# Project Writeup

## Background

FlightSurety is a sample application project for Udacity's Blockchain course.

It allows user to purchase flight insurance from a registered airline and get paid at a rate of 1.5 if the flight is delay due to airline fault.

FlightSurety is a collaboration between multiple airline, in order to be qualified as one, an airline must get the invitation from registered airlines, and fund at least 10 ether.

## Component

It consists of 4 components

* Smart application contract for app logic and oracles code
* Smart data contract for data persistence
* Client application for interacting with the user and airline
* Server application for answering why the flight is delay

## Technical specification

### Version

Truffle v5.1.21 (core: 5.1.21)

Solidity - ^0.4.24 (solc-js)

Node v9.4.0

Web3.js v1.2.1

### Setup development environment

Code: <https://github.com/ymlai87416/blockchain-nanodegree-project4>

#### Install

This repository contains Smart Contract code in Solidity (using Truffle), tests (also using Truffle), dApp scaffolding (using HTML, CSS and JS) and server app scaffolding.

To install, download or clone the repo, then:

npm install

truffle compile

#### Develop Client

Ganache is required to click start the project.

To run Ganache

ganache-cli -m "candy maple cake sugar pudding cream honey rich smooth crumble sweet treat" -a 50 -l 9999999 -q

To run truffle tests:

truffle test ./test/flightSurety.js

truffle test ./test/oracles.js

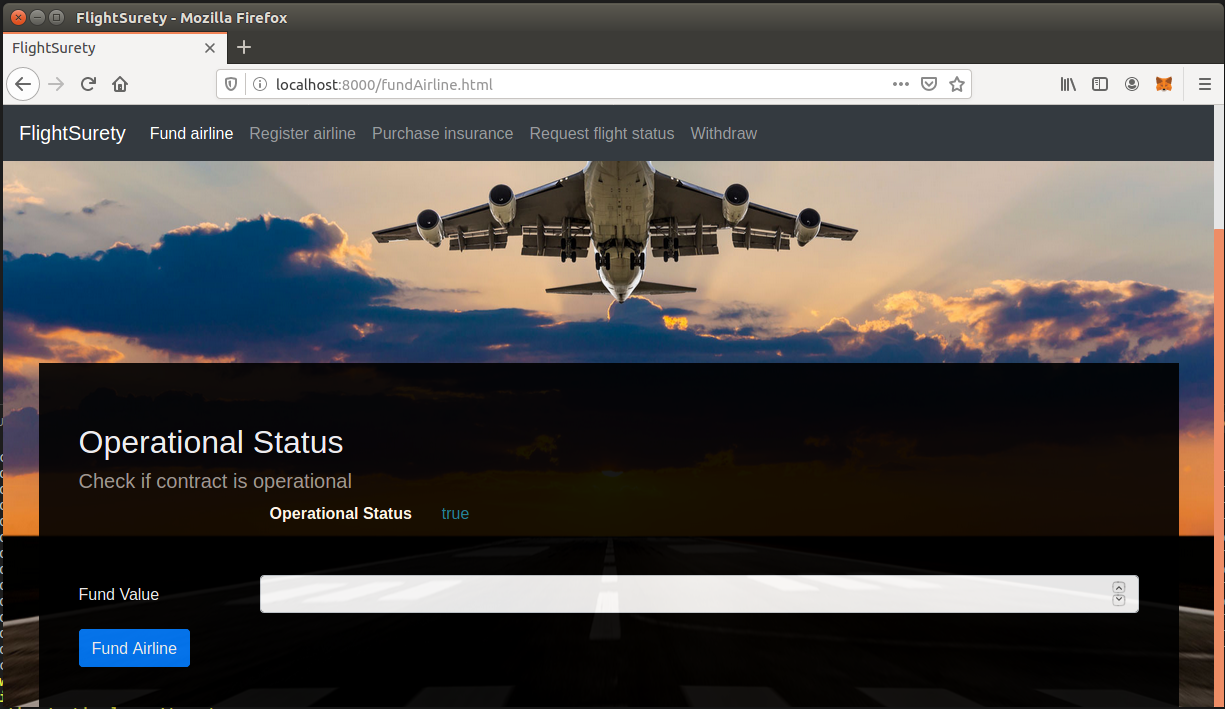
To use the dapp:

truffle migrate --reset --network development

npm run dapp

To view dapp:

http://localhost:8000



There are 5 pages, and each provides a function.

|  |  |
| --- | --- |
| **Page** | **Function** |
| Fund airline | Allow registered airline to provide funding |
| Register airline | Allow registered airline to invite another airline to join. |
| Purchase insurance | Allow passenger to buy insurance |
| Request flight status | Allow anyone to request flight status |
| Withdraw | Allow passenger or airline to get their ETH stored in the contract. |

#### Server

npm run server

truffle test ./test/oracles.js

#### Deploy

To build dapp for prod: npm run dapp:prod

Deploy the contents of the ./dapp folder

## Requirement

1. **Separation of Concerns, Operational Control and “Fail Fast”**

**1.1 Smart contract separation:**

1) FlightSuretyData.sol for data persistence  
2) FlightSuretyApp.sol for app logic and oracles code

**1.2 Dapp Created and Used for Contract Calls**

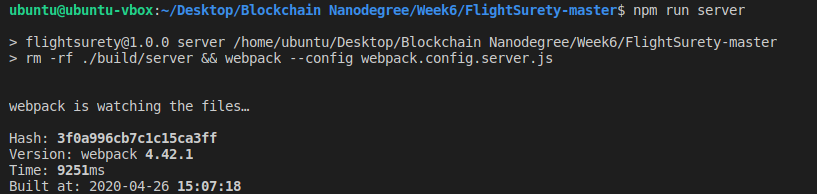
The Dapp is capable to do the following:

|  |  |
| --- | --- |
| Allow airline to provide funding | <http://localhost:8000/fundAirline.html> |
| Allow airline to register another airline | <http://localhost:8000/regAirline.html> |
| Allow passenger to purchase insurance | <http://localhost:8000/buyInsurance.html> |
| Allow anyone to request flight status | <http://localhost:8000/flightStatus.html> |
| Allow passenger or airline to withdraw credit. | <http://localhost:8000/withdraw.html> |

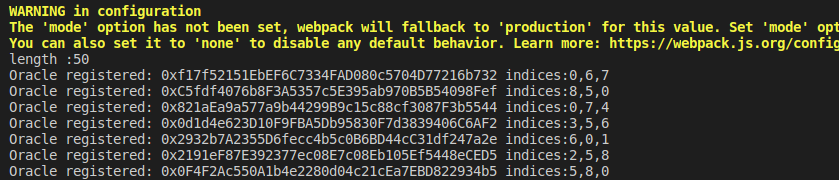
**1.3 Oracle Server Application**

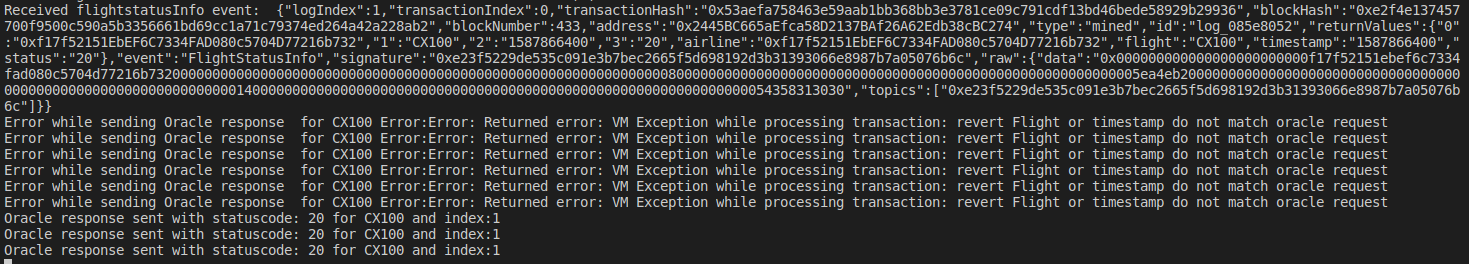
The oracle server is written to simulate a group of 30 oracles.

To start the server



Register oracle



Response to fetch flight status request

**1.4 Operational status control is implemented in contracts**

The operation status control is implemented to allow contract owner to stop people from buying insurance, new airline registration and withdraw credit.

* 1. **Fail Fast Contract**

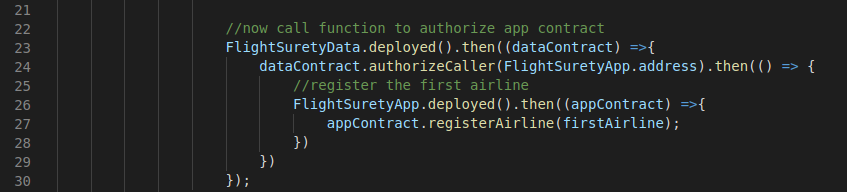
Contract functions “fail fast” by having a majority of “require()” calls at the beginning of function body.

1. **Airline**

**2.1 Airline Contract Initialization**

The first airline is registered by contract owner when contract is deployed.

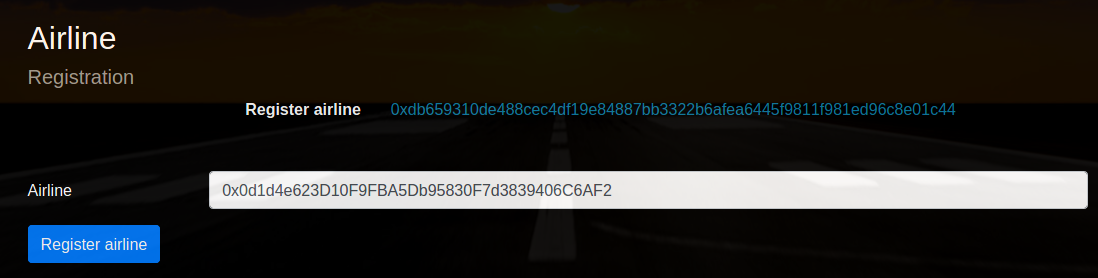
In file 2\_deploy\_contract.js line 26-28, first airline of address 0xf17f52151EbEF6C7334FAD080c5704D77216b732 is registered.



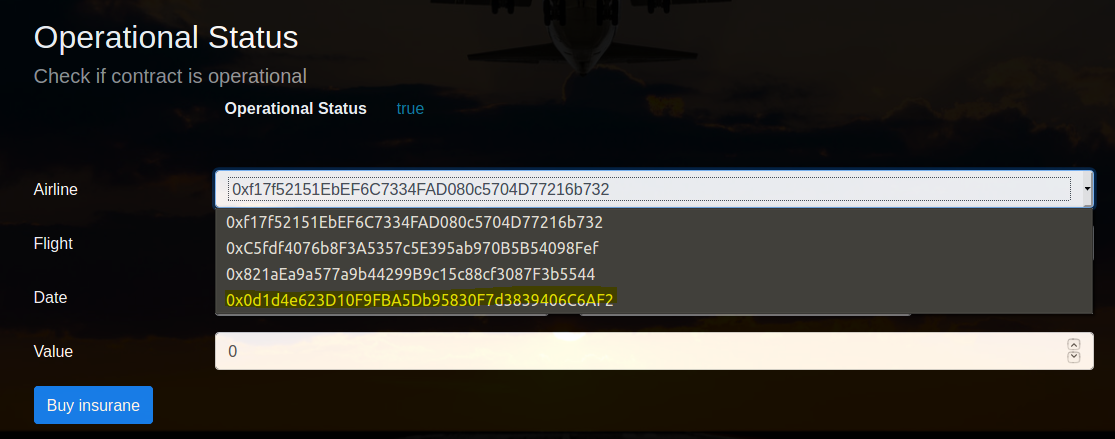
**2.2 – 2.3 Multiparty Consensus**

Only existing airline may register a new airline until there are at least 4 airlines registered

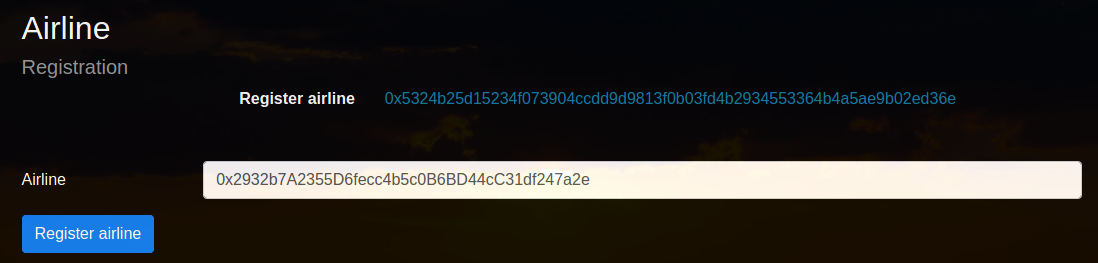
Registering the 4th airline 0x0d1d4e623D10F9FBA5Db95830F7d3839406C6AF2 by first airline.



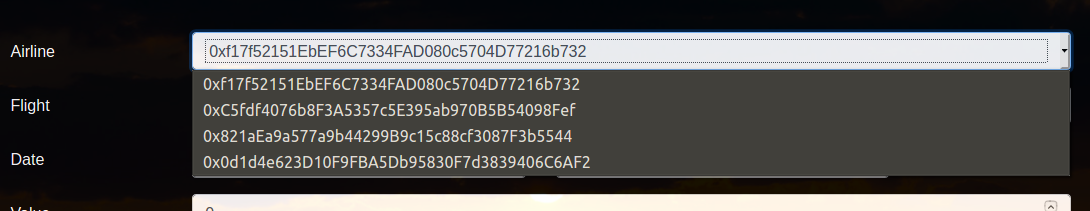
To confirm the airline is registered, go to “Purchase insurance”



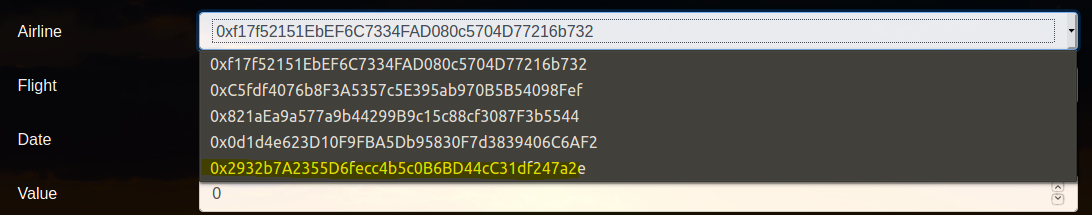
Registration of 5th and subsequent airlines requires multi-party consensus of 50% of registered airline



Even the 5th airline is registered by the 1st airline, it is not shown in the airline selection list.



Later, 2nd airline registered airline 5th airline. Now it is available for selection in the “Purchase insurance” page.

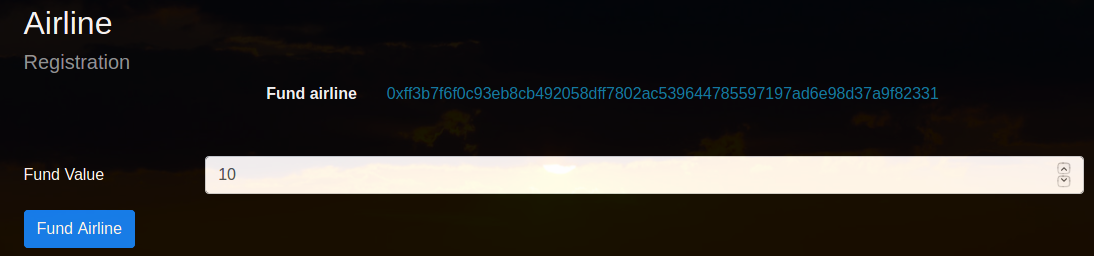


**2.4 Airline Ante**

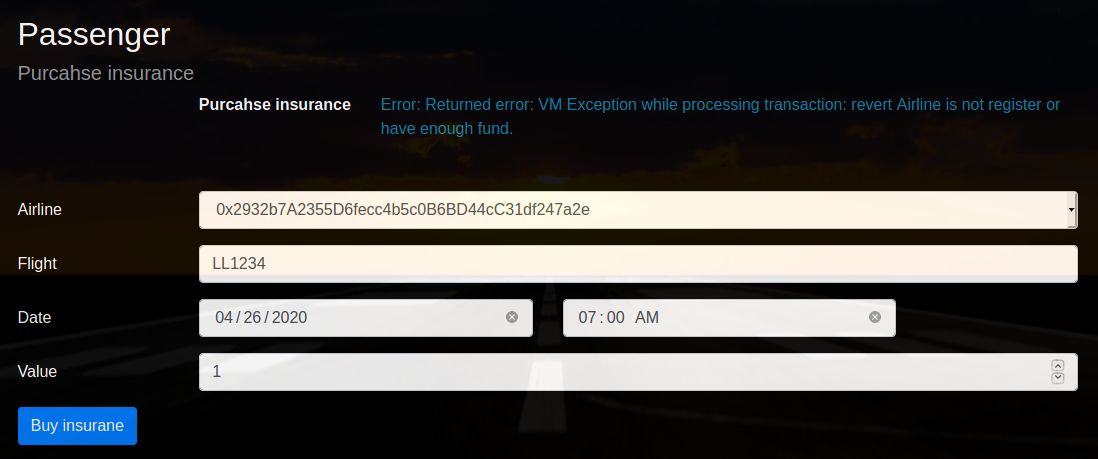
Airline can be registered, but does not participate in the contract until it submits funding of 10 ether

Using “Fund airline” function, an airline can submit their funding.

Below show airline 0xf17f52151EbEF6C7334FAD080c5704D77216b732 submit 10 ether to the contract.



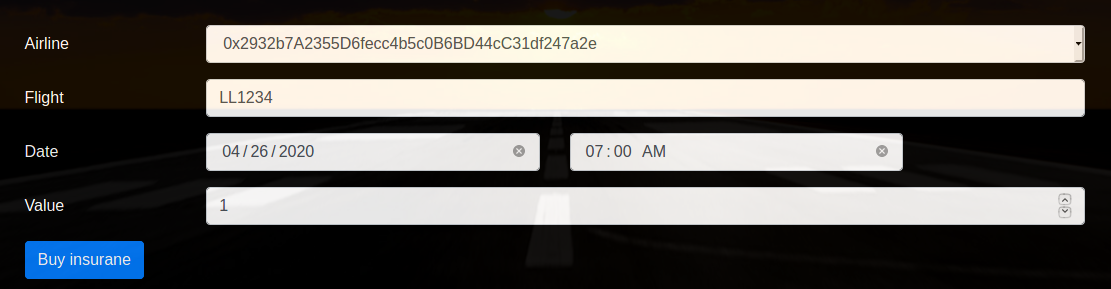
5th airline is registered, but has not yet submit any funding. When a passenger try to buy insurance for flight operating by 5th airline, system returns the following error.



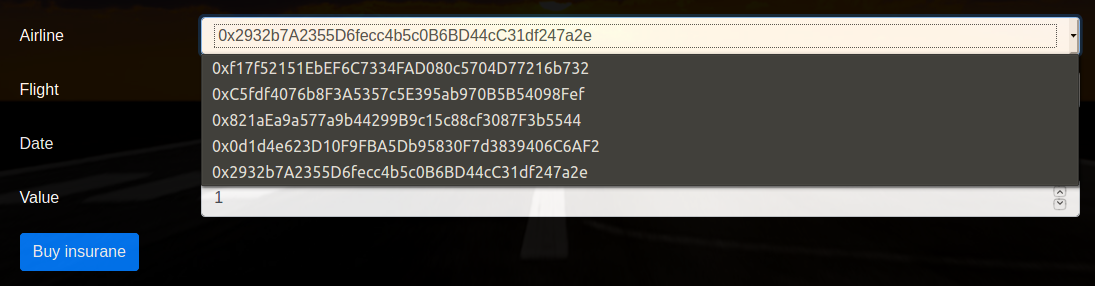
1. **Passenger**

**3.1 Passenger Airline Choice**

Passenger can go to the page “Purchase insurance”



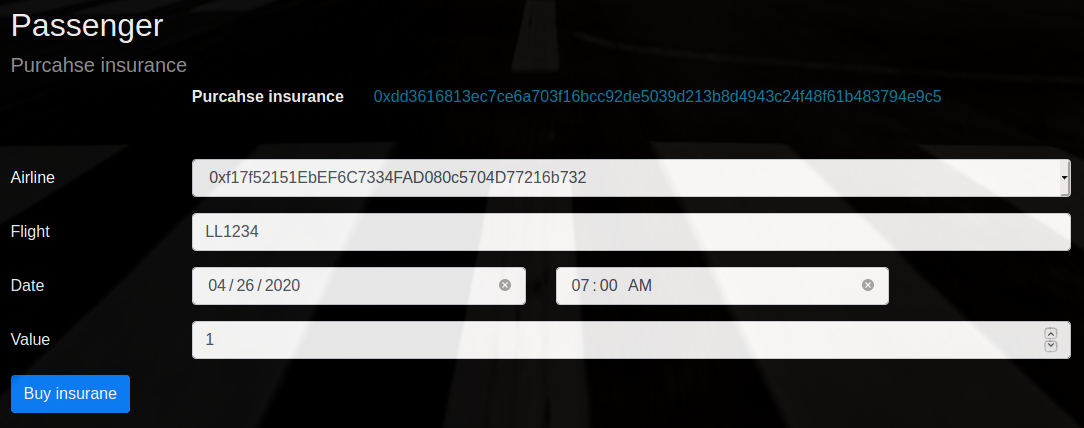
Passenger can choose from a list of airlines.



Passenger enters the flight information such as flight number and time of departure. He also decides how much money to put into this insurance project.

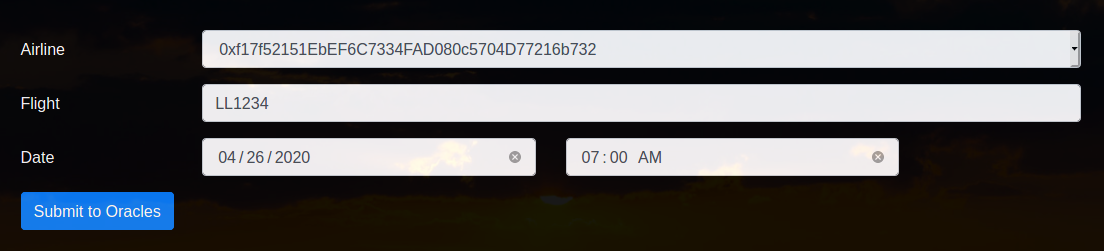
**3.2 Passenger Payment**

Passenger 0x2191eF87E392377ec08E7c08Eb105Ef5448eCED5 purchase an insurance worth 1 eth for the flight LL1234 departed at time 26th April 2020 7a.m. The transaction was successful.



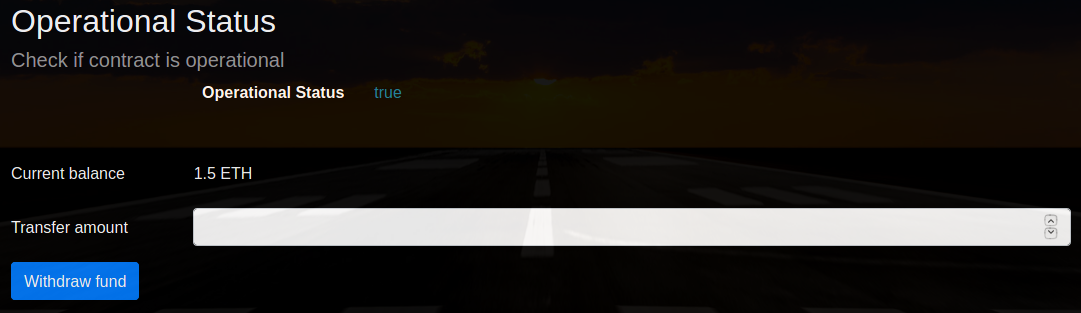
**3.3 Passenger Repayment**

After the flight is landed, anyone can trigger “**Request flight status**” to allow the contract to get the flight status. If the flight is delayed and it is due to the airline fault, passenger receives credit of 1.5X the amount they paid

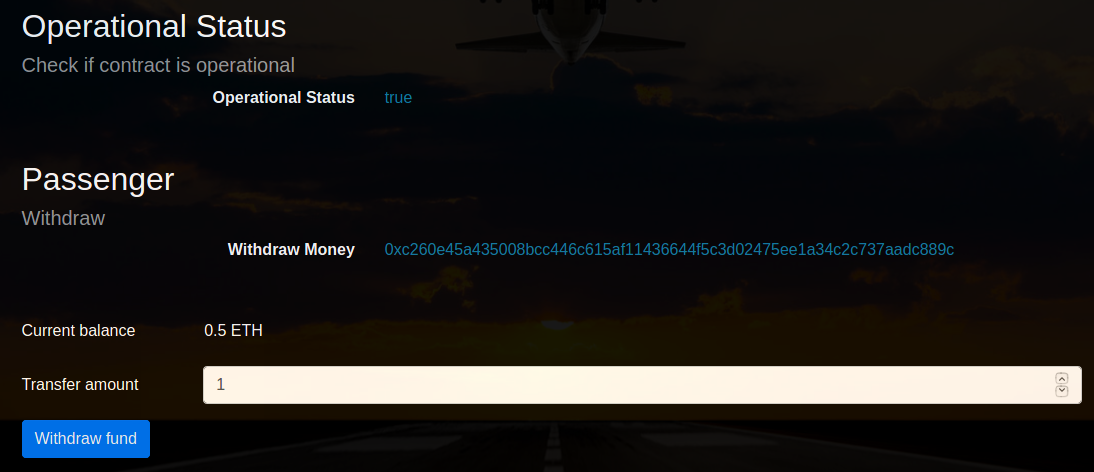


* 1. **Passenger Withdraw**

The oracles replied that the flight LL1234 at departed at time 26th April 2020 7a.m. is delayed due to airline fault. Now passenger 0x2191eF87E392377ec08E7c08Eb105Ef5448eCED5 can go to “**Withdraw**” page to check his credit balance. 1.5 ETH is now available for him to retrieve.



The passenger can enter the amount of ETH he wants to retrieve and click “Withdraw fund” and the remaining balance is updated to 0.5 ETH



* 1. **Insurance Payouts**

Insurance payouts are not sent directly to passenger’s wallet

**4 Oracle (Server App)**

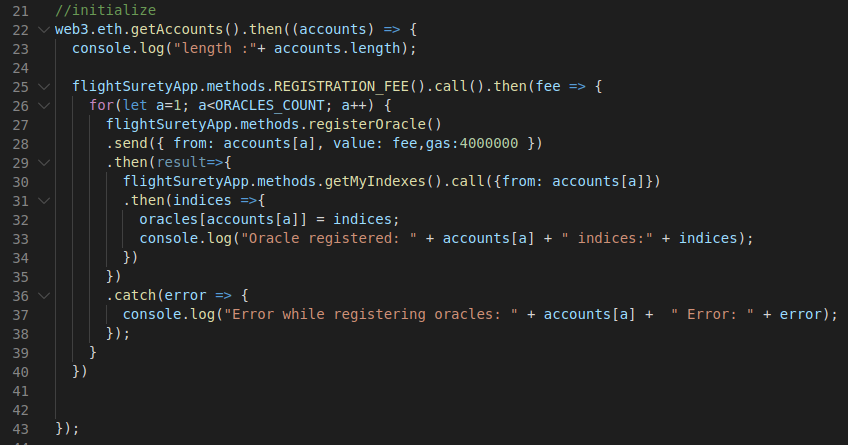
**4.1 Functioning Oracle**

Oracle functionality is implemented in the server app. The logic is at ROOT/src/server/server.js

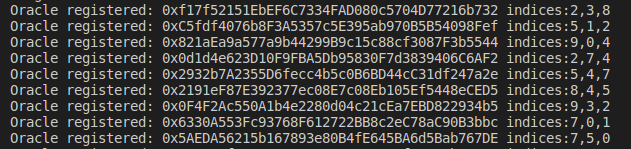
**4.2 Oracle Initialization**

Upon startup, 20+ oracles are registered and their assigned indexes are persisted in memory

Upon start the server register 30 oracles. Below is the code fragment showing the server registers 30 oracles.

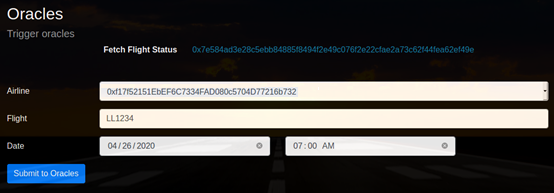


When server starts, server log showing oracles are registered.



**4.3 Oracle Update**

Client Dapp can trigger a flight status request using page “**Request flight status**”.



Server receives the event, and submit oracle responses to the contract. The server log below shows the server is handling the **OracleRequest** event

|  |
| --- |
| Error while sending Oracle response for LL1234 Error:Error: Returned error: VM Exception while processing transaction: revert Index does not match oracle request  Error while sending Oracle response for LL1234 Error:Error: Returned error: VM Exception while processing transaction: revert Flight or timestamp do not match oracle request  Error while sending Oracle response for LL1234 Error:Error: Returned error: VM Exception while processing transaction: revert Flight or timestamp do not match oracle request  Error while sending Oracle response for LL1234 Error:Error: Returned error: VM Exception while processing transaction: revert Flight or timestamp do not match oracle request  Oracle response sent with statuscode: 20 for LL1234 and index:7  Oracle response sent with statuscode: 20 for LL1234 and index:7 |

The server log below shows that it gets the **FlightStatusInfo** event, and display it to the console. This event shows that Flight is a delayed due to airline fault.

Flight information:

Airline: 0xf17f52151EbEF6C7334FAD080c5704D77216b732

Flight number: LL1234

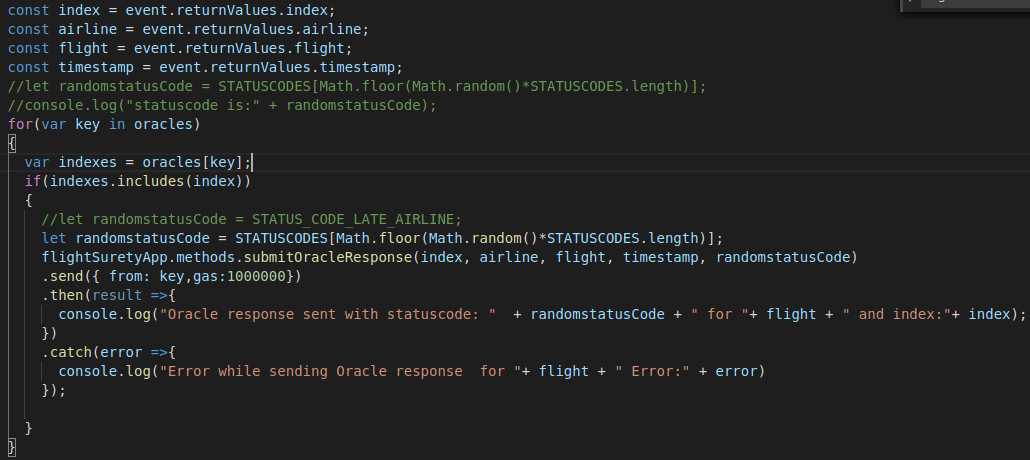
Departure time: 1587855600 => Sunday, April 26, 2020 7:00:00 AM

|  |
| --- |
| Received flightstatusInfo event: {"logIndex":1,"transactionIndex":0,"transactionHash":"0xde180da7e366108339682d4b04b5b597dfd7e2f374f2d6238e53c1e767861bcf","blockHash":"0xc6907788d754bced57cec20853a8e0fdf60d7c60bd64c85d468c0382707ce4ff","blockNumber":646,"address":"0xABa7902442c5739c6f0c182691d48D63d06A212E","type":"mined","id":"log\_ac15cd9b","returnValues":{"0":"0xf17f52151EbEF6C7334FAD080c5704D77216b732","1":"LL1234","2":"1587855600","3":"20","airline":"0xf17f52151EbEF6C7334FAD080c5704D77216b732","flight":"LL1234","timestamp":"1587855600","status":"20"},"event":"FlightStatusInfo","signature":"0xe23f5229de535c091e3b7bec2665f5d698192d3b31393066e8987b7a05076b6c","raw":{"data":"0x000000000000000000000000f17f52151ebef6c7334fad080c5704d77216b7320000000000000000000000000000000000000000000000000000000000000080000000000000000000000000000000000000000000000000000000005ea4c0f0000000000000000000000000000000000000000000000000000000000000001400000000000000000000000000000000000000000000000000000000000000064c4c31323334","topics":["0xe23f5229de535c091e3b7bec2665f5d698192d3b31393066e8987b7a05076b6c"]}} |

**4.4 Oracle Functionality**

Server will loop through all registered oracles, identify those oracles for which the OracleRequest event applies, and respond by calling into FlightSuretyApp contract with random status code of Unknown (0), On Time (10) or Late Airline (20), Late Weather (30), Late Technical (40), or Late Other (50)

The code fragment in ROOT/src/server/server.js shows that



## Resources

* [How does Ethereum work anyway?](https://medium.com/@preethikasireddy/how-does-ethereum-work-anyway-22d1df506369)
* [BIP39 Mnemonic Generator](https://iancoleman.io/bip39/)
* [Truffle Framework](http://truffleframework.com/)
* [Ganache Local Blockchain](http://truffleframework.com/ganache/)
* [Remix Solidity IDE](https://remix.ethereum.org/)
* [Solidity Language Reference](http://solidity.readthedocs.io/en/v0.4.24/)
* [Ethereum Blockchain Explorer](https://etherscan.io/)
* [Web3Js Reference](https://github.com/ethereum/wiki/wiki/JavaScript-API)