FINAL PROJECT

WORKFLOW COORDINATION WITH SECURITY FEATURES

Patrick Walter
B00707506
CSCI 4145 Cloud Computing
Winter 2018

Table of Contents

URL of Web Service

http://project.us-east-1.elasticbeanstalk.com/

Environment and Resources

Hardware:

Acer Aspire 7, Intel® Core™ i5-6200U CPU @ 2.3GHz, 8 GB RAM

Software:

Windows 10 Home, WinRar, Google Chrome, Internet Explorer, PHPStorm, MAMP, NotePad++, MariaDB Client

Languages: PHP

Cloud Services:

Amazon Web Service's Relational Database Service, Azure Logic Apps, Azure Service Bus, Gmail

Files

config.php	Sets up the variables names for the database running on AWS RDS
common.php	Declares a escape function to be used for outputting
index.php	A index page to find all of the various agents websites since they are all hosted on the same url
HomepageEMP.php	The homepage of the employer EMP. Used to look up employee records.
HomepageRE.php	The homepage of the real estate RE. Used to look up various records and compile them.
INSfront.php	A homepage for the insurance company which doesn't have any front end capabilities. Just instructions of how to get the insurance record.
auth.php	This is an implementation of JWT. I did not write it.
MBRcreate.php	The mortgage brokers front page for submitting applications

Database Design

Overview

This project uses five tables. Each of the tables represents a database of a different organization.

projectmbr: stores the applications, this is populated by MBR front end.

projectemp: stores the employee records. Accessed by EMP front end.

projectinsinc: stores the insurance records. Accessed by RE front end.

projectre: stores the real estate records. Accessed by RE front end.

projectmun: stores the municipality records. Accessed by MUN front end.

Database Access

MariaDB (mysql) running on AWS RDS:
host: a4.cxiyvutsdox7.us-east-1.rds.amazonaws.com
username:pt365049
password:Dark1010

Database Schema and Sample Populations

```
MariaDB [assignment4]> select * from projectinsinc;
+-------+--------------+------------+
| misid | insuredvalue | deductible |
+-------+--------------+------------+
|     1 |       500000 |       5000 |
|     2 |       300000 |       3000 |
|     3 |       450000 |       4500 |
|     4 |       588000 |       5080 |
+-------+--------------+------------+
```

```
MariaDB [assignment4]> SELECT * FROM projectemp;
+-------+------------+--------+-----------+
| empid | name       | salary | startdate |
+-------+------------+--------+-----------+
|     1 | Bob Smith  |  50000 |      2008 |
|     2 | Bob Smith  |  75000 |      1998 |
|     3 | Doug Smith |  79000 |      1994 |
|     4 | Mike Ross  |  85000 |      2012 |
|     5 | Tom Sawyer |  45000 |      2014 |
+-------+------------+--------+-----------+
```

```
MariaDB [assignment4]> select * from projectmun;
+-------+------+
| misid | code |
+-------+------+
|     1 | 1011 |
|     2 |  101 |
|     3 | 1001 |
|     4 | 1011 |
|     5 |  101 |
|     6 | 1111 |
|     7 | 1001 |
+-------+------+
```

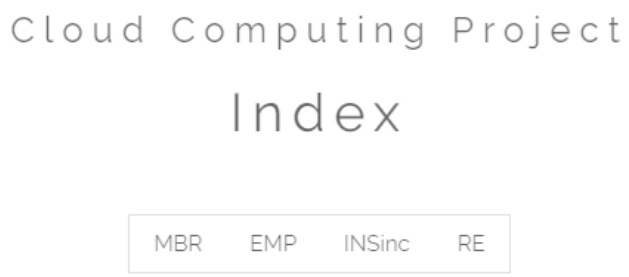
```
MariaDB [assignment4]> select * from projectre
    -> ;
+-------+----------------+
| misid | appraisedvalue |
+-------+----------------+
|     1 |         670000 |
|     2 |         540000 |
|     4 |         980000 |
|     5 |         350000 |
|     6 |         550000 |
|     7 |         450000 |
+-------+----------------+
```

```
+--------+----------------+-------+-----------+-----------------+
| mortid | name           | misid | mortvalue | email           |
+--------+----------------+-------+-----------+-----------------+
|     12 | Patrick Walter |     1 |    500000 | pt365049@dal.ca |
|     13 | Rick Rickson   |     2 |   2500000 | pt365049@dal.ca |
|     14 | Rick Rickson   |     2 |   2500000 | pt365049@dal.ca |
|     15 | Patrick        |     1 |    500000 | pt365049@dal.ca |
|     16 | ppp            |     1 |         1 | pt365049@dal.ca |
|     17 | Patrick        |     1 |         1 | pt365049@dal.ca |
|     18 | Patrick        |     1 |         5 | pt365049@dal.ca |
|     19 | Patrick        |     1 |         1 | pt365049@dal.ca |
+--------+----------------+-------+-----------+-----------------+
```

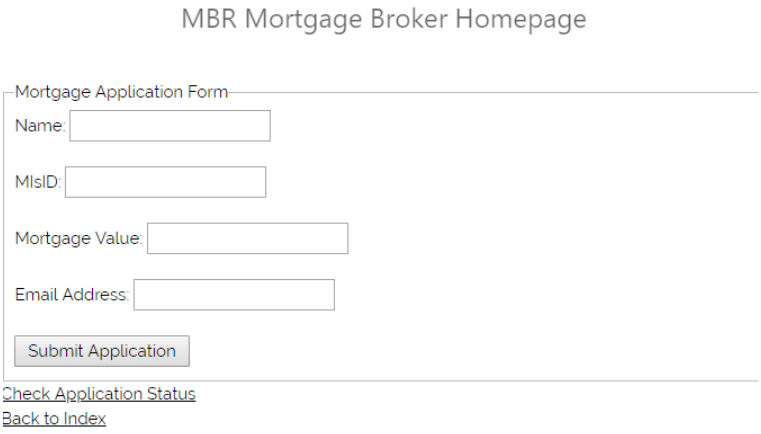
Front End Design

Index

index.php: This is how to access each of the different organizations homepages.

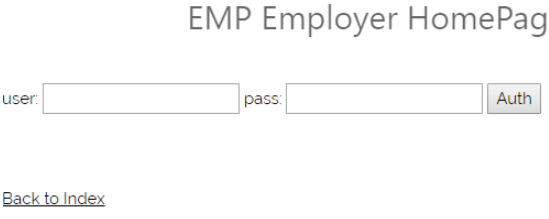


Mortgage Broker



Employer

Employer user: testuser pass: mypw



After JWT authentication token generated

EMP Employer HomePage

Enter your Employee ID:

Employee ID: I Agree

Back to Index

Real Estate

Real Estate: HomepageRE.php

Real Estate RE

Get an Appraisal

Back to index

Appraisal Form

Property Appraisal Form

Name:

MlsID:

Mortgage ID:

Submit Appraisal

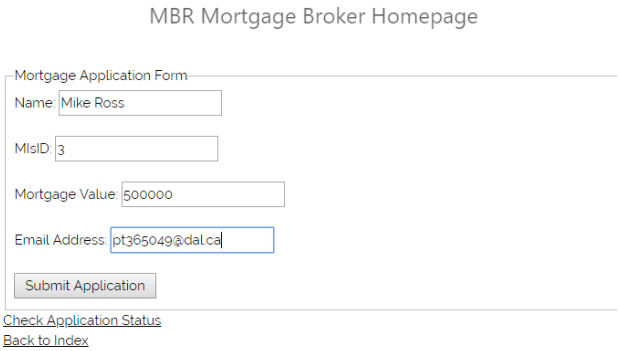
Real Estate RE

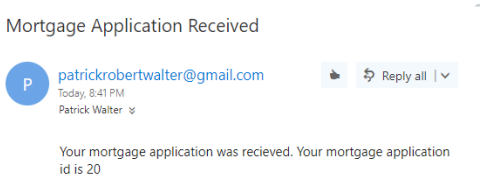
Get an Appraisal

Back to index

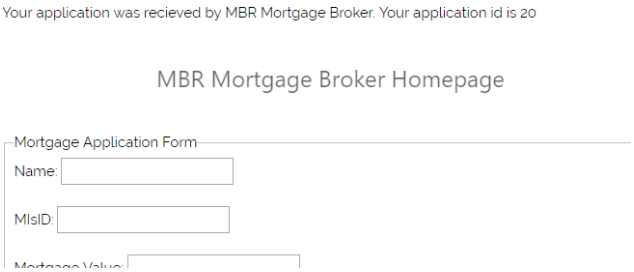
Front End Testing

Mortgage Broker

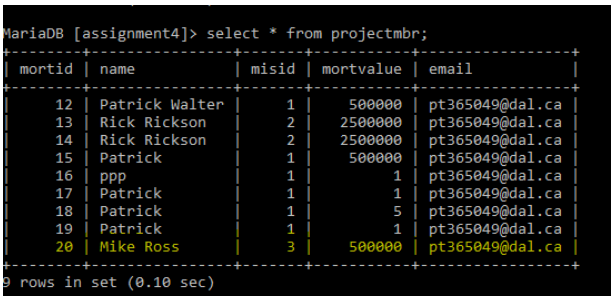




Form filled and submitted. Response on front end



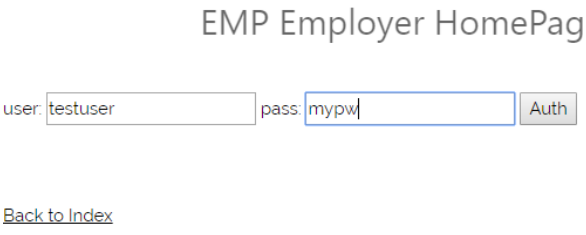
Email Update Received. Record Added to table.



```
MariaDB [assignment4]> select * from projectmbr;
+--------+----------------+-------+-----------+----------------+
| mortid | name           | misid | mortvalue | email          |
+--------+----------------+-------+-----------+----------------+
|     12 | Patrick Walter |     1 |    500000 | pt365049@dal.ca |
|     13 | Rick Rickson   |     2 |   2500000 | pt365049@dal.ca |
|     14 | Rick Rickson   |     2 |   2500000 | pt365049@dal.ca |
|     15 | Patrick        |     1 |    500000 | pt365049@dal.ca |
|     16 | ppp            |     1 |         1 | pt365049@dal.ca |
|     17 | Patrick        |     1 |         1 | pt365049@dal.ca |
|     18 | Patrick        |     1 |         5 | pt365049@dal.ca |
|     19 | Patrick        |     1 |         1 | pt365049@dal.ca |
|     20 | Mike Ross      |     3 |    500000 | pt365049@dal.ca |
+--------+----------------+-------+-----------+----------------+
9 rows in set (0.10 sec)
```

Employer JWT Authentication

Employer Authentication with JWT Token



Output the token for testing display

EMP Employer HomePage

{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MTg3LCJleHAiOjE1MjMzMjUzMzl9.eCPcgoQCy6aVe7tWVYH9tVoHetETIpaa8tHsTcFLHYc"}

Enter your Employee ID:

Employee ID: I Agree

Back to Index

Authentication function

```php
function authenticate () {
    $db_usr = "testuser";
    $db_usr_pw = "mypw";
    $db_usr_id = 187;
    $secret_key = 'some_test_key';
    $valid_for = '3600';
    if ($_POST['usr'] && $_POST['pw']) {
        $usr = $_POST['usr'];
        $pw = $_POST['pw'];
        if ($usr == $db_usr && $pw == $db_usr_pw) { // took out the hash part: if ($usr == $db_
            $token = array();
            $token['id'] = $db_usr_id;
            $token['exp'] = time() + $valid_for;
            $json_token =json_encode(array('token' => JWT::encode($token, $secret_key)));
            echo $json_token;
            return $json_token;
```

After its authenticated. The record is retrived from the database and sent to the logic app.

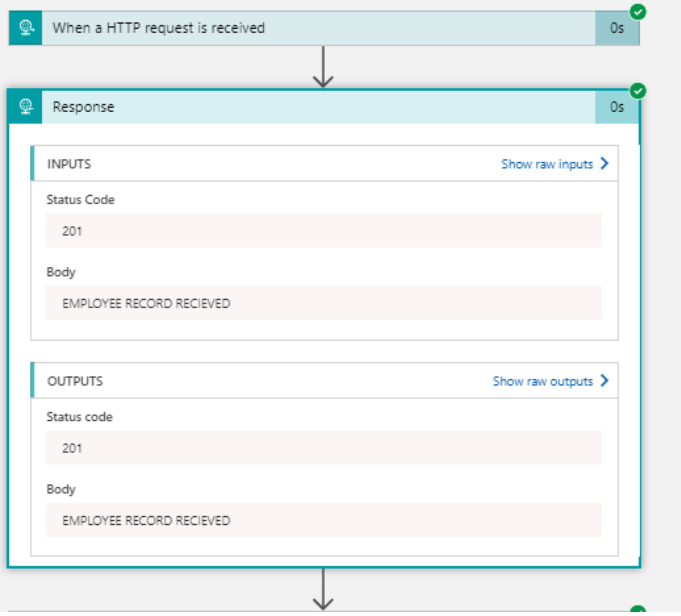
Results

Employee ID: 1 Authenticated.

Response from Mortage Broker Server: EMPLOYEE RECORD RECIEVED

EMP Employer HomePage

user: pass: Auth



Insurance Records

Insurance front end

Property Appraisal Form

Name: Mike Ross

MlsID: 3

Mortgage ID: 20

Submit Appraisal

Records complied and shown on front end

Results

Real Estate House Appraisal Record:

Date: Tuesday, April 10, 2018

Name: Mike Ross

Mortgage ID: 20

Insurance Record From INSinc:

MISID: 1

Insured Value: $500000.00

Deductible: $5000.00

Municipality Service Record From MUN:

MISID: 1

Utilities Code: 1011

Our Real Estate Appraisal:

MISID:

Appraised Value: $670000.00

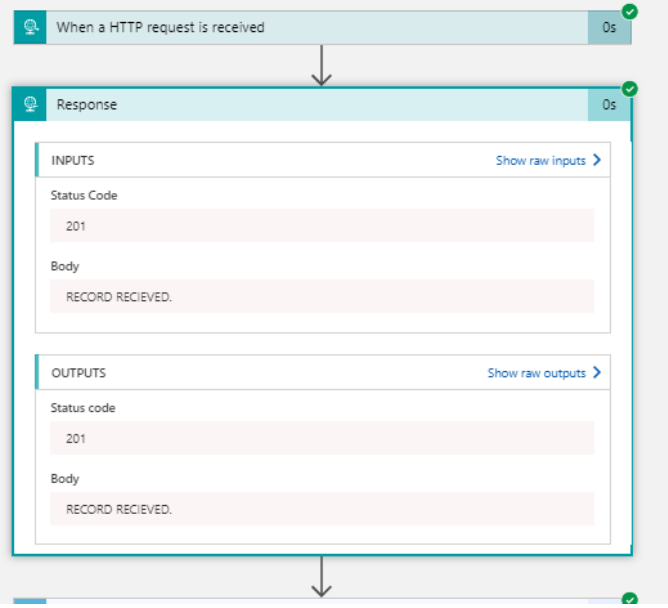
More frontend response from logic app. JSON shown for testing

["name":"Mike Ross","mortid":"20","misid":"1","insuredvalue":"500000","deductible":"5000","code":"1011","appraisedvalue":"670000"]

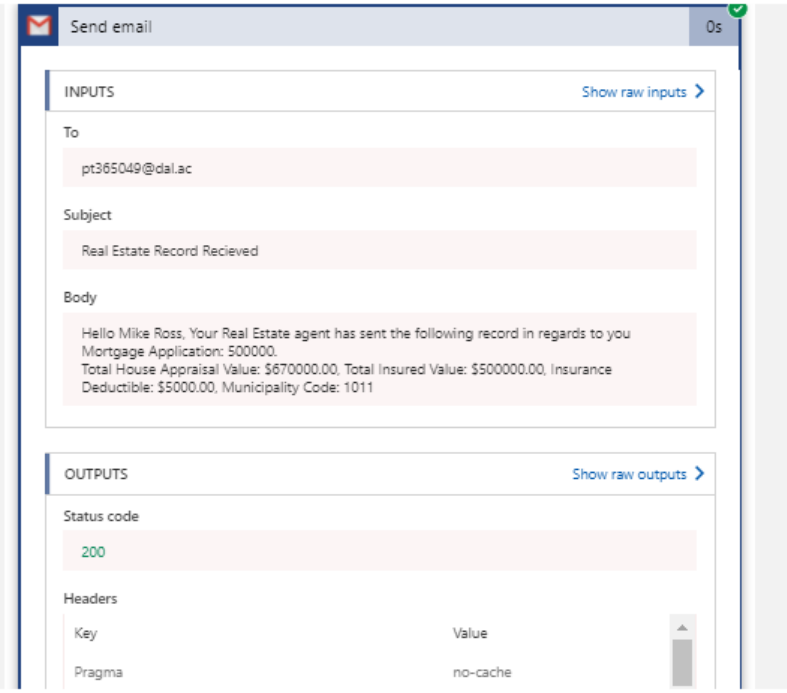
This record was sent to MBR:

Here is MBR response: RECORD RECIEVED..

Backend sending the response.



Backend sending an email to client with the compiled records



Received Email (from another test)

Real Estate Record Recieved

P patrickrobertwalter@gmail.com Reply all

Today, 8:47 AM

Patrick Walter

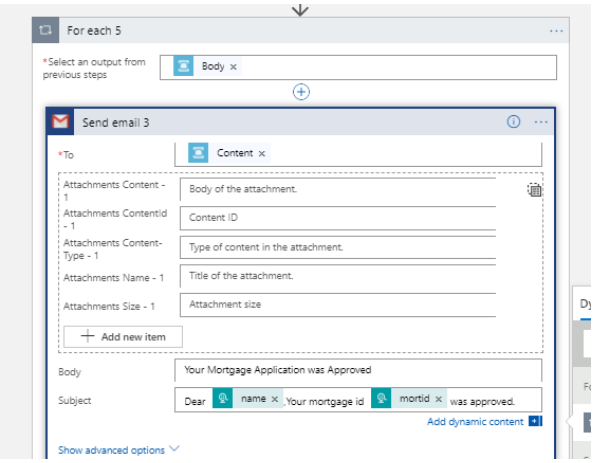
Hello Pat, Your Real Estate agent has sent the following record in regards to you Mortgage Application: 1. Total House Appraisal Value: $540000.00, Total Insured Value: $300000.00, Insurance Deductible: $3000.00, Municipality Code: 101

patrickrobertwalter@gmail.com
Deaar Mike Ross,Your mortgage id 24 was approved. 10:15 AM
Your Mortgage Application was Approved

patrickrobertwalter@gmail.com
Real Estate Record Recieved 10:15 AM
Hello Mike Ross, Your Real Estate agent has sent the following record in regards to you Mor...

patrickrobertwalter@gmail.com
Mortgage Application Received 10:13 AM
Your mortgage application was recieved. Your mortgage application id is 24

Backend view



Workflow Design

Workflow plan and overview

My planned workflow went as follows:

Applicant visits the mortgage brokers' website and fills out the application form information, including his email.

MRBcreate.php Uses an SQL insert command in put this record from the applicant into the projectmbr table. It also takes the email address and mortid, puts it in a JSON, and sends it via HTTP POST using PHP Curl the to *appreply* logic app. The appreply logic app parses the JSON and sends a email via gmail to the applicant.

The *appreply* logic app then sends the email address through the Azure Service bus in a message in a queue called email.

Applicant then visits the Employer Homepage.

readEmployer.php takes the applicants user name and password and authenticates it using a JWT Token. This token generated is not used to continue a safe and secure transmission. It simply generates the JWT and doesn't use it. It then takes the employees id number and check for it in the projectemp table as a valid employee_id.

readEmployer.php takes the record from the projectemp for that employee_id, converts it to a JSON then sends it through an HTTP Post using PHP Curl to *emp* Azure Logic App.

The *emp* logic app then sends a status message through Azure Service Bus queue named status, this was to indicate the emp has send the record. Used later to tell if all records were received. For some reason this second message queue would not function properly.

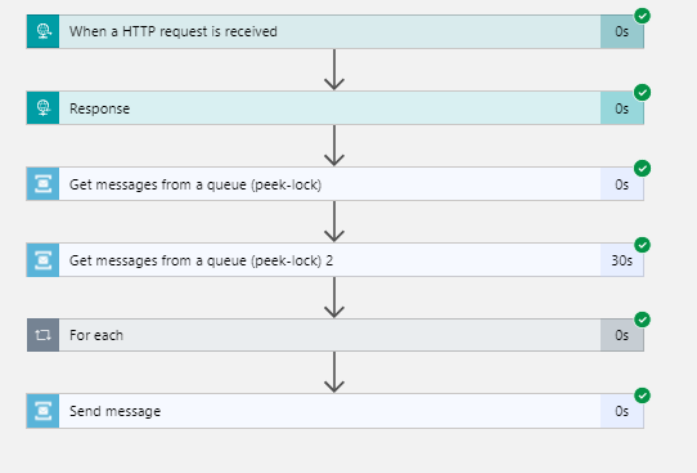
The applicant visits the Real Estate Agents homepage to submit an appraisal. The Real Estate page collects records from 3 database tables through a sequence of SQL Select statements send to the database on the AWS RDS. It collects all 3 records and compiles them together. It shows the applicant this compiled appraisal record, and sends it to the *mortgage* logic app.

The *mortgage* logic app parses this record. It gets the email address from the email Service Bus Queue and sends a detailed email to the applicant about the record it received from the Real Estate Agent.

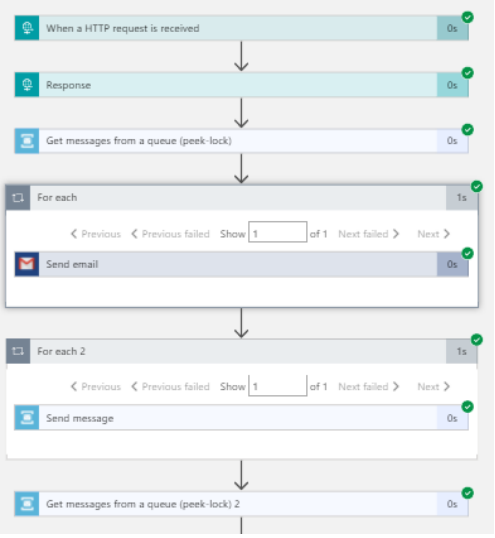
It next looks at the status queue in the Service Bus. Again this part doesn't seem to work. It should check if the emp logic app sent a status message through this queue. And if it has to send an approval email.

Instead the *mortgage* logic app simply sends the approval email based only on the one record it receives.

Employer logic app



Mortgage logic app



Get the record in a JSON from PHP Curl HTTP POST

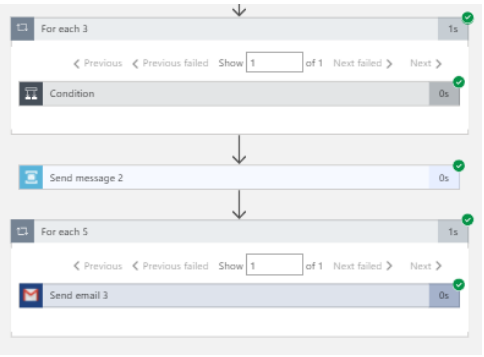
Send a response that is displayed on the front end

Get the email address sent from appreply

Parse the JSON and send a detailed email about the record

Send the email address back in the email queue

Get the status sent from emp logic app (this part doesn't work)



The condition checks if the status is sent from emp indicating that it got a record too.

If it did it sends a approval email.

If not it sends a status to emp to say it go the record (this doesn't work)

Since the above control flow works. I just send an approval based on the record using the email taken from the email queue.

Reference Material

https://stackoverflow.com/questions/6213509/send-json-post-using-php

Used code from this to send records and email addresses through CuRL (HTTP Request) see code for where it was used it is marked in comments.

https://stackoverflow.com/questions/6400300/https-and-ssl3-get-server-certificatecertificate-verify-failed-ca-is-ok

This one line of code : curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false); which fixed a problem I spent 3 days trying to fix. The joys of PHP.

http://php.net/manual/

Spent lots of time here learning PHP.

https://github.com/Simsso/PHP-Simple-JWT-Auth

I used the JWT authentication from this project.