Twillio as the source system:
URL: notificationapivs.us-e2.cloudhub.io/api/sendSMS
Query Parameter: To_Phone_Number (*US Phone number only*)
Payload Requirements: Non- Array message body (*application/json*)

Let's start implementing the solution

1) Navigate to anypoint.mulesoft.com
2) Choose 'Design Center' and click Create
3) Select 'Create new Application'
4) We will now implement the solution, using the following connectors:
    a) HTTP Listener & Requestor
        i) Create endpoint
        ii) Add notification connection details
    b) Salesforce Connector
        i) Add connection details
        ii) Add the select query for listing houses based on Zip code:

```
SELECT Name, Zip__c, Price__c FROM Property__c where Zip__c = '02420'
```

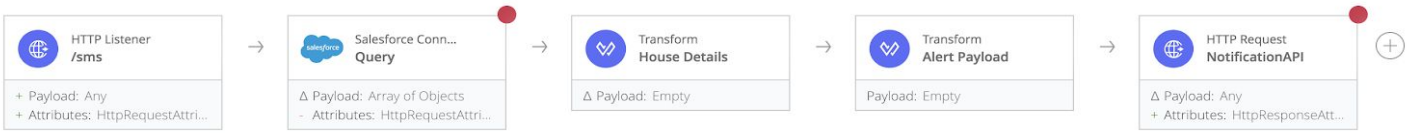
    c) Transform Connector
        i) Transform data to JSON, to collect zip, name and price information:

```
%dw 2.0
output application/json
---
payload map {

    "message": $.Name ++ " is priced at USD " ++ $.Price__c ++ " for area code " ++ $.Zip__c
}
```

ii) Transform data to non-array JSON message, for notification message body

```
%dw 2.0
output application/json
---
payload.message joinBy " |||| "
```



iii) Finally, add a http request connector and configure it using the notification API details

iv) Run the application and copy the URL

v) Either in a browser or REST client, add the URL with the queryParameter and your phone number

Note: Test the logic by reviewing the message body/content in SMS

Credit Pre-approval check - Lab 3

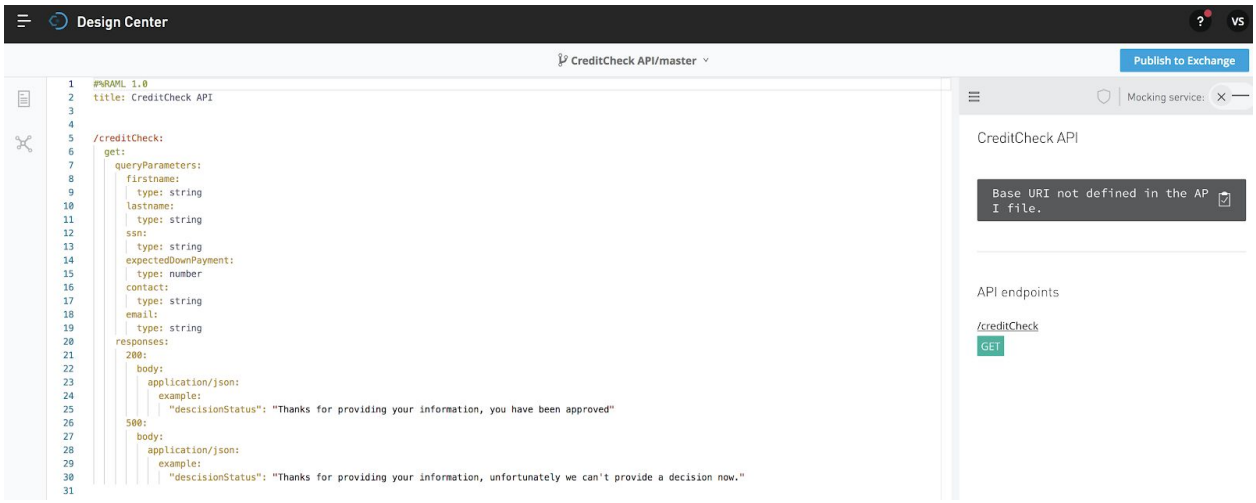
Orchestrating and triggering flows based on certain business logic, makes an integration solution more intelligent & efficient. Let's implement this logic based on the techniques learnt in the previous use cases.

Objective

In this use case, the end-user upon finalizing a property wants to validate their financial status by filling a pre-approval form. A pre-approval process involves checking credit scores from different bureaus, calculating the result based on estimated down payment in USD and other financial factors. We will start with creating an **experience API** that accepts user data as input and then linking the API to the existing process and system APIs

Let's start implementing the solution

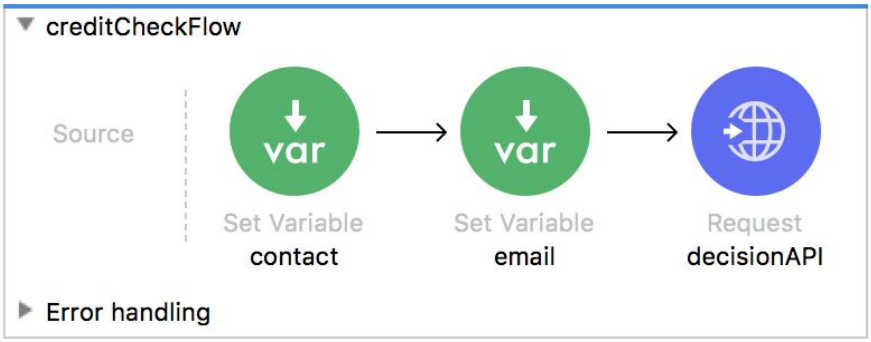
1) Using a design first approach, let's start creating an API Spec
2) Configure the API Spec to include a GET Method, that accepts the end-user's information in the form of Query Parameters:
    a) First Name, Last Name, SSN, Contact and Downpayment



3) Create a new Project in Anypoint Studio and import the Api specification
4) For this solution, we will have touch points for all the 3 levels - Experience API (where the user adds their information), Process API (where the decision is determined based on certain calculation & credit score) and System API (credit score database(s)):

Note: Test the logic by inquiring multiple users

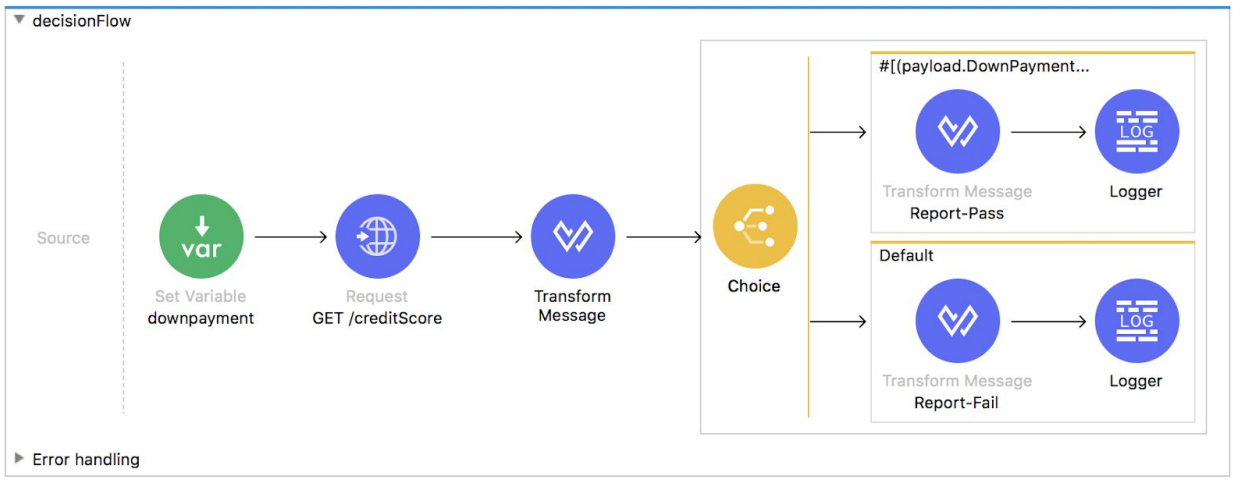
Experience API (CreditCheck API)



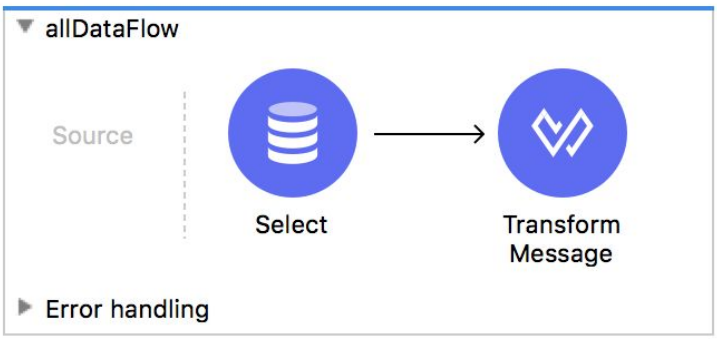
1. Save the contact and email in Variables
2. Add an HTTP request and add the following details:
    a. decisionapi.us-e2.cloudhub.io/api/decision
    b. Query Parameters - firstname, lastname, ssn, expectedDownPayment

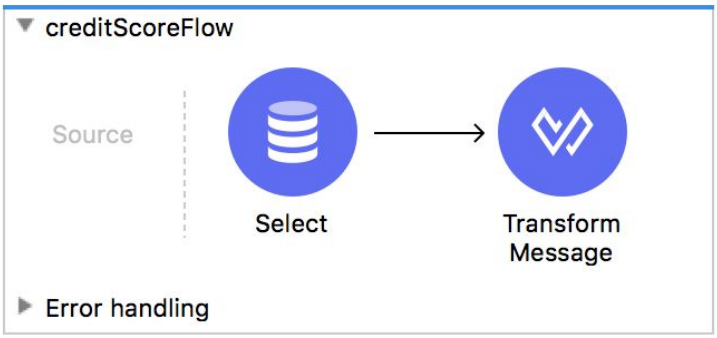
---

Process API (decisionAPI)



System API (Credit Score API)


allDataFlow
Source
Select
Transform Message
Error handling


creditScoreFlow
Source
Select
Transform Message
Error handling