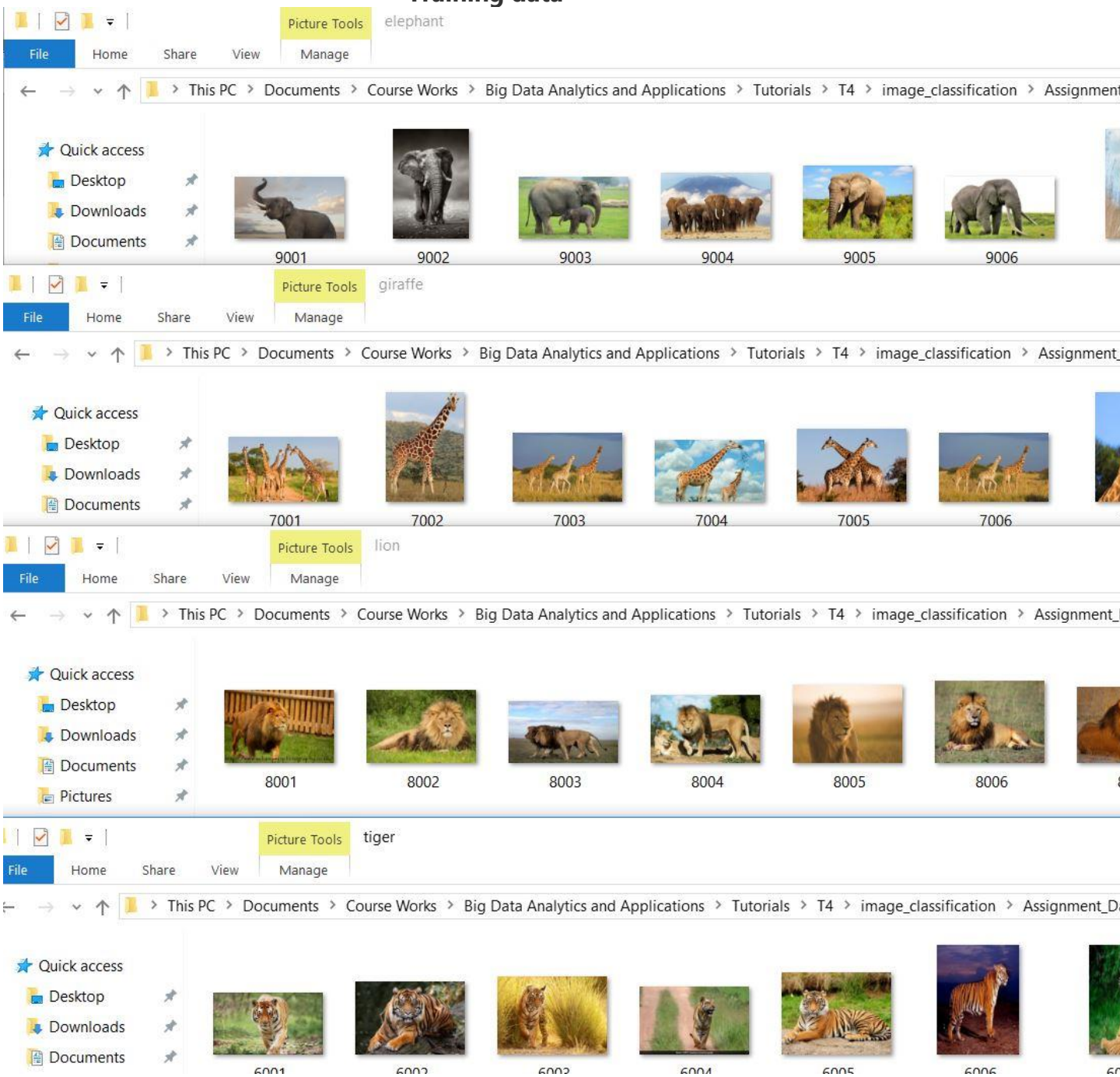
Big Data Analytics and Applications Lab Assignment 5

Task 1:

Training data



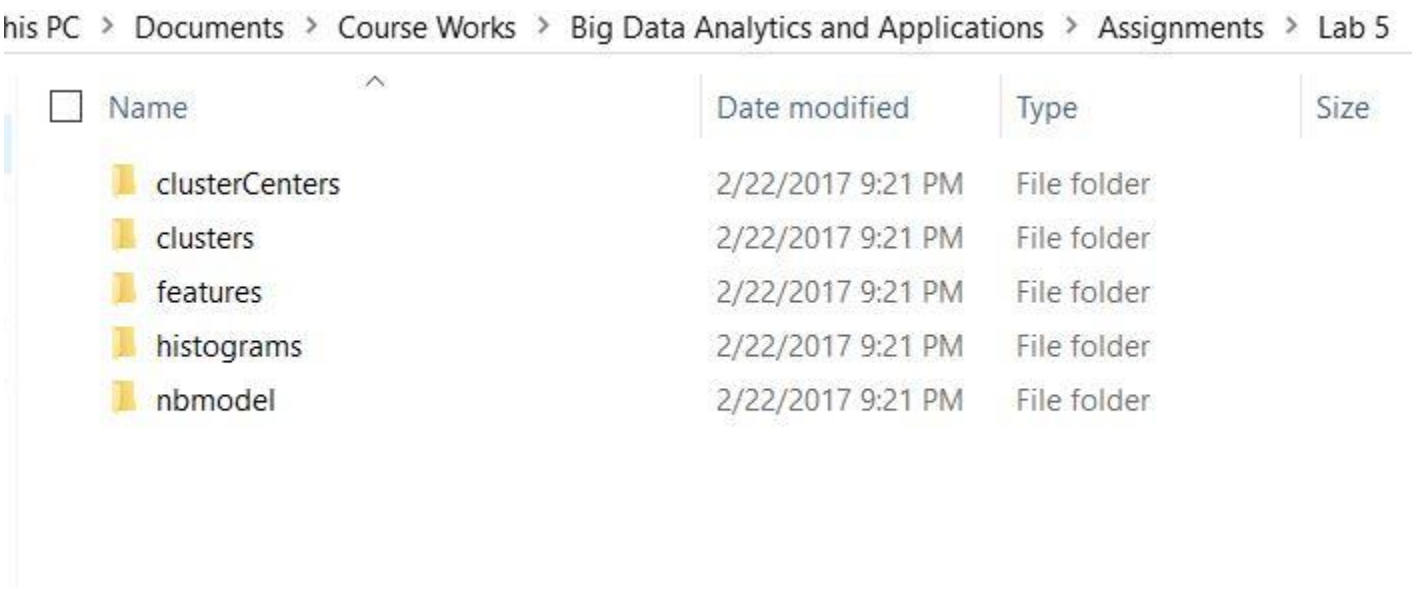
- Four image categories are taken in the source code. They are "elephant","giraffe","lion","tiger".

Image Categories

```
object IPApp {
  val featureVectorsCluster = new mutable.MutableList[String]

  val IMAGE_CATEGORIES = List("elephant", "giraffe", "lion", "tiger")
  /**
    *
    * @param sc     : SparkContext
    * @param images : Images list from the training set
    */
```

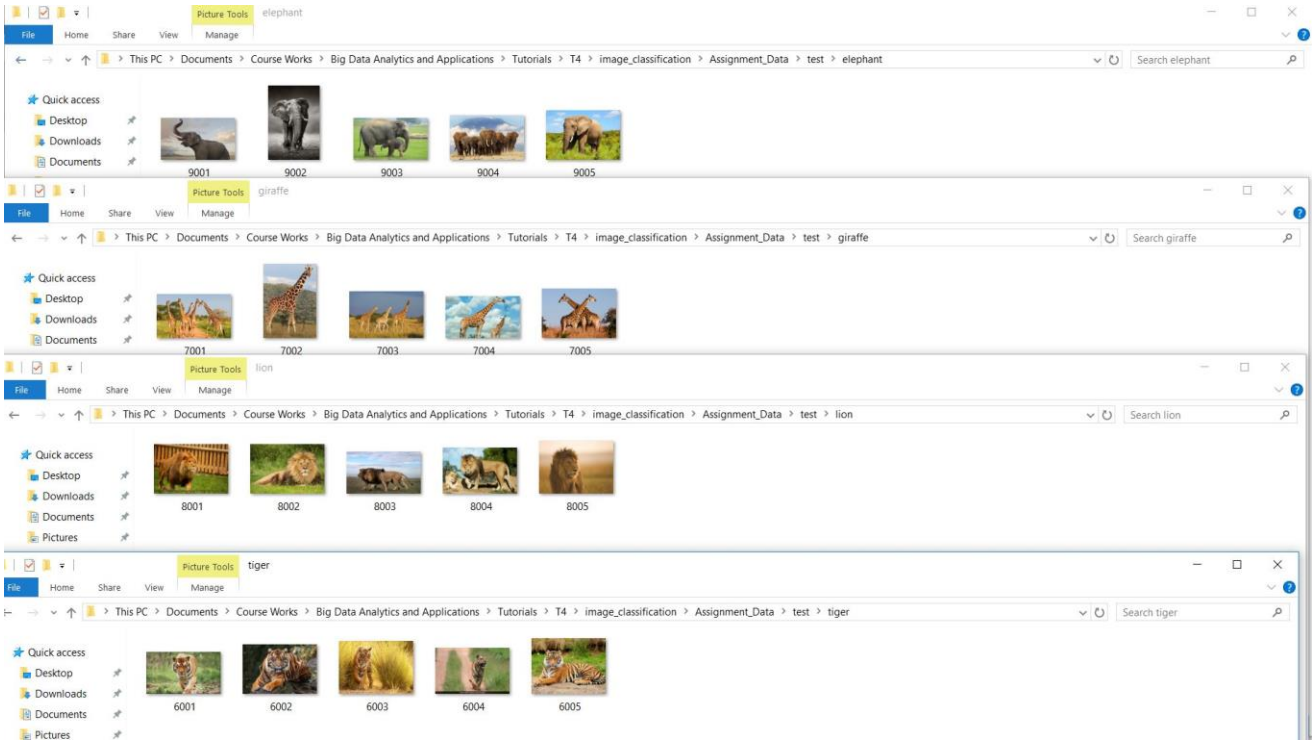
- Generated data was saved under 'features','clusters','clusterCenters','histograms' and 'nbmodel' folders.

Saved models



- Test data contains the same four classes and each class contains 5 images.

Test Data



- Following confusion matrix and an accuracy of 70% was recorded from this task.

Confusion Matrix and Accuracy

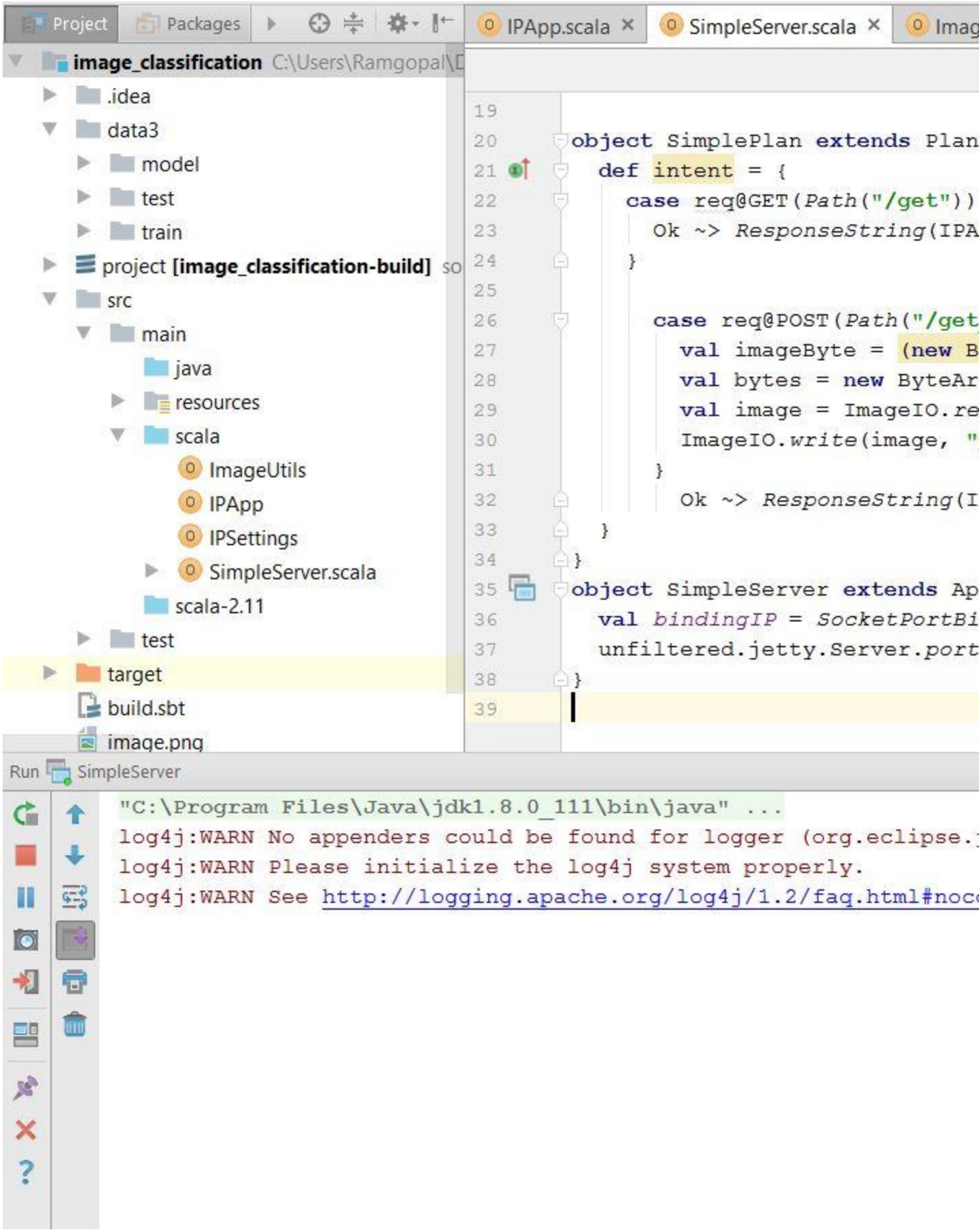
```
[Stage 4760:============================>                          (1 + 1) / 2]0.7
 |=================== Confusion matrix ==========================
4.0  0.0  0.0  1.0
0.0  4.0  0.0  1.0
0.0  1.0  3.0  1.0
0.0  1.0  1.0  3.0
0.7
17/02/22 21:18:14 INFO RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
17/02/22 21:18:14 INFO RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.

Process finished with exit code 0
```

Task 2:

Write a client application using the Spark API to connecting between Spark and your client. Your client can be either Web application or Android application. Refer to Tutorial 5 Spark API tutorial.

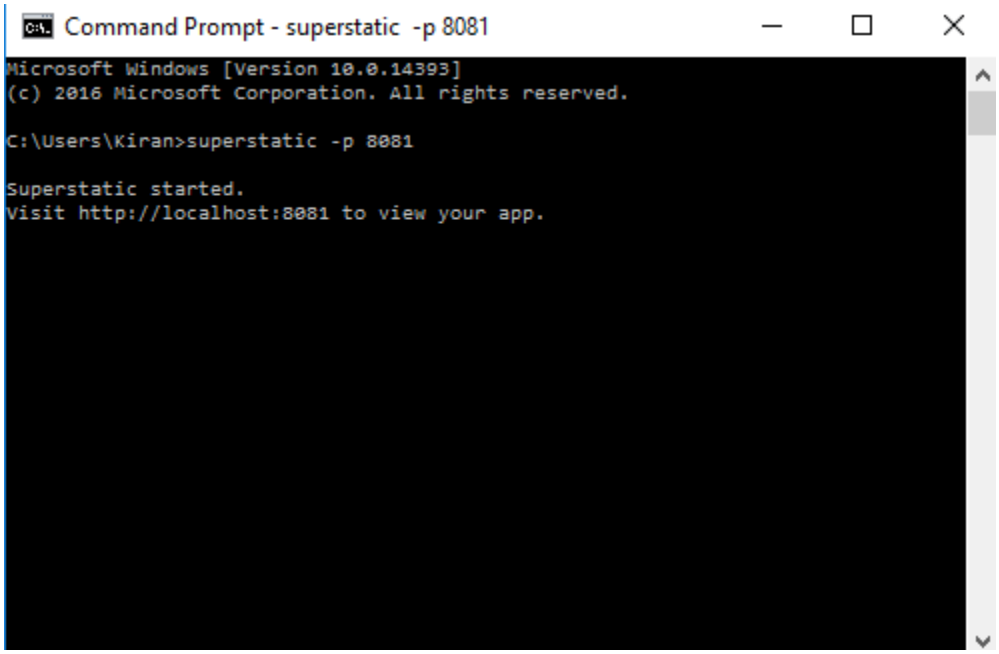
- To implement the client application for spark, we need to run the server initially. Following screenshot indicates the server running and waiting for the process.

Running Server

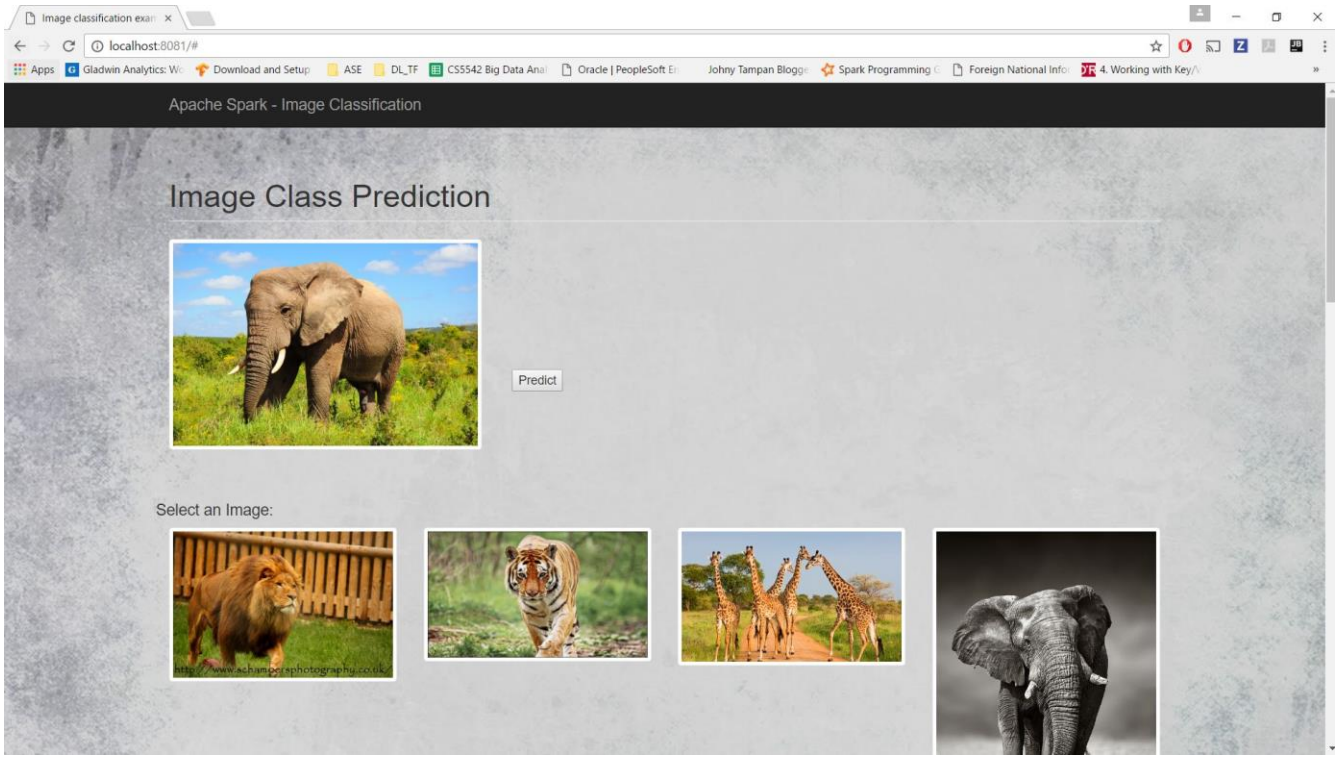

Project
Packages
image_classification C:\Users\Ramgopal\D
.idea
data3
model
test
train
project [image_classification-build] so
src
main
java
resources
scala
ImageUtils
IPApp
IPSettings
SimpleServer.scala
scala-2.11
test
target
build.sbt
image.png
IPApp.scala
SimpleServer.scala
Imag
19
20 object SimplePlan extends Plan
21 def intent = {
22 case req@GET(Path("/get"))
23 Ok ~> ResponseString(IPA
24 }
25
26 case req@POST(Path("/get
27 val imageByte = (new B
28 val bytes = new ByteAr
29 val image = ImageIO.re
30 ImageIO.write(image, "
31 }
32 Ok ~> ResponseString(I
33 }
34 }
35 object SimpleServer extends Ap
36 val bindingIP = SocketPortBi
37 unfiltered.jetty.Server.port
38 }
39
Run SimpleServer
"C:\Program Files\Java\jdk1.8.0_111\bin\java" ...
log4j:WARN No appenders could be found for logger (org.eclipse.
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noc

- By using superstatic from the command prompt, we can run the web page on server.

Superstatic command

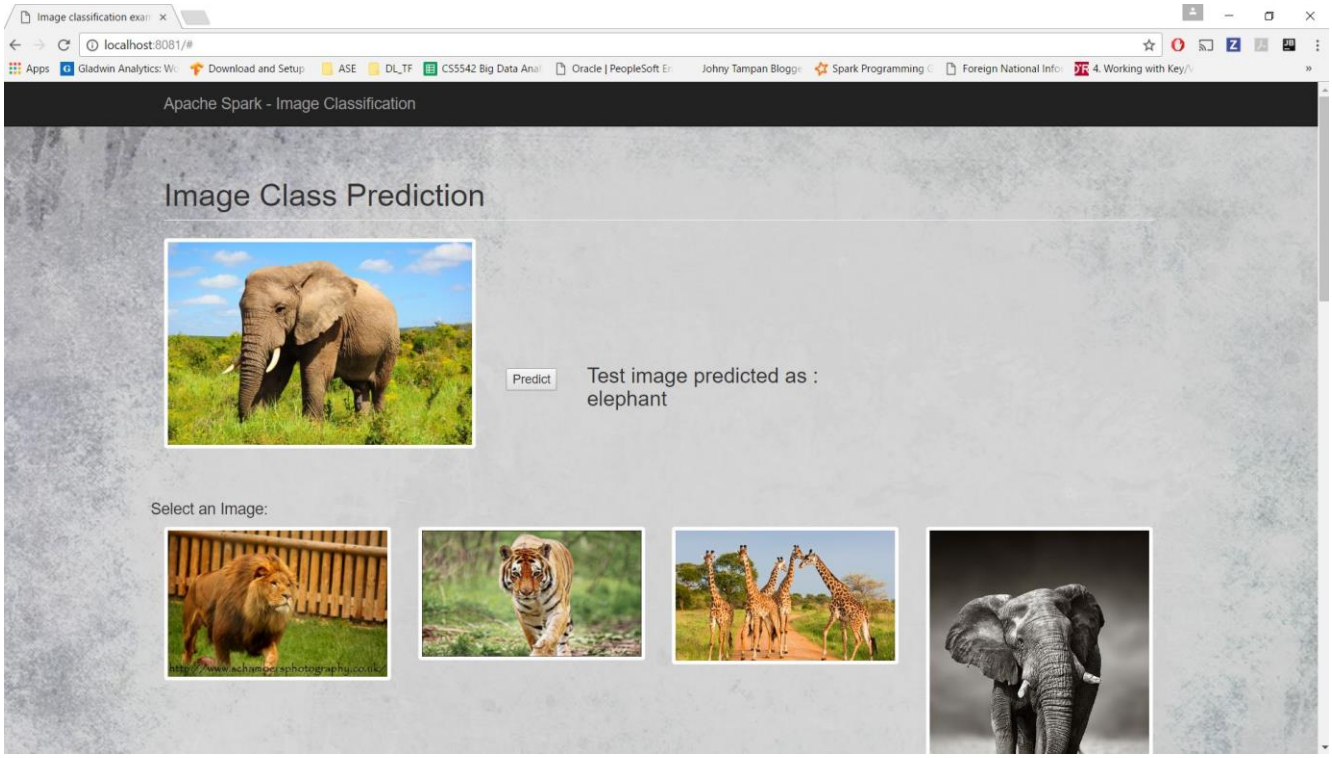


Client

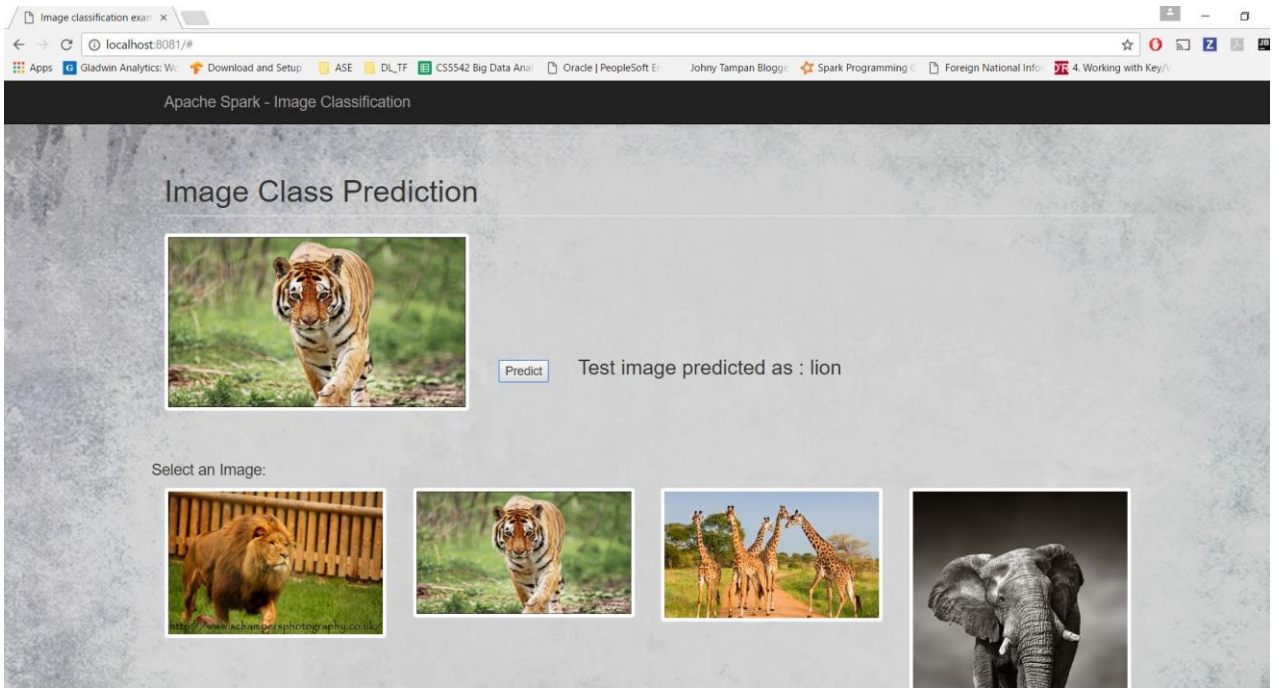


- My training model has an accuracy of 70%, so there were some wrong predictions as well.Following images indicate the predictions on the client side based on the executions on the server.

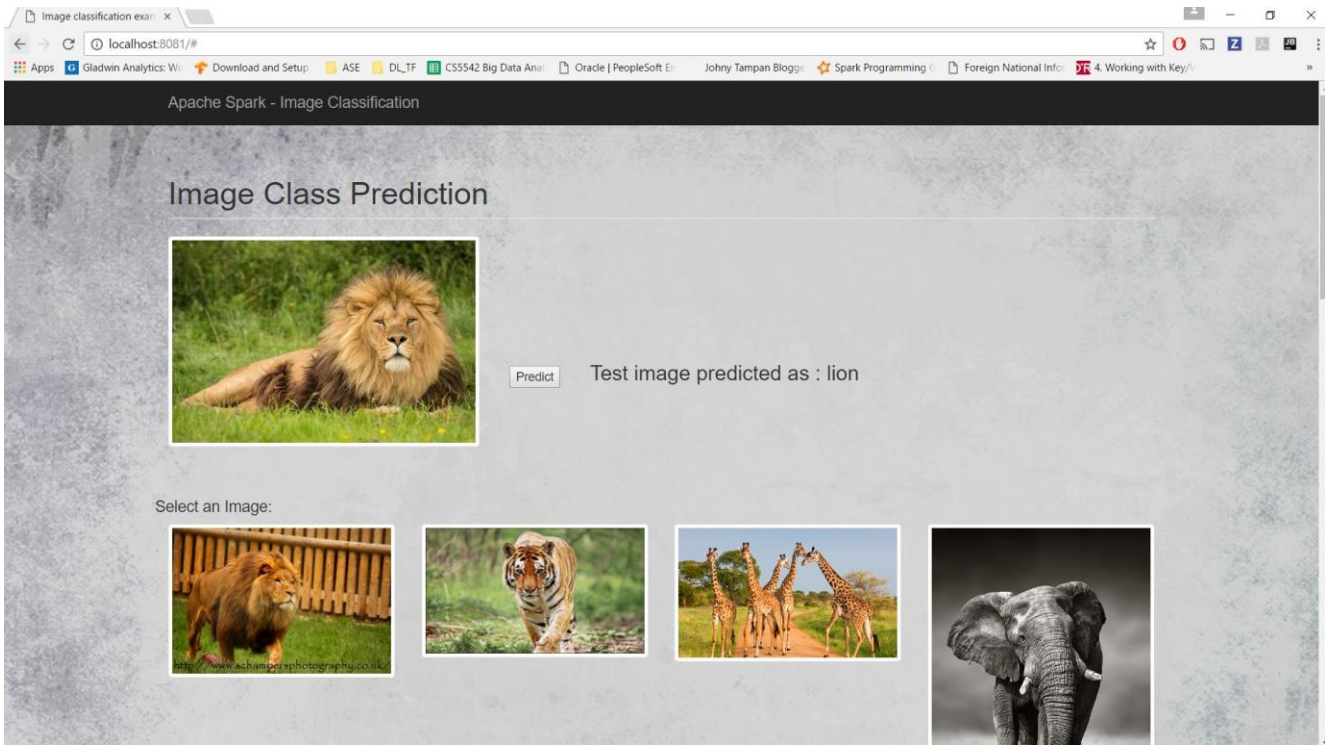
Elephant Prediction



Wrong Prediction: Tiger as Lion

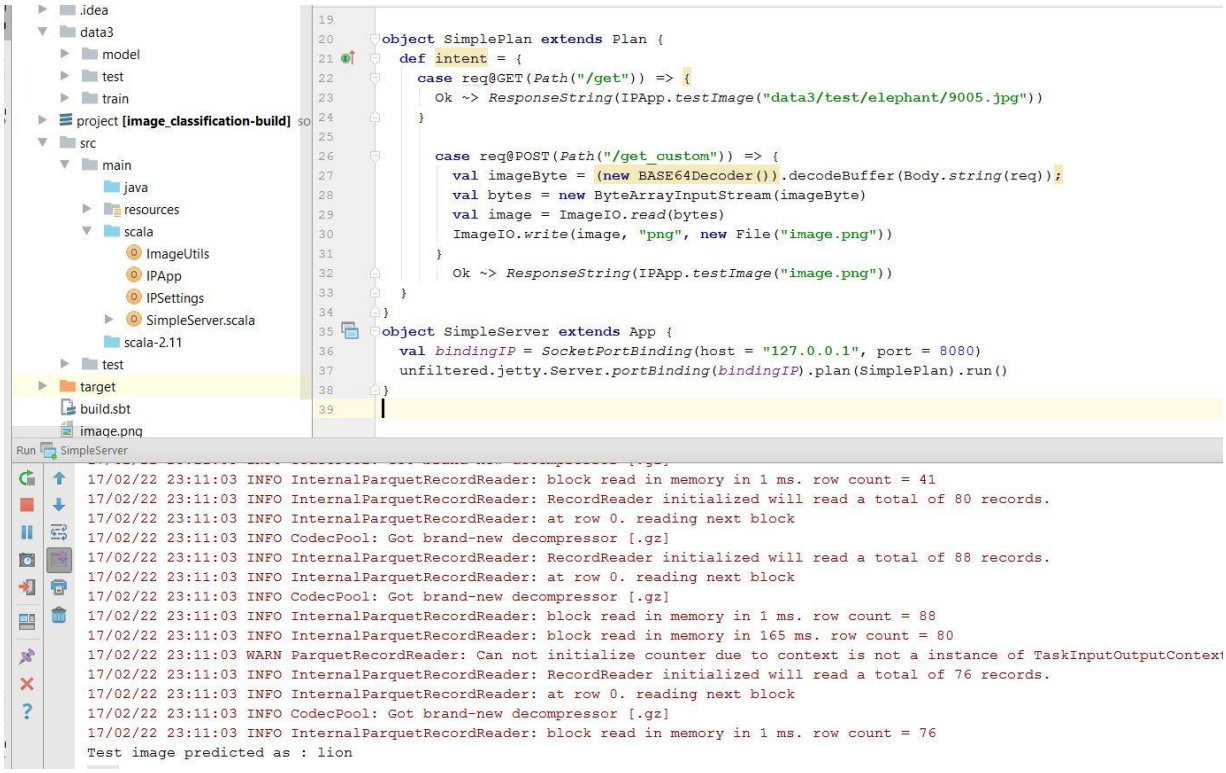


Lion Prediction



- Following image indicates the corresponding output on server console.

Server console

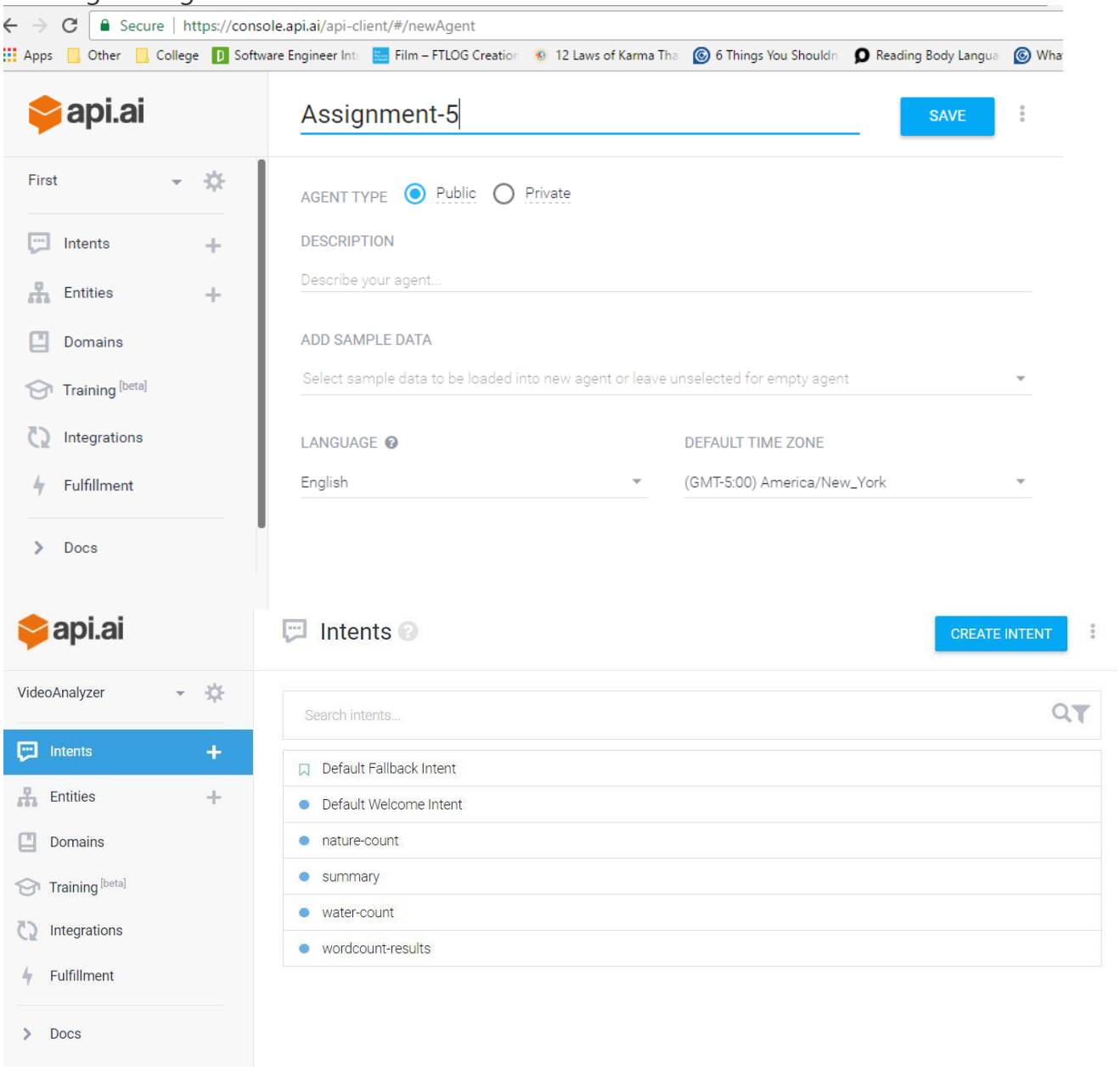


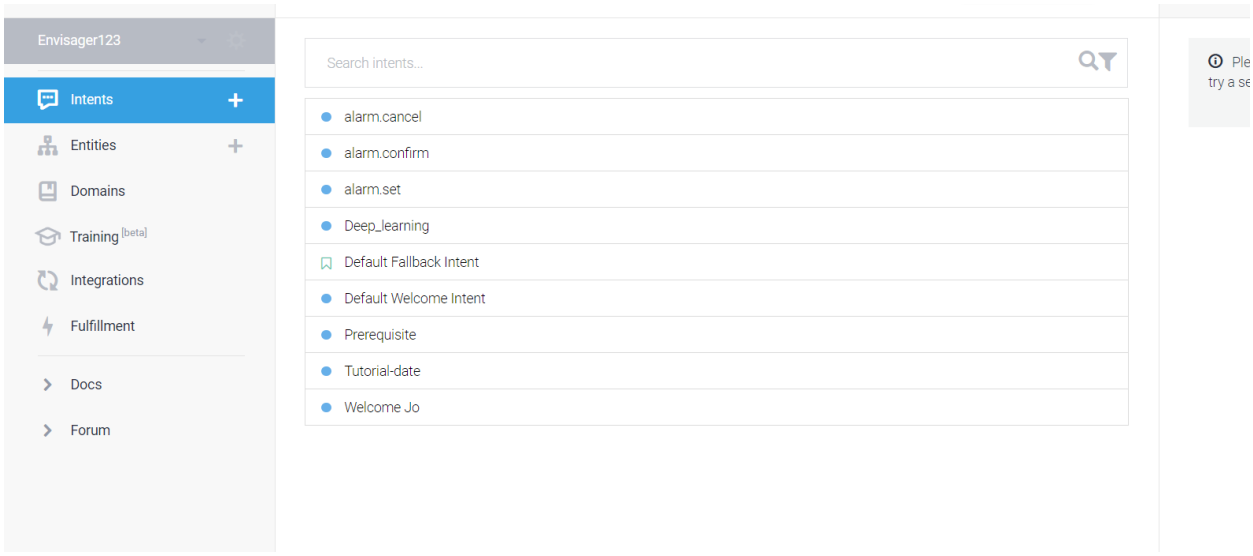
Task 3:

Build a simple application to have a conversion using Google Conversation Actions API about the summary you had generated about your video. Refer to Tutorial 5 Conversion Actions API tutorial.

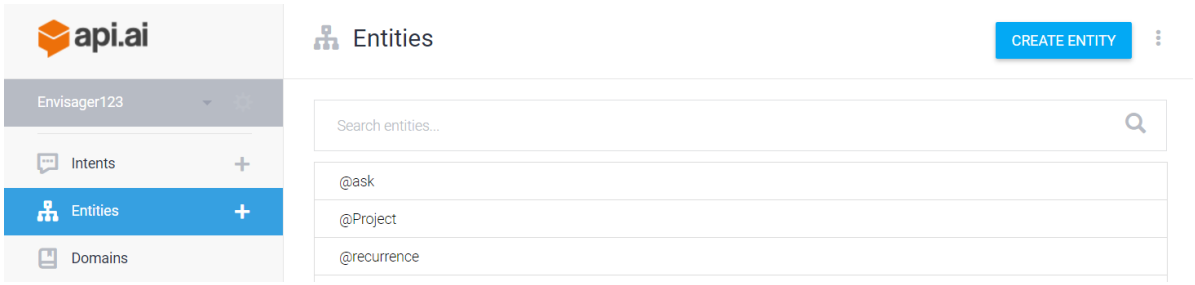
- To implement the conversation actions api, we need to start by creating a agent in api.ai. Then we need to create the corresponding Intents and Entities for the agent.

Creating and Agent

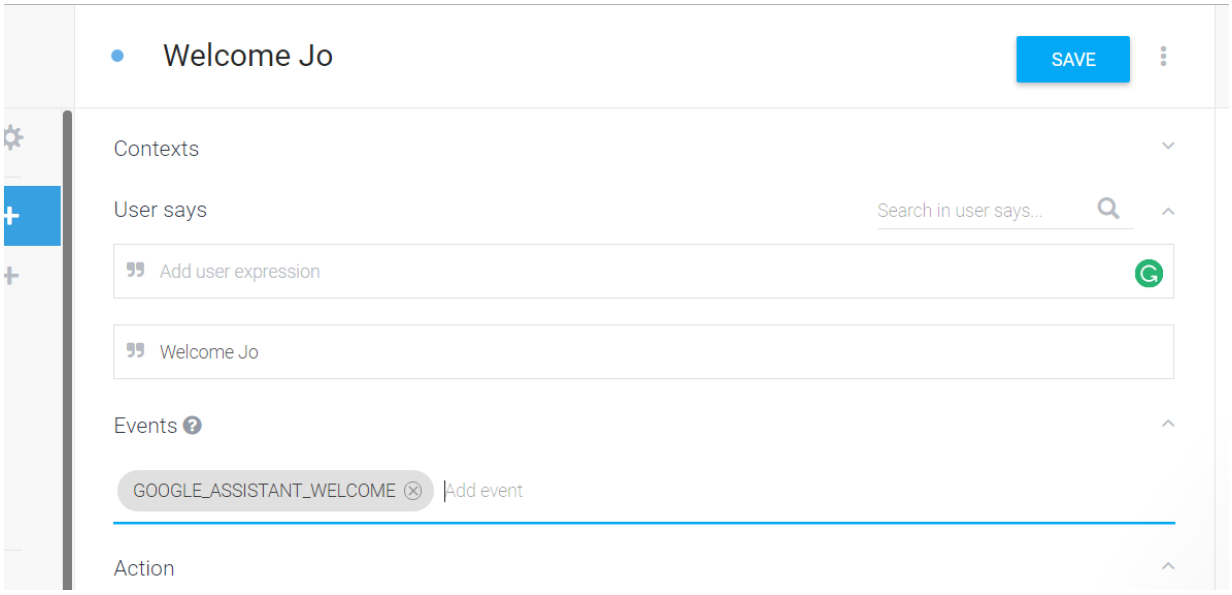




Entities were created to identify with different names



User queries were added



+ New parameter

Response

Text response

1	Hi How can I help you
2	How may I help you
3	Ask me anything
4	Enter a text response variant...

ADD MESSAGE CONTENT

Actions on Google