

Technical Challenge Instructions:

1. Write a Golang Kafka Publisher that takes a CSV file and published each line as a message to a topic with 4 partitions.
2. Write a Golang Kafka Consumer that uses a consumer group to read off the topic for any message that calculates the sum of all the numbers and output the sum, read off the topic for any messages with English words and sorts the words alphabetically and outputs.
3. Show a high-level design document of the application and how it follows a RESTful design pattern.
4. Use *go lint*, *go vet*, and *go test* for testing the application.
5. Demo of the application taking the provided CSV file.

Solution:

1. The *Golang Kafka Publisher* is a RESTful Go application having a *POST* REST endpoint (*localhost:9090/csv*) that takes a JSON payload having a *location* field that is the location of the CSV file to be retrieved and published as messages to the *testTopic* topic in the Kafka cluster.
2. The *Golang Kafka Consumer* is a Go application that is a part of the *group-id-1* consumer group and consumes messages published to the *testTopic* topic in the Kafka cluster. The consumer application performs a check on the message to determine if the line contains numbers or strings values. It adds the numbers together and prints out the sum to the console if it contains numbers, otherwise it sorts the strings and prints out the sorted values to the console.
3. High-Level Design of the Architecture.

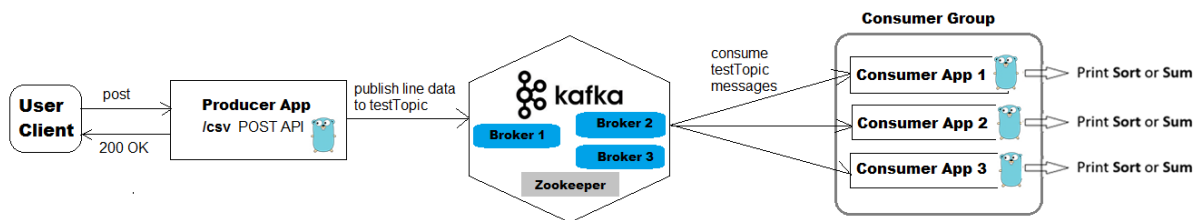


Fig 1: High-level Architecture

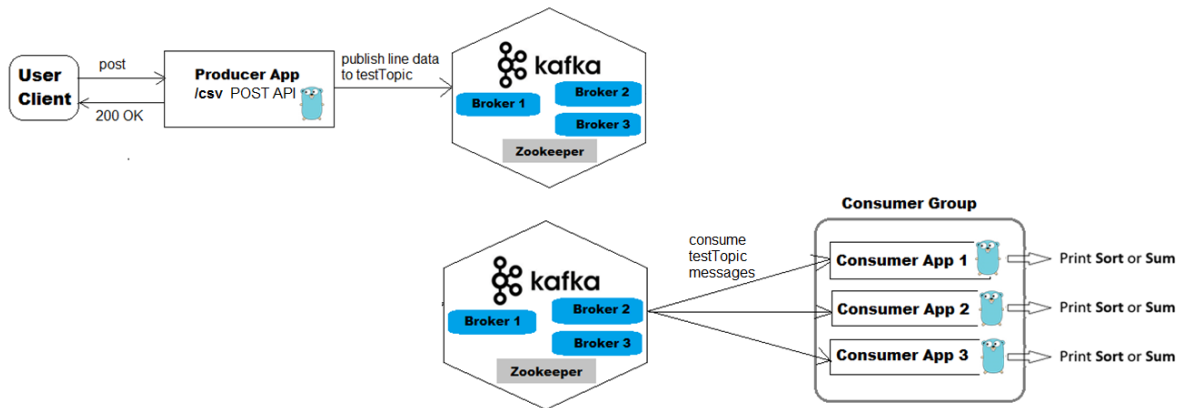


Fig 2: Decoupled High-level Architecture showing separate message publishing and consumption views

- The Kafka cluster consists of 3 brokers.
- A topic called *testTopic* having 4 partitions with a replication factor of 3. It was created using the *kafka-tool* as Fig 7.
- Several other figures are provided below that illustrate how to create and start the Zookeeper and Kafka clusters, the Kafka brokers, the Producer and Consumer applications.
- The Producer POST RESTful endpoint at *localhost:9090/csv* was triggered with Postman client.

- Sample output of the raw CSV data and the console outputs of the processed Consumer application results.
- The development environment was configured to use *go lint* and *go vet* by running the commands *go get -u golang.org/x/lint/golint* and *go tool vet *.go* in the respective project directories.
- The Unit testing file *consumer_test.go* is shown in Fig 15 and the unit test results for testing the Consumer application are then displayed in Fig 16.

Fig 3: Start Zookeeper using the `.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties` command

Fig 4: Start Kafka broker 1 as brokerId=0 using `.\bin\windows\kafka-server-start.bat .\config\server-1.properties` command

```
Administrator: Command Prompt - .\bin\windows\kafka-server-start.bat .\config\server-1.properties
[2018-11-18 09:16:18.478] INFO Replica loaded for partition TestTopicXYZ-0 with initial high watermark 0 <kafka.cluster.Replica>
[2018-11-18 09:16:18.478] INFO Replica loaded for partition TestTopicXYZ-0 with initial high watermark 83 <kafka.cluster.Replica>
[2018-11-18 09:16:18.478] INFO Replica loaded for partition TestTopicXYZ-0 with initial high watermark 0 <kafka.cluster.Replica>
[2018-11-18 09:16:18.478] INFO Replica loaded for partition TestTopicABC-0 with initial high watermark 25 <kafka.cluster.Replica>
[2018-11-18 09:16:18.478] INFO Replica loaded for partition challengeTopic-1 with initial high watermark 0 <kafka.cluster.Replica>
[2018-11-18 09:16:18.494] INFO Replica loaded for partition challengeTopic-1 with initial high watermark 1035 <kafka.cluster.Replica>
[2018-11-18 09:16:18.494] INFO Replica loaded for partition challengeTopic-1 with initial high watermark 0 <kafka.cluster.Replica>
[2018-11-18 09:16:18.527] INFO [ReplicaFetcherManager on broker 1] Removed fetcher for partitions TestTopicXYZ-0,challengeTopic-3,challengeTopic-1,challengeTopic-0,TestTopicXYZ-1,csvtopic-0,challengeTopic-2 <kafka.server.ReplicaFetcherManager>
[2018-11-18 09:16:18.509] INFO [ReplicaFetcher replicaId=1, leaderId=0, fetcherId=0] Starting <kafka.server.ReplicaFetcherThread>
[2018-11-18 09:16:18.527] INFO [ReplicaFetcherManager on broker 1] Added fetcher for partitions List(<challengeTopic-1, initOffset 1035 to broker BrokerEndPoint(0,Elite8300-PC.9092)>), <challengeTopic-2, initOffset 1 to broker BrokerEndPoint(0,Elite8300-PC.9092)>), <TestTopicXYZ-1, initOffset 868 to broker BrokerEndPoint(0,Elite8300-PC.9092)>), <challengeTopic-3, initOffset 2 to broker BrokerEndPoint(0,Elite8300-PC.9092)>), <csvtopic-0, initOffset 1 to broker BrokerEndPoint(0,Elite8300-PC.9092)>), <challengeTopic-0, initOffset 580 to broker BrokerEndPoint(0,Elite8300-PC.9092)>), <TestTopicXYZ-0, initOffset 83 to broker BrokerEndPoint(0,Elite8300-PC.9092)>). <kafka.server.ReplicaFetcherManager>
[2018-11-18 09:16:18.527] INFO [ReplicaAlterLogDirManager on broker 1] Added fetcher for partitions List(<> <kafka.server.ReplicaAlterLogDirManager>)
[2018-11-18 09:16:18.543] INFO [Log partition=csvtopic-0, dir=C:\tmp\kafka-logs-1] Truncating to 1 has no effect as the largest offset in the log is 0 <kafka.log.Log>
[2018-11-18 09:16:18.543] INFO [Log partition=challengeTopic-1, dir=C:\tmp\kafka-logs-1] Truncating to 1035 has no effect as the largest offset in the log is 1034 <kafka.log.Log>
[2018-11-18 09:16:18.543] INFO [Log partition=challengeTopic-0, dir=C:\tmp\kafka-logs-1] Truncating to 580 has no effect as the largest offset in the log is 579 <kafka.log.Log>
[2018-11-18 09:16:18.543] INFO [Log partition=challengeTopic-3, dir=C:\tmp\kafka-logs-1] Truncating to 2 has no effect as the largest offset in the log is 1 <kafka.log.Log>
[2018-11-18 09:16:18.543] INFO [Log partition=TestTopicXYZ-1, dir=C:\tmp\kafka-logs-1] Truncating to 868 has no effect as the largest offset in the log is 867 <kafka.log.Log>
[2018-11-18 09:16:18.543] INFO [Log partition=challengeTopic-2, dir=C:\tmp\kafka-logs-1] Truncating to 1 has no effect as the largest offset in the log is 0 <kafka.log.Log>
[2018-11-18 09:16:18.543] INFO [Log partition=TestTopicXYZ-0, dir=C:\tmp\kafka-logs-1] Truncating to 83 has no effect as the largest offset in the log is 82 <kafka.log.Log>
[2018-11-18 09:16:18.543] INFO [ReplicaFetcherManager on broker 1] Removed fetcher for partitions TestTopicABC-0 <kafka.server.ReplicaFetcherManager>
[2018-11-18 09:16:18.543] INFO [Partition TestTopicABC-0 broker=1] TestTopicABC-0 starts at Leader Epoch 4 from offset 25. Previous Leader Epoch was: -1 <kafka.cluster.Partition>
[2018-11-18 09:16:18.558] INFO [ReplicaAlterLogDirManager on broker 1] Added fetcher for partitions List(<> <kafka.server.ReplicaAlterLogDirManager>)
```

Fig 5: Start Kafka broker 2 as brokerId=1 using `.\bin\windows\kafka-server-start.bat .\config\server-2.properties` command

```
Administrator: Command Prompt - .\bin\windows\kafka-server-start.bat .\config\server-2.properties
[2018-11-18 09:17:15.856] INFO Replica loaded for partition TestTopicXYZ-0 with initial high watermark 0 <kafka.cluster.Replica>
[2018-11-18 09:17:15.856] INFO Replica loaded for partition TestTopicXYZ-0 with initial high watermark 0 <kafka.cluster.Replica>
[2018-11-18 09:17:15.872] INFO Replica loaded for partition csvtopic-2 with initial high watermark 195 <kafka.cluster.Replica>
[2018-11-18 09:17:15.872] INFO Replica loaded for partition csvtopic-2 with initial high watermark 0 <kafka.cluster.Replica>
[2018-11-18 09:17:15.875] INFO Replica loaded for partition challengeTopic-1 with initial high watermark 1035 <kafka.cluster.Replica>
[2018-11-18 09:17:15.875] INFO Replica loaded for partition challengeTopic-1 with initial high watermark 0 <kafka.cluster.Replica>
[2018-11-18 09:17:15.875] INFO [ReplicaFetcherManager on broker 2] Removed fetcher for partitions TestTopicXYZ-0,challengeTopic-3,challengeTopic-1,challengeTopic-0,TestTopicXYZ-1,csvtopic-2,challengeTopic-2 <kafka.server.ReplicaFetcherManager>
[2018-11-18 09:17:15.906] INFO [ReplicaFetcher replicaId=2, leaderId=0, fetcherId=0] Starting <kafka.server.ReplicaFetcherThread>
[2018-11-18 09:17:15.906] INFO [ReplicaFetcherManager on broker 2] Added fetcher for partitions List(<challengeTopic-1, initOffset 1035 to broker BrokerEndPoint(0,Elite8300-PC.9092)>), <csvtopic-2, initOffset 195 to broker BrokerEndPoint(0,Elite8300-PC.9092)>), <challengeTopic-2, initOffset 1 to broker BrokerEndPoint(0,Elite8300-PC.9092)>), <TestTopicXYZ-1, initOffset 868 to broker BrokerEndPoint(0,Elite8300-PC.9092)>), <challengeTopic-0, initOffset 580 to broker BrokerEndPoint(0,Elite8300-PC.9092)>), <TestTopicXYZ-0, initOffset 83 to broker BrokerEndPoint(0,Elite8300-PC.9092)>). <kafka.server.ReplicaFetcherManager>
[2018-11-18 09:17:15.922] INFO [ReplicaAlterLogDirManager on broker 2] Added fetcher for partitions List(<> <kafka.server.ReplicaAlterLogDirManager>)
[2018-11-18 09:17:15.922] INFO [Log partition=challengeTopic-1, dir=C:\tmp\kafka-logs-2] Truncating to 1035 has no effect as the largest offset in the log is 1034 <kafka.log.Log>
[2018-11-18 09:17:15.922] INFO [Log partition=challengeTopic-0, dir=C:\tmp\kafka-logs-2] Truncating to 580 has no effect as the largest offset in the log is 579 <kafka.log.Log>
[2018-11-18 09:17:15.937] INFO [Log partition=csvtopic-2, dir=C:\tmp\kafka-logs-2] Truncating to 195 has no effect as the largest offset in the log is 194 <kafka.log.Log>
[2018-11-18 09:17:15.937] INFO [Log partition=challengeTopic-3, dir=C:\tmp\kafka-logs-2] Truncating to 2 has no effect as the largest offset in the log is 1 <kafka.log.Log>
[2018-11-18 09:17:15.937] INFO [Log partition=TestTopicXYZ-1, dir=C:\tmp\kafka-logs-2] Truncating to 868 has no effect as the largest offset in the log is 867 <kafka.log.Log>
[2018-11-18 09:17:15.937] INFO [Log partition=challengeTopic-2, dir=C:\tmp\kafka-logs-2] Truncating to 1 has no effect as the largest offset in the log is 0 <kafka.log.Log>
[2018-11-18 09:17:15.937] INFO [Log partition=TestTopicXYZ-0, dir=C:\tmp\kafka-logs-2] Truncating to 83 has no effect as the largest offset in the log is 82 <kafka.log.Log>
[2018-11-18 09:17:15.937] INFO [ReplicaFetcherManager on broker 2] Removed fetcher for partitions csvtopic-1 <kafka.server.ReplicaFetcherManager>
[2018-11-18 09:17:15.937] INFO [Partition csvtopic-1 broker=2] csvtopic-1 starts at Leader Epoch 7 from offset 29. Previous Leader Epoch was: -1 <kafka.cluster.Partition>
[2018-11-18 09:17:15.937] INFO Replica loaded for partition csvtopic-1 with initial high watermark 0 <kafka.cluster.Replica>
[2018-11-18 09:17:15.937] INFO [ReplicaAlterLogDirManager on broker 2] Added fetcher for partitions List(<> <kafka.server.ReplicaAlterLogDirManager>)
[2018-11-18 09:17:16.842] INFO [Partition csvtopic-1 broker=2] Expanding ISR from 2 to 2.1 <kafka.cluster.Partition>
```

Fig 6: Start Kafka broker 3 as brokerId=2 using `.\bin\windows\kafka-server-start.bat .\config\server-3.properties` command

```
Administrator: Command Prompt
C:\kafka\kafka_2.12-2.0.0>.bin\windows\kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 3 --partitions 4 --topic testTopic
Created topic "testTopic".
C:\kafka\kafka_2.12-2.0.0>
```

Fig 7: Create a new topic *testTopic* using the `.bin\windows\kafka-topics.bat - -create - -zookeeper localhost:2181 - -replication-factor 3 - -partitions 4 - -topic testTopic` command

```
Administrator: Command Prompt - .bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic testTopic
C:\kafka\kafka_2.12-2.0.0>.bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic testTopic
```

Fig 8: Create a console consumer for the *testTopic* topic using the `.bin\windows\kafka-console-consumer.bat - -bootstrap-server localhost:9092 - -topic testTopic` command

```
Administrator: Command Prompt - .bin\windows\kafka-topics.bat --zookeeper localhost:2181 --describe --topic testTopics
C:\kafka\kafka_2.12-2.0.0>.bin\windows\kafka-topics.bat --zookeeper localhost:2181 --describe --topic testTopics
Topic:testTopics      PartitionCount:4      ReplicationFactor:3      Configs:
Isr: 2,1,0
Topic: testTopics      Partition: 0      Leader: 2      Replicas: 2,1,0
Isr: 0,2,1
Topic: testTopics      Partition: 1      Leader: 0      Replicas: 0,2,1
Isr: 1,0,2
Topic: testTopics      Partition: 2      Leader: 1      Replicas: 1,0,2
Isr: 2,0,1
Topic: testTopics      Partition: 3      Leader: 2      Replicas: 2,0,1
```

Fig 9: View partition distribution across brokers using the `.bin\windows\kafka-topics.bat - -zookeeper localhost:2181 - -describe - -topic testTopics` command

```
1: bash, go, go
Elite8300@Elite8300-PC MINGW64 ~/Documents/GoP...
$ go run main.go
Start receiving from Kafka

Elite8300@Elite8300-PC MINGW64 ~/Documents/GoP...
$ go run main.go
Start receiving from Kafka
```

Fig 10: Starting the 2 consumers in the same consumer group for the *testTopic* topic

```
data.csv - moduletest - Visual Studio Code
File Edit Selection View Go Debug Tasks Help

EXPLORER
main.go rest-api-to-kafka-ms
main.go a-technical-project-consum...
main.go a-technical-project-consum...
main.go a-technical-project-producer
data.csv a-technical-project-producer
main.go kafka-to-mongodb-ms

MODULETEST
msvcrt120.dll
rdkafka.pc
zlib.dll
a-technical-project-consumer-2
librdkafka.dll
librdkafka.pc
main.go
msvcrt120.dll
rdkafka.pc
zlib.dll
a-technical-project-producer
data.csv
librdkafka.dll
librdkafka.pc
main.go
msvcrt120.dll
rdkafka.pc

1 1,2,3,4,5,6
2 7,8,9,3,2,1
3 rain,snow,bag,fly,dump,egg
4 3,5,6,7,8,2
5 8,4,6,7,2,2
6 rtru,aiis,shuss,syuud,didid,auao
7 6,2,1,5,6,7
8 eueyue,dod,shsjhs,hhdhd,sbhsa,dghhf
9 3,5,6,8,9,0
10 rurur,ffof,hhsd,iudif,sgshys,osjd
11 1,2,3,4,5,6
12 7,8,9,3,2,1
13 rain,snow,bag,fly,dump,egg
14 3,5,6,7,8,2
15 8,4,6,7,2,2
16 rtru,aiis,shuss,syuud,didid,auao
17 6,2,1,5,6,7
18 eueyue,dod,shsjhs,hhdhd,sbhsa,dghhf
19 3,5,6,8,9,0
20 rurur,ffof,hhsd,iudif,sgshys,osjd
21 1,2,3,4,5,6
22 7,8,9,3,2,1
23 rain,snow,bag,fly,dump,egg
24 3,5,6,7,8,2
25 8,4,6,7,2,2
26 rtru,aiis,shuss,syuud,didid,auao
27 6,2,1,5,6,7
28 eueyue,dod,shsjhs,hhdhd,sbhsa,dghhf
29 3,5,6,8,9,0
30 rurur,ffof,hhdhd,iudif,sgshys,osjd
```

Fig 11: CSV file to be read by producer on POST request and sent to Kafka cluster *testTopic* topic

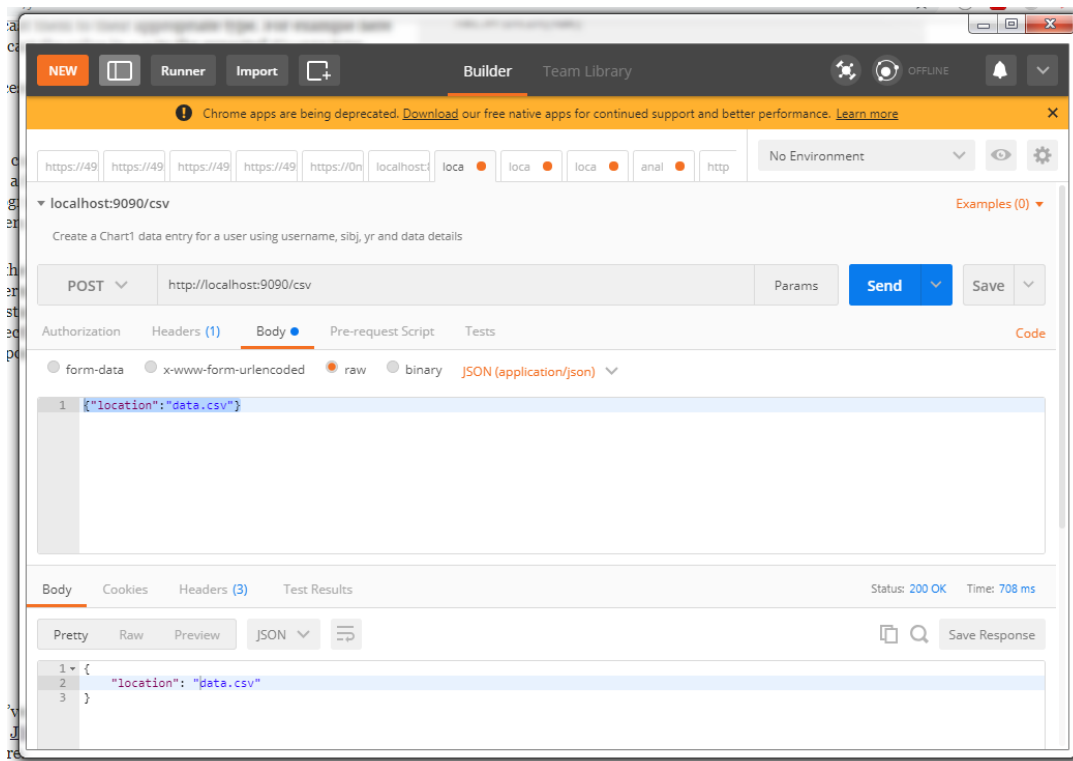


Fig 12: Producer POST API showing CSV file location to be retrieved and sent to Kafka broker as messages

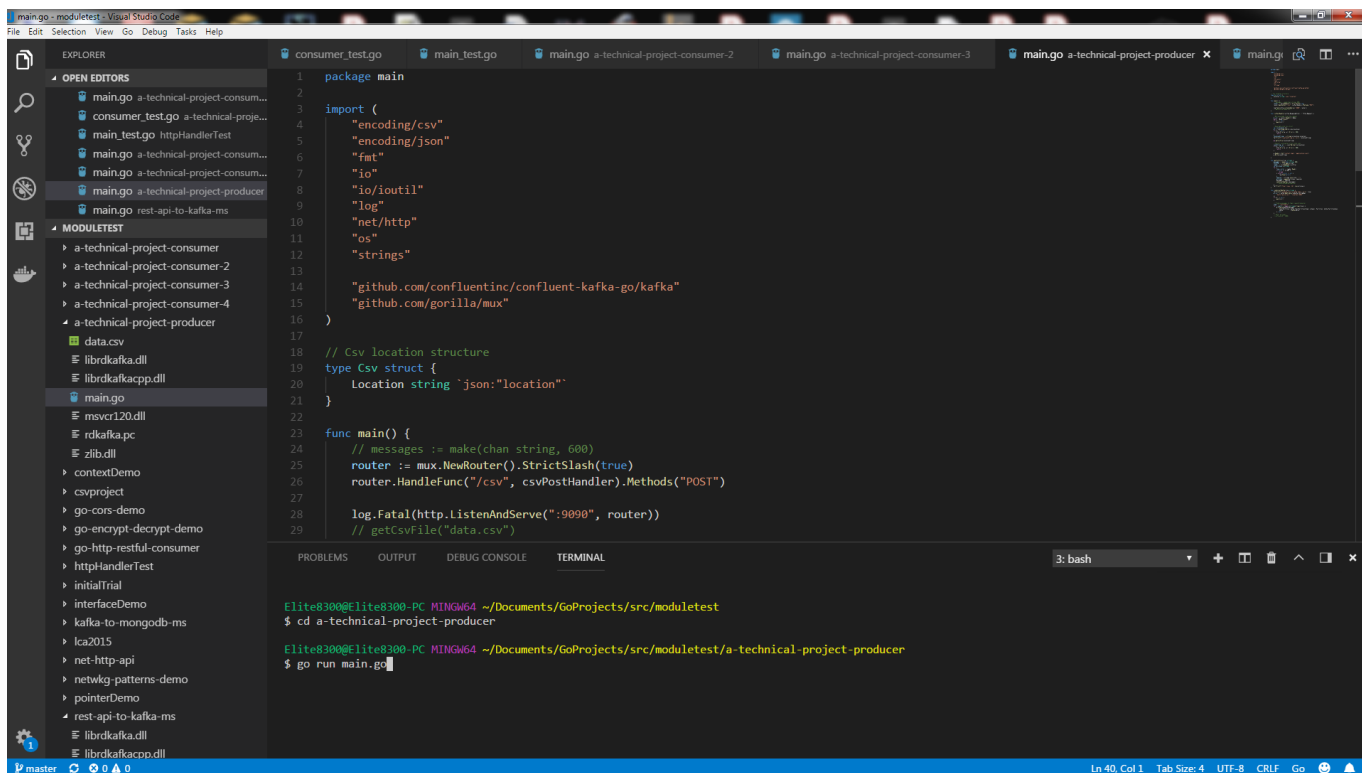


Fig 13: Producer code *main.go* file that takes a POST call with CSV file location and sends the CSV data lines as messages to the *testTopic* topic


```
package main

import (
    "encoding/csv"
    "encoding/json"
    "fmt"
)
```

```
Elite8300@Elite8300-PC MINGW64 ~/Documents/GoProjects/src/moduletest
$ cd a-technical-project-producer

Elite8300@Elite8300-PC MINGW64 ~/Documents/GoProjects/src/moduletest/a-technical-project-producer
$ go run main.go
locationString is: data.csv
sending line to kafka topic: 1,2,3,4,5,6
sending line to kafka topic: 7,8,9,3,2,1
sending line to kafka topic: rain,snow,bag,fly,dump,egg
sending line to kafka topic: 3,5,6,7,8,2
sending line to kafka topic: 8,4,6,7,2,2
sending line to kafka topic: rtru,aiis,shuss,syuud,didid,auao
sending line to kafka topic: 6,2,1,5,6,7
sending line to kafka topic: eueyue,dod,shs,jhs,hhdhd,sbhsa,dghhf
sending line to kafka topic: 3,5,6,8,9,0
sending line to kafka topic: rurur,ffof,hhsd,iudif,sgshys,osjd
sending line to kafka topic: 1,2,3,4,5,6
sending line to kafka topic: 7,8,9,3,2,1
sending line to kafka topic: rain,snow,bag,fly,dump,egg
sending line to kafka topic: 3,5,6,7,8,2
sending line to kafka topic: 8,4,6,7,2,2
sending line to kafka topic: rtru,aiis,shuss,syuud,didid,auao
sending line to kafka topic: 6,2,1,5,6,7
sending line to kafka topic: eueyue,dod,shs,jhs,hhdhd,sbhsa,dghhf
sending line to kafka topic: 3,5,6,8,9,0
sending line to kafka topic: rurur,ffof,hhsd,iudif,sgshys,osjd
sending line to kafka topic: 1,2,3,4,5,6
sending line to kafka topic: 7,8,9,3,2,1
sending line to kafka topic: rain,snow,bag,fly,dump,egg
sending line to kafka topic: 3,5,6,7,8,2
sending line to kafka topic: 8,4,6,7,2,2
sending line to kafka topic: rtru,aiis,shuss,syuud,didid,auao
sending line to kafka topic: 6,2,1,5,6,7
sending line to kafka topic: eueyue,dod,shs,jhs,hhdhd,sbhsa,dghhf
sending line to kafka topic: 3,5,6,8,9,0
sending line to kafka topic: rurur,ffof,hhsd,iudif,sgshys,osjd
sending line to kafka topic: 1,2,3,4,5,6
sending line to kafka topic: 7,8,9,3,2,1
```

Fig 14: Sample Console log of messages from CSV file after POST call to Producer API

```
run package tests | run file tests
package main

import (
    "testing"
)

run test | debug test
func TestSplitMessageInt(t *testing.T) {
    t.Run("1,2,3,4,5,6", testSplitMessageIntFunc("1,2,3,4,5,6", 0))
    t.Run("10,20,30,40,50,60", testSplitMessageIntFunc("10,20,30,40,50,60", 0))
}

func testSplitMessageIntFunc(intline string, expected int) func(*testing.T) {
    return func(t *testing.T) {
        actual := splitMessage(intline)
        if actual != expected {
            t.Errorf("Expected the sum of %s to be %d but instead got %d!", intline, expected, actual)
        }
    }
}

run test | debug test
func TestSplitMessageIntSingle(t *testing.T) {
    intline := "12,22,23,24,25,26"
    expected := 0
    actual := splitMessage(intline)
}
```

```
Elite8300@Elite8300-PC MINGW64 ~/Documents/GoProjects/src/moduletest
$ cd a-technical-project-consumer

Elite8300@Elite8300-PC MINGW64 ~/Documents/GoProjects/src/moduletest/a-technical-project-consumer
$ go test -v
```

Fig 15: Unit Test file `consumer_test.go` for the consumer app

```
consumer_test.go x main_test.go main.go a-technical-project-consumer-2 main.go a-technical-project-consumer-3 main.go a-technical-project-producer main.go  
run package tests | run file tests  
package main  
import (  
    "testing"  
)  
  
$ go test -v  
=== RUN TestSplitMessageInt  
--- PASS: TestSplitMessageInt/[1,2,3,4,5,6] (0.00s)  
msgStrmsgStr: 1,2,3,4,5,6  
Topic sum: 21  
=== RUN TestSplitMessageInt/[10,20,30,40,50,60]  
msgStrmsgStr: 10,20,30,40,50,60  
Topic sum: 210  
--- PASS: TestSplitMessageInt (0.00s)  
--- PASS: TestSplitMessageInt/[1,2,3,4,5,6] (0.00s)  
--- PASS: TestSplitMessageInt/[10,20,30,40,50,60] (0.00s)  
=== RUN TestSplitMessageIntSingle  
msgStrmsgStr: 12,22,23,24,25,26  
Topic sum: 132  
--- PASS: TestSplitMessageIntSingle (0.00s)  
=== RUN TestSplitMessageString  
msgStrmsgStr: car,plane,table,fence,bungalow  
Topic sorted: bungalow,car,fence,plane,table  
--- PASS: TestSplitMessageString (0.00s)  
=== RUN TestAddNumbers  
Topic sum: 21  
--- PASS: TestAddNumbers (0.00s)  
=== RUN TestSortWords  
Topic sorted: bungalow,car,fence,plane,table  
--- PASS: TestSortWords (0.00s)  
=== RUN TestDisplaySumValueToConsole  
Topic sum: 123  
--- PASS: TestDisplaySumValueToConsole (0.00s)  
=== RUN TestDisplaySortedValueToConsole  
Topic sorted: car,plane,table,fence,bungalow  
--- PASS: TestDisplaySortedValueToConsole (0.00s)  
PASS  
ok      moduletest/a-technical-project-consumer 0.066s  
$
```

Fig 16: Unit test results for consumer application.