1. **Checking Total fare is consistent or not**

**testMain.sh**

#! /bin/bash

# every test case should begin with these two steps

curl -s http://localhost:8081/reset

curl -s http://localhost:8082/reset

# Run two test scripts in parallel

sh sh1.sh &

sh sh2.sh & wait

# sh1 creates the output file sh1out, which contains fares

# of all rides given in sh1. Similarly, sh2out.

totalFare=0

totalBalance=0

testPassed="YES"

((initBalance=10000\*2))

for i in $(cat sh1out sh2out);

do

totalFare=$(expr $totalFare + $i)

done

#Checks wallet balance for the Customers 201 & 202.

balance=$(curl -s "http://localhost:8082/getBalance?custId=201")

((totalBalance=totalBalance+balance))

balance=$(curl -s "http://localhost:8082/getBalance?custId=202")

((totalBalance=totalBalance+balance))

if [ "$totalBalance" != "$initBalance-$totalFare" ];

then

echo "Inconsistent balance amounts!"

testPassed="NO"

else

echo "Possibly valid balance amounts!"

fi

echo "Test Passing Status: " $testPassed

#echo $totalFare $totalBalance $initBalance

**sh1.sh**

rm -f sh1out

#Step 1 : cab 101 signs in

resp=$(curl -s "http://localhost:8080/signIn?cabId=101&initialPos=100")

if [ "$resp" = "true" ];

then

echo "Cab 101 signed in"

else

echo "Cab 101 could not sign in"

fi

# Step 2: customer 201 requests a cab

rideDetails=$(curl -s "http://localhost:8081/requestRide?custId=201&sourceLoc=110&destinationLoc=200")

rideId=$(echo $rideDetails | cut -d' ' -f 1)

cabId=$(echo $rideDetails | cut -d' ' -f 2)

fare=$(echo $rideDetails | cut -d' ' -f 3)

if [ "$rideId" != "-1" ];

then

echo "Ride by customer 201 started"

else

fare=0

echo "Ride to customer 201 denied"

fi

echo $fare >> sh1out

1. **Checking number of rides and fares of individual customers.**

**Testmain.sh**

# every test case should begin with these two steps

curl -s http://localhost:8081/reset

curl -s http://localhost:8082/reset

# Run two test scripts in parallel

sh sh1.sh & sh sh2.sh & sh sh3.sh & wait

# sh1 creates the output file sh1out, which contains fares

# of all rides given in sh1. Similarly, sh2out and sh3out.

testPassed="YES"

initBalance=10000

cid=201

#Checks balance amount for each customer

for fare in $(cat sh1out sh2out sh3out);

do

balance=$(curl -s "http://localhost:8082/getBalance?custId=$cid")

((expBalance = initBalance - fare))

if [ "$balance" != "$expBalance" ];

then

echo "Invalid balance amount for the customer $cid!"

testPassed="NO"

else

echo "Valid balance amount for the customer $cid!"

fi

((cid=cid+1))

done

numRides=$(curl -s "http://localhost:8080/numRides?cabId=101")

echo $numRides

if [ "$numRides" != "1" ];

then

echo "Invalid Number of rides"

testPassed="NO"

else

echo "Correct number of ride"

fi

echo "Test Passing Status: " $testPassed

**sh1.sh**

rm -f sh1out

#Step 1 : cab 101 signs in

resp=$(curl -s "http://localhost:8080/signIn?cabId=101&initialPos=100")

if [ "$resp" = "true" ];

then

echo "Cab 101 signed in"

else

echo "Cab 101 could not sign in"

fi

# Step 2: customer 201 requests a cab

rideDetails=$(curl -s "http://localhost:8081/requestRide?custId=201&sourceLoc=110&destinationLoc=200")

rideId=$(echo $rideDetails | cut -d' ' -f 1)

cabId=$(echo $rideDetails | cut -d' ' -f 2)

fare=$(echo $rideDetails | cut -d' ' -f 3)

if [ "$rideId" != "-1" ];

then

echo "Ride by customer 201 started"

else

fare=0

echo "Ride to customer 201 denied"

fi

echo $fare >> sh1out

**sh2.sh**

rm -f sh2out

#Step 1 : cab 101 signs in

resp=$(curl -s "http://localhost:8080/signIn?cabId=101&initialPos=100")

if [ "$resp" = "true" ];

then

echo "Cab 101 signed in"

else

echo "Cab 101 could not sign in"

fi

# Step 2: customer 202 requests a cab

rideDetails=$(curl -s "http://localhost:8081/requestRide?custId=202&sourceLoc=90&destinationLoc=200")

rideId=$(echo $rideDetails | cut -d' ' -f 1)

cabId=$(echo $rideDetails | cut -d' ' -f 2)

fare=$(echo $rideDetails | cut -d' ' -f 3)

if [ "$rideId" != "-1" ];

then

echo "Ride by customer 202 started"

else

fare=0

echo "Ride to customer 202 denied"

fi

echo $fare >> sh2out

**sh3.sh**

rm -f sh3out

#Step 1 : cab 101 signs in

resp=$(curl -s "http://localhost:8080/signIn?cabId=101&initialPos=100")

if [ "$resp" = "true" ];

then

echo "Cab 101 signed in"

else

echo "Cab 101 could not sign in"

fi

# Step 2: customer 203 requests a cab

rideDetails=$(curl -s "http://localhost:8081/requestRide?custId=203&sourceLoc=190&destinationLoc=200")

rideId=$(echo $rideDetails | cut -d' ' -f 1)

cabId=$(echo $rideDetails | cut -d' ' -f 2)

fare=$(echo $rideDetails | cut -d' ' -f 3)

if [ "$rideId" != "-1" ];

then

echo "Ride by customer 203 started"

else

fare=0

echo "Ride to customer 203 denied"

fi

echo $fare >> sh3out

1. **Checking number of rides after a series of ride requests in loop and rideEnd request in the mid of the loop.**

**testMain.sh**

#! /bin/bash

# Color

YELLOW='\033[0;33m'

# every test case should begin with these two steps

curl -s http://localhost:8081/reset

curl -s http://localhost:8082/reset

# Run two test scripts in parallel

bash sh1.sh &

bash sh2.sh & wait

# sh1 creates the output file sh1out, which contains fares

# of all rides given in sh1. Similarly, sh2out.

testPassed="YES"

totalNoOfRides=0

for num in $(cat sh1out sh2out);

do

((totalNoOfRides=totalNoOfRides+num))

done

numRides=$(curl -s "http://localhost:8080/numRides?cabId=101")

echo $numRides

if [ "$numRides" != "$totalNoOfRides" ];

then

echo "Invalid Number of rides"

testPassed="no"

else

echo "Correct number of rides"

fi

echo -e "${YELLOW}Test Passing Status: " $testPassed

**sh1.sh**

#! /bin/sh

rm -f sh1out

# Color

RED='\033[0;31m'

GREEN='\033[0;32m'

YELLOW='\033[0;33m'

NC='\033[0m' # No Color

noOfIteration=10

onGoingRideId=-1

noOfRide=0

#Step 1 : cab 101 signs in

resp=$(curl -s "http://localhost:8080/signIn?cabId=101&initialPos=100")

if [ "$resp" = "true" ];

then

echo -e "${GREEN}1: Cab 101 signed in${NC}"

else

echo "1: Cab 101 could not sign in"

fi

# Step 2: customer 201 requests a cab

for ((i=1;i<=noOfIteration;i++));

do

rideDetails=$(curl -s "http://localhost:8081/requestRide?custId=201&sourceLoc=100&destinationLoc=105")

rideId=$(echo $rideDetails | cut -d' ' -f 1)

if [ "$rideId" != "-1" ];

then

onGoingRideId=$rideId

((noOfRide=noOfRide+1))

echo -e "${GREEN}sh1: Request $i >> Ride by customer 201 started${NC}"

else

echo "sh1: Request $i >> Ride to customer 201 denied"

fi

#Endride in between

((mid=noOfIteration/2))

if [ "$onGoingRideId" != "-1" ] && [ $i == $mid ];

then

resp=$(curl -s "http://localhost:8080/rideEnded?cabId=101&rideId=$onGoingRideId")

if [ "$resp" = "true" ];

then

echo -e "${YELLOW}1: " $onGoingRideId " has ended" at $i"th iteration ${NC}"

else

echo -e "${RED}1: Could not end${NC}" $onGoingRideId

testPassed="no"

fi

fi

done

echo $noOfRide >> sh1out

**sh2.sh**

#! /bin/sh

rm -f sh2out

# Color

RED='\033[0;31m'

GREEN='\033[0;32m'

YELLOW='\033[0;33m'

NC='\033[0m' # No Color

noOfIteration=10

onGoingRideId=-1

noOfRide=0

#Step 1 : cab 101 signs in

resp=$(curl -s "http://localhost:8080/signIn?cabId=101&initialPos=110")

if [ "$resp" = "true" ];

then

echo -e "${GREEN}2: Cab 101 signed in${NC}"

else

echo "Cab 101 could not sign in"

fi

# Step 2: customer 202 requests a cab

for ((i=1;i<=noOfIteration;i++));

do

rideDetails=$(curl -s "http://localhost:8081/requestRide?custId=202&sourceLoc=110&destinationLoc=105")

rideId=$(echo $rideDetails | cut -d' ' -f 1)

if [ "$rideId" != "-1" ];

then

onGoingRideId=$rideId

((noOfRide=noOfRide+1))

echo -e "${GREEN}sh2: Request $i >> Ride by customer 202 started ${NC}"

else

echo "sh2: Request $i >> Ride to customer 202 denied"

fi

#Endride in between

((mid=noOfIteration/2))

if [ "$onGoingRideId" != "-1" ] && [ $i == $mid ];

then

resp=$(curl -s "http://localhost:8080/rideEnded?cabId=101&rideId=$onGoingRideId")

if [ "$resp" = "true" ];

then

echo -e "${YELLOW}2: " $onGoingRideId " has ended" at $i"th iteration${NC}"

else

echo -e "${RED}2: Could not end" $onGoingRideId "${NC}"

testPassed="no"

fi

fi

done

echo $noOfRide >> sh2out

1. **Checking number of rides after a series of ride requests in loop and rideEnd request in the next iteration of the loop.**

**testMain.sh**

#! /bin/bash

# Color

# every test case should begin with these two steps

curl -s http://localhost:8081/reset

curl -s http://localhost:8082/reset

# Run two test scripts in parallel

bash sh1.sh &

bash sh2.sh & wait

# sh1 creates the output file sh1out, which contains fares

# of all rides given in sh1. Similarly, sh2out.

YELLOW='\033[0;33m'

testPassed="YES"

totalNoOfRides=0

for num in $(cat sh1out sh2out);

do

((totalNoOfRides=totalNoOfRides+num))

done

numRides=$(curl -s "http://localhost:8080/numRides?cabId=101")

echo $numRides

if [ "$numRides" != "$totalNoOfRides" ];

then

echo "Invalid Number of rides"

testPassed="no"

else

echo "Correct number of rides"

fi

echo -e "${YELLOW}Test Passing Status: " $testPassed

**sh1.sh**

#! /bin/sh

rm -f sh1out

# Color

RED='\033[0;31m'

GREEN='\033[0;32m'

YELLOW='\033[0;33m'

NC='\033[0m' # No Color

noOfIteration=10

onGoingRideId=-1

noOfRide=0

#Step 1 : cab 101 signs in

resp=$(curl -s "http://localhost:8080/signIn?cabId=101&initialPos=100")

if [ "$resp" = "true" ];

then

echo -e "${GREEN}1: Cab 101 signed in${NC}"

else

echo "1: Cab 101 could not sign in"

fi

# Step 2: customer 201 requests a cab

for ((i=1;i<=noOfIteration;i++));

do

#End previous ride

if [ "$onGoingRideId" != "-1" ] && [ $i != "1" ];

then

resp=$(curl -s "http://localhost:8080/rideEnded?cabId=101&rideId=$onGoingRideId")

if [ "$resp" = "true" ];

then

echo -e "${GREEN}1: " $onGoingRideId " has ended" at $i"th iteration ${NC}"

onGoingRideId=-1

else

echo -e "${RED}1: Could not end${NC}" $onGoingRideId

testPassed="no"

fi

fi

rideDetails=$(curl -s "http://localhost:8081/requestRide?custId=201&sourceLoc=100&destinationLoc=105")

rideId=$(echo $rideDetails | cut -d' ' -f 1)

if [ "$rideId" != "-1" ];

then

onGoingRideId=$rideId

((noOfRide=noOfRide+1))

echo -e "${GREEN}sh1: Request $i >> Ride by customer 201 started${NC}"

else

echo "sh1: Request $i >> Ride to customer 201 denied"

fi

done

echo $noOfRide >> sh1out

**sh2.sh**

#! /bin/sh

rm -f sh2out

# Color

RED='\033[0;31m'

GREEN='\033[0;32m'

YELLOW='\033[0;33m'

NC='\033[0m' # No Color

noOfIteration=10

onGoingRideId=-1

noOfRide=0

#Step 1 : cab 101 signs in

resp=$(curl -s "http://localhost:8080/signIn?cabId=101&initialPos=110")

if [ "$resp" = "true" ];

then

echo -e "${GREEN}2: Cab 101 signed in${NC}"

else

echo "Cab 101 could not sign in"

fi

# Step 2: customer 202 requests a cab

for ((i=1;i<=noOfIteration;i++));

do

#End previous ride

echo -e "${RED}>>" $onGoingRideId "${NC}"

if [ "$onGoingRideId" != "-1" ] && [ $i != "1" ];

then

resp=$(curl -s "http://localhost:8080/rideEnded?cabId=101&rideId=$onGoingRideId")

if [ "$resp" = "true" ];

then

echo -e "${YELLOW}2: " $onGoingRideId " has ended" at $i"th iteration${NC}"

onGoingRideId=-1

else

echo -e "${RED}2: Could not end" $onGoingRideId "${NC}"

testPassed="no"

fi

fi

rideDetails=$(curl -s "http://localhost:8081/requestRide?custId=202&sourceLoc=110&destinationLoc=105")

rideId=$(echo $rideDetails | cut -d' ' -f 1)

if [ "$rideId" != "-1" ];

then

onGoingRideId=$rideId

((noOfRide=noOfRide+1))

echo -e "${GREEN}sh2: Request $i >> Ride by customer 202 started ${NC}"

else

echo "sh2: Request $i >> Ride to customer 202 denied"

fi

done

echo $noOfRide >> sh2out