

**Activity 1 – Hands on creating a QnA chatbot**

In this activity, we will learn:

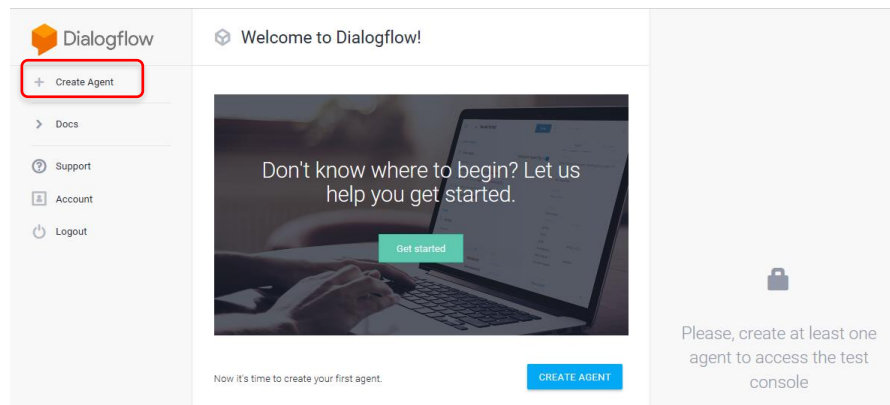
- ☐ Create DialogFlow Knowledge Base.
- ☐ Create an Integration to use a knowledge base
- ☐ Chat with the bot to verify the code is working
- ☐ Link Telegram to bot's channel

**1. Prerequisites**

- a) To start using Dialogflow Knowledge, you will need to have a Google account. Head over to here (<https://accounts.google.com/signup/v2>) to create one if necessary.

**2) Create an Agent**

- a) Login in to Dialogflow (<https://console.dialogflow.com/api-client/#/login> ).
- b) If it is the first time you login to Dialogflow with this account, you will be asked to allow Dialogflow to access your Google account. Click on **Allow** to continue.
- c) In the next screen, review your account settings. The only required action is to check “*Yes, I have read and accept the agreement*”. Click **ACCEPT** to continue.
- d) You may be asked to authenticate again. If everything is ok, you should see the following screen. Click on **Create Agent**



- e) Give your agent an appropriate name. We can leave the **DEFAULT LANGUAGE** and **DEFAULT TIME ZONE** settings. Click **CREATE** to continue. After a while, the screen will refresh and bring you to details configuration page.
- f) Click on **Setting** icon. In the **General** tab, turn on **BETA FEATURES** and click **Save**. Do note the **Dialogflow Knowledge** is still in beta at this point in time.

**Dialogflow** MyFirstAgent SAVE

**General** Languages ML Settings Export and Import Speech Share Advanced

**DESCRIPTION**  
Describe your agent

**DEFAULT TIME ZONE**  
(GMT+8:00) Asia/Hong\_Kong  
Date and time requests are resolved using this timezone.

**GOOGLE PROJECT**

Project ID	myfirstagent-cisxwh
Service Account ?	dialogflow-qnmpyc@myfirstagent-cisxwh.iam.gserviceaccount.com

**API VERSION**

**V2 API**  
Use Cloud API as default for the agent. Your webhook will receive and return V2 format messages.

**BETA FEATURES**  
☒ Enable beta features and APIs  
Be the first to get access to the newest features and latest APIs. ([Full V2-beta API reference](#))

**API KEYS (V1)**

**\*If "Service Account" is empty, click on the "+" button.**

### 3) Create Knowledge Base

- a) On the left panel, click **Knowledge [beta]**, and then **CREATE KNOWLEDGE BASE** to create a new knowledge base.

**Dialogflow** Knowledge Bases CREATE KNOWLEDGE BASE

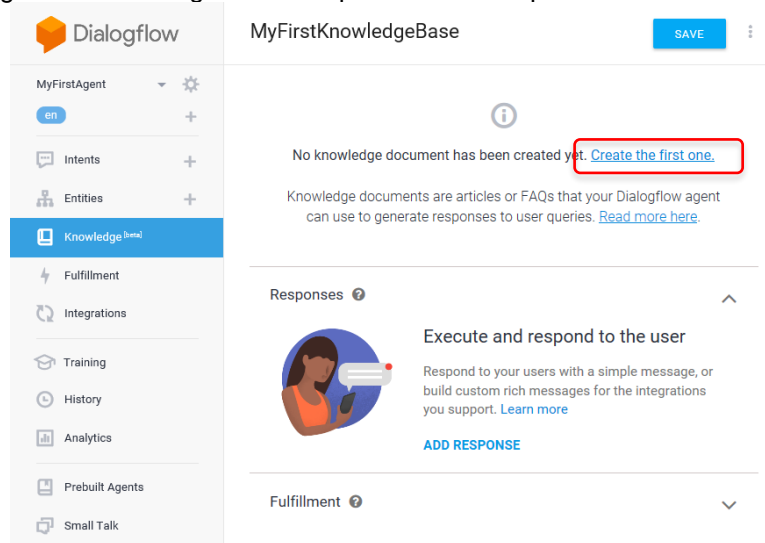
No knowledge base has been created yet. [Create the first one.](#)

Knowledge Connector provides your Dialogflow agent with the capacity to answer queries based on predetermined sources of knowledge such as websites, FAQs or knowledge base articles. [Read more here.](#)

**ADJUST KNOWLEDGE RESULTS PREFERENCE**  
When your query also matches an intent, specify how strongly you prefer knowledge results.

Weaker ————— Stronger

- b) Enter a suitable knowledge base name and click on **SAVE**.
- c) Click on **Create the first one** to create a knowledge document. Knowledge documents are articles or FAQs that your Dialogflow agent can use to generate responses to user queries.



- d) Enter values according to your data.
- Document Name:** this can be anything that you want
- Knowledge Type:** this can be selected as FAQ. For FAQ, only text/html and text/csv is supported for now.
- MIME type:** this is the type of data you are going to feed to the bot. It has options like (text/plain, text/html, text/csv, application/pdf)
- Data Source:** this can be provided as a file on cloud storage or local, you can also give this as a URL to a public page. For our purpose of this activity, we will use a URL. Enter <https://www.bicentennial.sg/frequently-asked-questions/>. You are free to try with another FAQ URLs.

Click **CREATE** to continue.

- e) Wait for it to generate all the data from the source that you have provided. This usually takes about 2–5 minutes depending upon the size of your data. You can see 'My First FAQ' knowledge base has been created.

Dialogflow MyFirstKnowledgeBase

Search documents

Document Name	Knowledge Type	Mime Type	Source/Path
My First FAQ ( View Detail )	FAQ	text/html	https://www.bicentennial.sg/frequently-asked-questions/

+ New Document

Responses

Execute and respond to the user

Respond to your users with a simple message, or build custom rich messages for the integrations you support. [Learn more](#)

ADD RESPONSE

Fulfillment

f) You can look at the content of data inside FAQ by clicking on '**view details**'. It should look something like this

Dialogflow My First FAQ

SAVE

Search question answer entries

1 OF 3

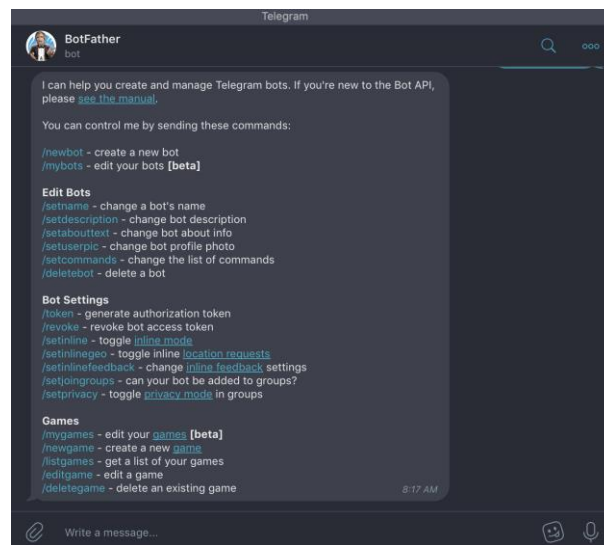
Question	Status
<b>Question:</b> Is it a fixed sequence to go through the Time Traveller first, before going for the Pathfinder? <b>Answer:</b> Time Traveller is a timed experience and visitors are allowed entry according to the ticket time chosen when booking online. The time is indicated in your email booking confirmation. Pathfinder is a free & easy experience which you can visit any time.	ENABLED
<b>Question:</b> Where is the nearest parking facility available? <b>Answer:</b> If you are driving to the Fort Canning Centre, the nearest parking available is at Fort Canning Carparks A and C. Please note that parking lots are limited. Alternatively, you may park at buildings within or near Fort Canning hill, including: Hotel Fort Canning Approximately 100 parking lots available. National Museum Singapore Outdoor parking facilities – approximately 30 parking lots available. SMU – School of Law Approximately 30 parking lots available. Liang Court Shopping Centre Approximately 727 parking lots available. Parking at Fort Canning is very limited. Visitors are encouraged to park at Liang Court Shopping Centre, National Museum Singapore or SMU, then take a Grab with our discount codes. Alternatively, park at Raffles City or Plaza Singapura, then take our free shuttle buses. Please set aside sufficient time from these alternative parking venues to Fort Canning Centre.	ENABLED
<b>Question:</b> Once I have booked online, where can I collect the free tickets?	ENABLED

- g) Click the menu button (three vertical dots) beside the SAVE button and select **back** in the dropdown menu to return to the Knowledge base view.
- h) Select "ADD RESPONSE"

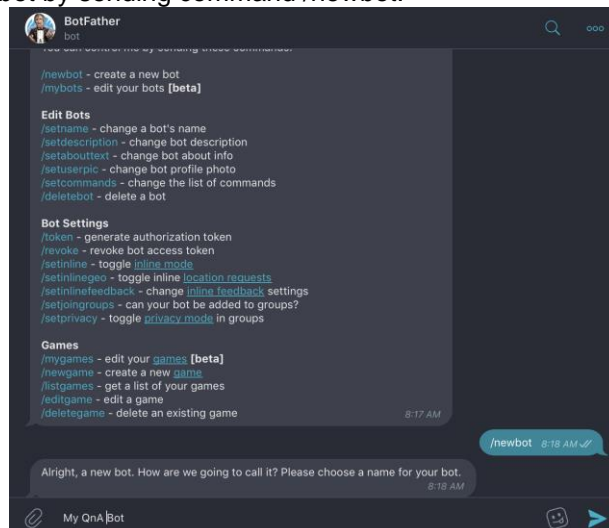
#### 4) Create telegram bot

For most integration, you must provide configuration information to run your bot with Dialogflow. Most channels require that your bot have an account on the channel, and others, like Facebook Messenger, require your bot to have an application registered with the channel also.

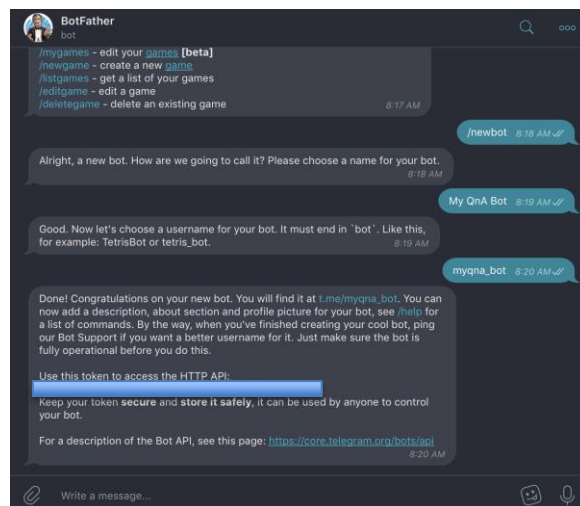
- If you do not have telegram installed on your mobile phone, please install via Google Play store or Apple App Store.
- If you have telegram installed on your laptop, use this [link](https://telegram.me/botfather) to connect to Bot Father (<https://telegram.me/botfather>) or add botfather. Otherwise, if you are using telegram on your mobile, start telegram and search for botfather.
- Click on **Start** to start a conversation with botfather.



- Create a new Telegram bot by sending command `/newbot`.



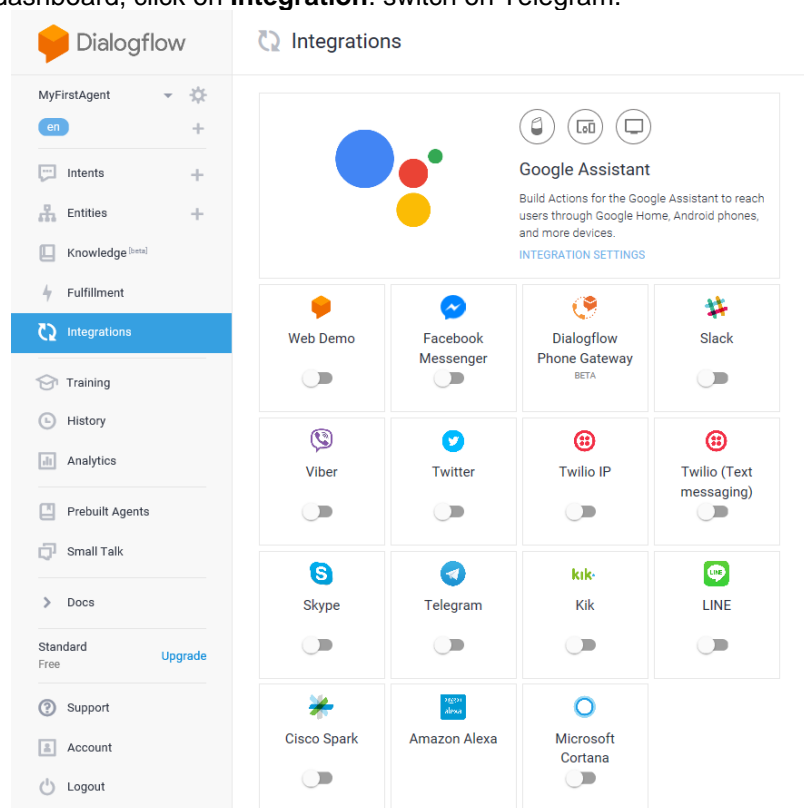
- Give the bot a friendly name.
- Specify a username.



- g) Copy the Telegram bot's access token provide in the screen above. You will need this info to configure integration for the Agent.

## 5) Integrate telegram with Dialogflow

- a) Back to Dialogflow dashboard, click on **Integration**. switch on Telegram.



- b) In the Telegram configuration pop up, enter the telegram bot's access token you obtained in Step 4. Click **START** and after a while, you should see a flash message saying the bot is started.

Telegram

Build a conversational bot for Telegram.

When your Dialogflow agent is ready, follow these instructions to connect it to your Telegram bot:

- Get a Telegram access token from BotFather and insert it in the 'Telegram Token' field.
- Click 'START' below.

[More in documentation.](#)

Telegram token

Choose an environment to use with this integration.

Environment  
Draft

START

c) Click on Close (top right) to finish the configuration.

## 6) Add Responses

a) Switch on to use responses from the DEFAULT tab as the first responses for telegram. Click **Save**.

Dialogflow

MyFirstAgent

en

Intents

Entities

Knowledge (beta)

Fulfillment

Integrations

Training

History

Analytics

Prebuilt Agents

Small Talk

Docs

Standard Free Upgrade

MyFirstKnowledgeBase

SAVE

Search documents

Document Name	Knowledge Type	Mime Type	Source/Path
My First FAQ ( View Detail )	FAQ	text/html	https://www.bicentennial.sg/frequently-asked-questions/

+ New Document

Responses ?

DEFAULT GOOGLE ASSISTANT TELEGRAM +

Responses from this tab will be sent to the Telegram integration.

Use responses from the DEFAULT tab as the first responses.

ADD RESPONSES

Set this intent as end of conversation ?

## 7) Small Talk

a) Your agent can learn how to support small talk without any extra development. By default, it will respond with predefined phrases. (By default, if you send "How are you?", agent will respond with "Sorry, can you say that again?")

**Dialogflow** **Small Talk** SAVE

MyFirstAgent en

- Intents
- Entities
- Knowledge [beta]
- Fulfillment
- Integrations
- Training
- History
- Analytics
- Prebuilt Agents
- Small Talk**

Your agent can learn how to support small talk without any extra development. By default, it will respond with predefined phrases. Use the form below to customize responses to the most popular requests.

User: How are you?  
Agent: Wonderful as always. Thanks for asking.

User: You're so sweet.  
Agent: Thanks! The feeling is mutual.

☒ Enable

Based on [Actions on Google policy](#), enabling Small Talk in its entirety will cause your action to be rejected. See [Import the prebuilt agent](#) for steps on how to import and select subsets of Small Talk features that comply with Action on Google's policy.

**Small Talk Customization Progress** 0%

Category	Progress	Percentage
About agent	<div></div>	0%
Courtesy	<div></div>	0%

**8) Test your agent from Telegram**

- Launch telegram and add your bot. You can search for your bot by adding a @ to the username you used in step 3.
- Click start to start a conversation with the bot.
- Start asking your bot!

Activity wrap-up:

We learn how to

- ☐ Create a Dialogflow agent.
- ☐ Create a Dialogflow knowledge base
- ☐ Use Small Talk to add personality to agent
- ☐ Integrate Telegram to agent



## Activity 2 – Training an Image Classifier

In this activity, we will:

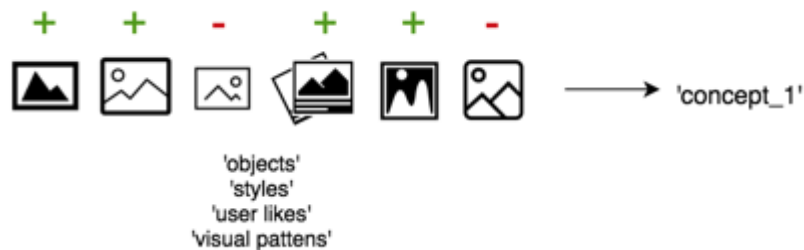
- ☐ Learn how to build a classifier through Clarifai AI service.
- ☐ Learn how to train a (supervised learning) image classification model
- ☐ Learn to evaluate the model
- ☐ Deploy your model to a chatbot

### NOTE:

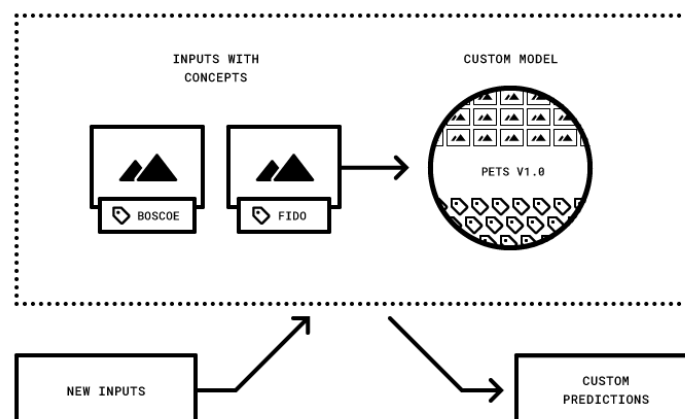
- ☐ **Same security and network settings may disable proper registration on Clarifai. Please check that you are using Guest@RP wireless network!**

### Custom Training

Custom Training is about teaching computers to see the world in a way that is specific to your own content and context. You can think of Custom Training as a series of inputs where you ultimately teach a neural network what concept\_1 is and what it is not. We define the training data, taxonomy (the model's 'concepts', this is equivalent to "labelling") and other performance characteristics.



If your content is visually distinct and easy to identify, 25-50 positive examples per concept will provide robust and accurate predictions. **A 'concept' is synonymous with 'tag', 'category' or 'keyword'.** Concepts are your business' world view as to what an object, visual pattern, or style may represent.



What makes for a well-built concept?

- **Accurate labels.** Mis-labelled images introduces noise into your model and can lead to weak or confusing predictions.
- **Balanced training data.** Skewed training sets where several concepts have 5-20x as many positive training images as others may affect model performance.
- **Matching training and prediction context.** It's crucial that your training images for your concepts resemble the conditions and context of imagery you'll be making predictions on.

As an example, training a flower identification model solely with stock photography and then attempting to predict on user generated smartphone photos will not be ideal.

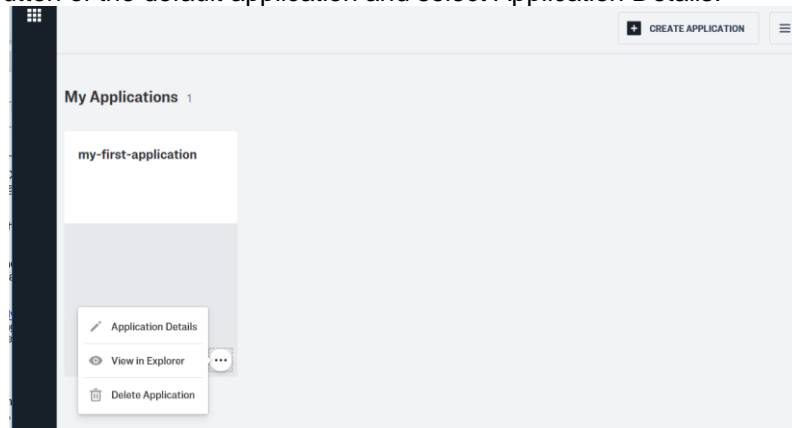
## 1. Create an account with Clarifai

- Sign up for an account at clarifai.com (<https://portal.clarifai.com/signup>)

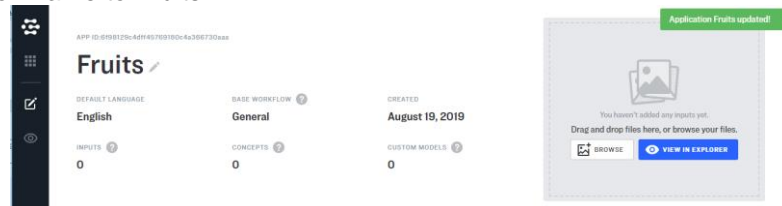
## 2. Edit your application

Operations are tied to an account and application, and any model that you create and add images to will be contained within a specific application. By default, you'll have an application in your account already so let's change that one's name to whatever you'd like. We are going to name ours "Fruits" for practical reasons

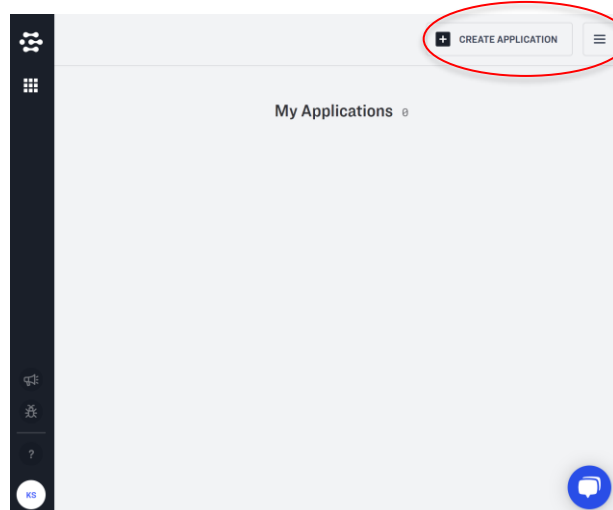
- Click on the details button of the default application and select Application Details.



- Change the application name to Fruits.



- Alternatively, you can also use "Create New Application" to create a new application and fill in the details as shown below.



### Create an application

A Clarifai Application is a workspace in which you can manage data and models. You will need an application to use any of our products.

Create your application here!

APP NAME

Fruits

DEFAULT LANGUAGE

English (en)

BASE WORKFLOW ?

General

Note: an API Key will be created automatically for this application.

CANCEL

CREATE

3. Copy the API key from the Application created: [https://cognimate.me:2635/vision\\_home](https://cognimate.me:2635/vision_home)

## API Keys ?

DETAILS

### Test2-all-scopes

96fdf30985094e5b899c78b3ba0e5486

Sep 3, 2020



4. Give your project a name and click on the “Set” button.

### What is your project name?

Unicorns vs Narhwals

Set

5. Paste the API key that you had copied earlier into the empty field and click on the “Set” button.

### What is your Clarifai key? ?

Paste API Key here

Set Key

[How-to guide](#)

[Sign up & get keys](#)

6. Key in the category of item that you want to train and click on “Add Category”.

### What categories should your model have? ?


Unicorns

Add Category

7. Drag and drop 10 or more images of the item of choice. (Suggest to have two or more categories)

**What categories should your model have?** ?

Orange




Drag and drop 10 or more images

8. Click on “Train Model” once you the images are uploaded.


**Train Model**

9. Drag and drop an image of choice from the “test\_image” folder which you had downloaded earlier followed by clicking on the “Predict” button.

**Upload an image to test your model**



Drop images here, or click to choose



28.4 KB

— or —

Category: Orange  
Confidence score: 100.00%

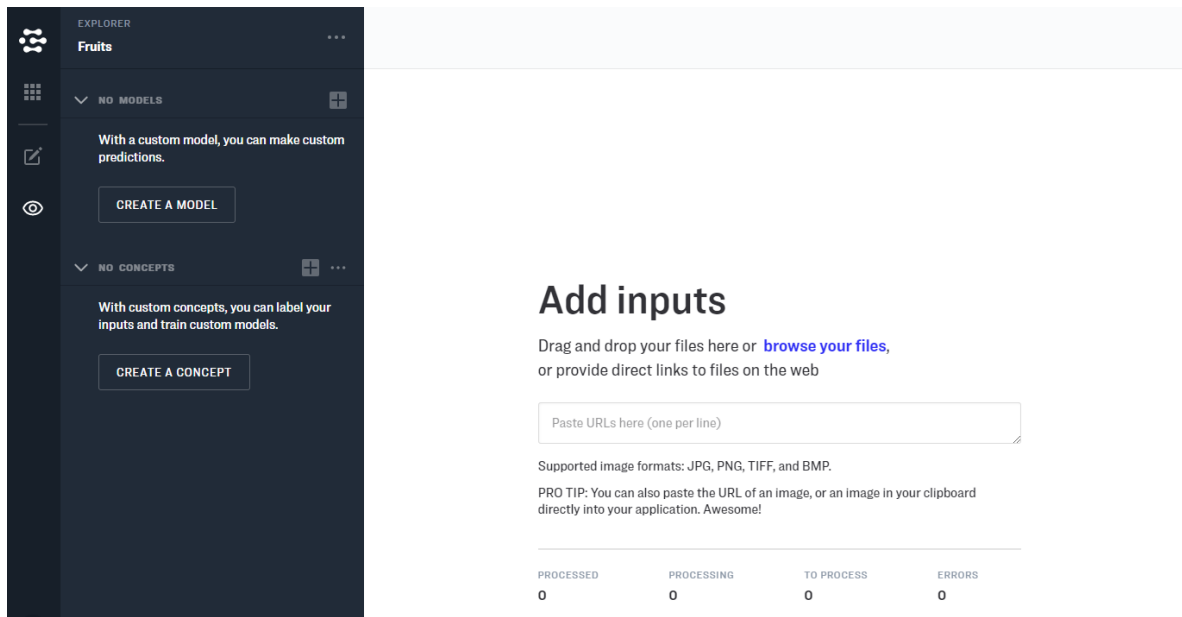
\*The predicted category will show together with the confidence level.

#You may also perform steps 3-9 directly in clarifai by doing steps 10-14.

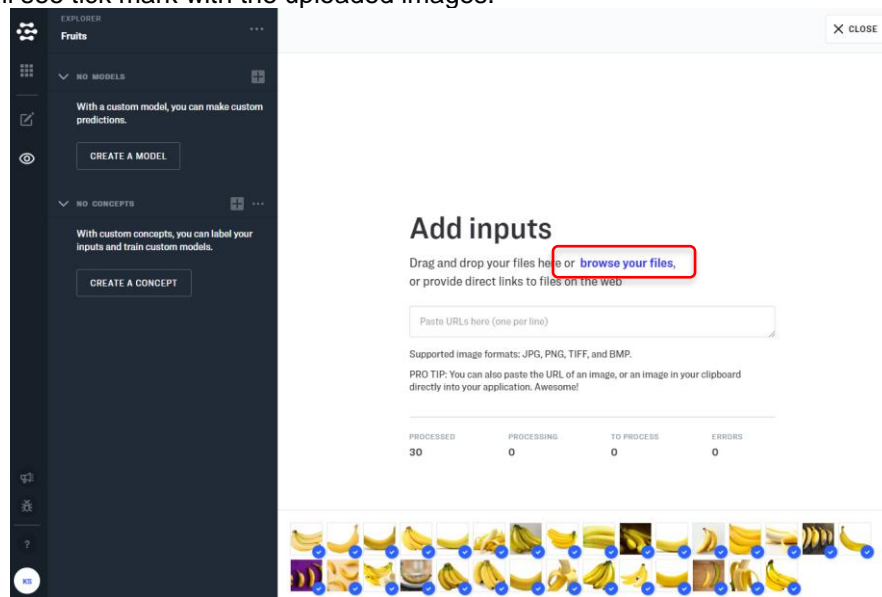
^If you have done the above, please skip to step 15.

10. Go to the following URL:

- Click on **VIEW IN EXPLORER** button to start adding images to your application. Click **browse your files**. Custom models are built by training on your own data, and they will be able to make predictions specific to your own unique content and context.



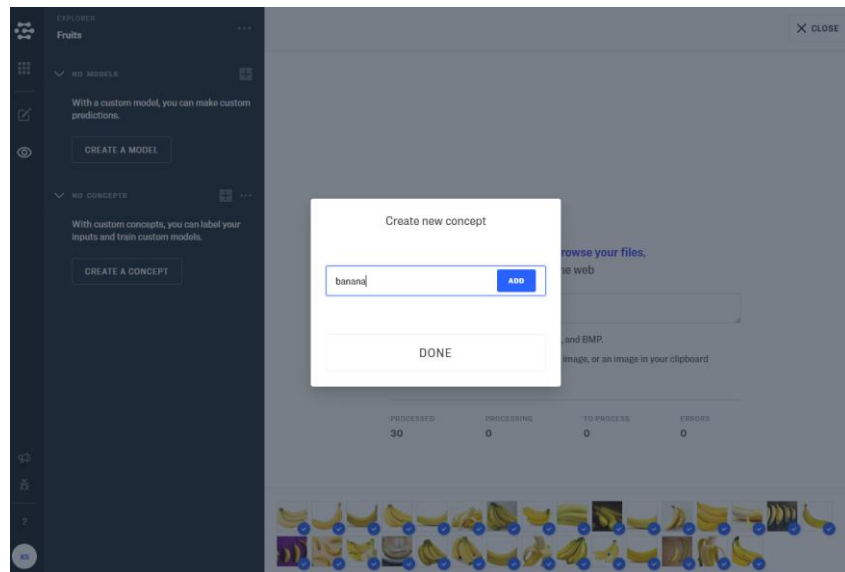
- b. Start uploading your images, say, those banana images. It may take a while to upload the images. Once uploaded, you will see tick mark with the uploaded images.



\*Note: It may take a while to upload the images.

## 11. Add concept(s)

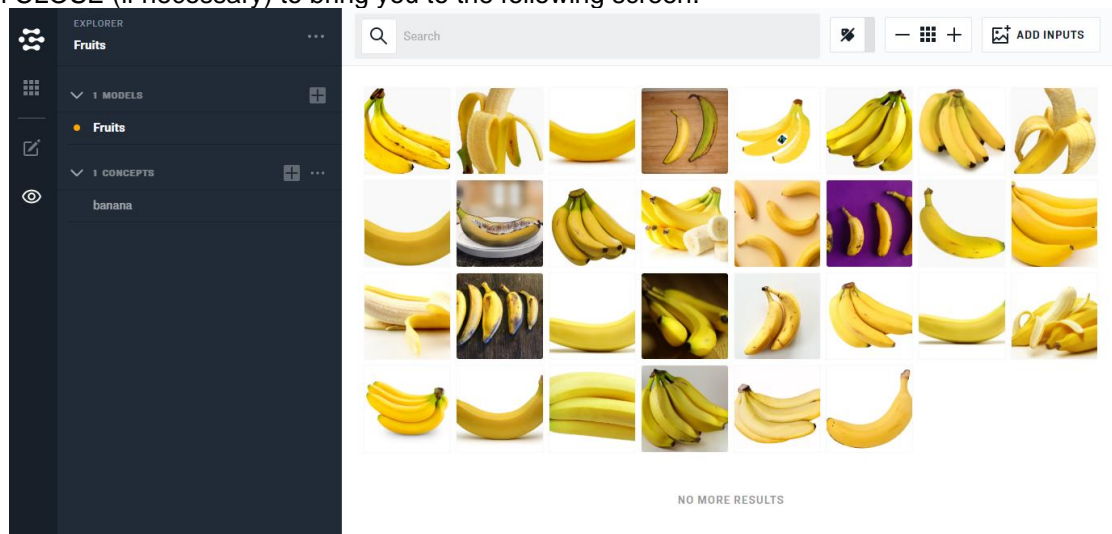
- Click on **CREATE A CONCEPT**.
- Enter banana and then click on **ADD** and then **DONE**.



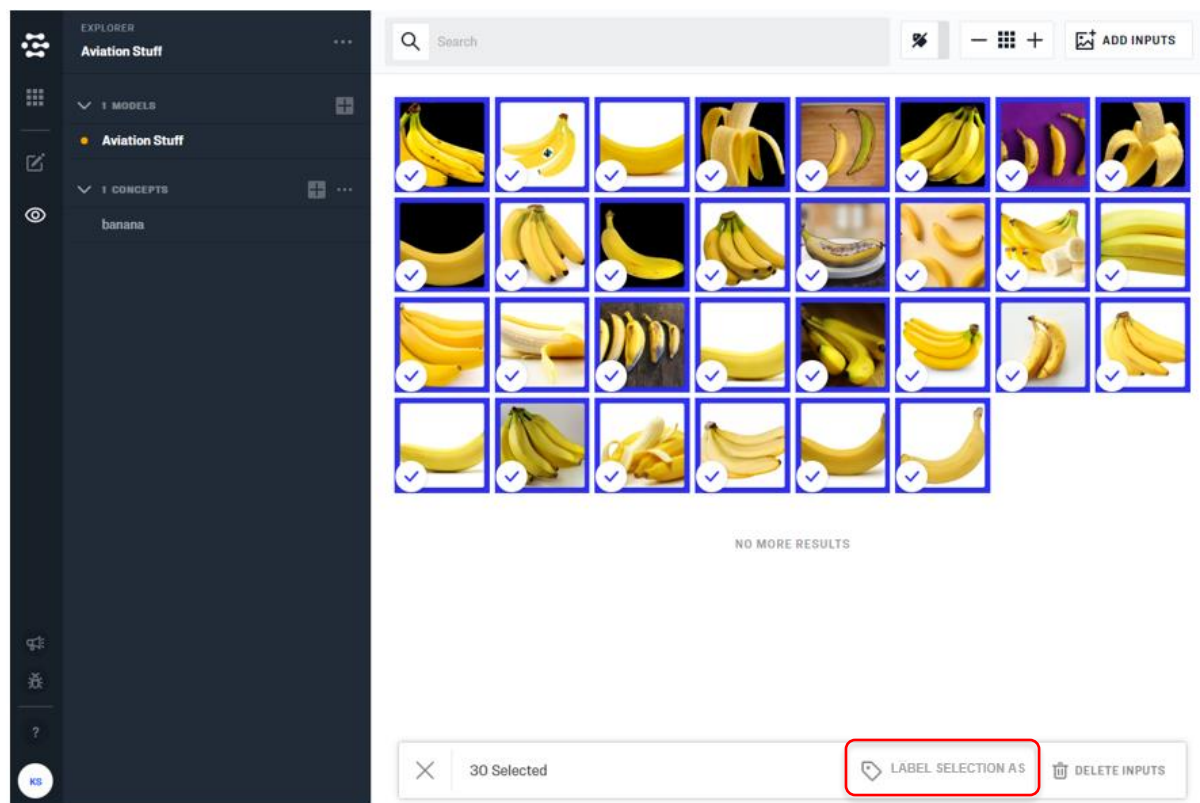
## 12. Label your images with those concepts

Now that you have some concepts, let's add them to your images! All images in your application are referred to as inputs so if you ever see that lingo, the names are essentially interchangeable. As you add concepts to images, you'll notice the counts in the Concept Panel on the left showing the updated total of how many images are labelled with each, respectively.

- a) Click on **CLOSE** (if necessary) to bring you to the following screen.



- b) Select all the banana images and select **ADD CONCEPTS**.



c) Select **banana**

## Label selection

ADD CONCEPTS TO INPUT

Create new concept

ADD

Include Concepts

☒ banana

LABEL WITH SELECTED CONCEPTS

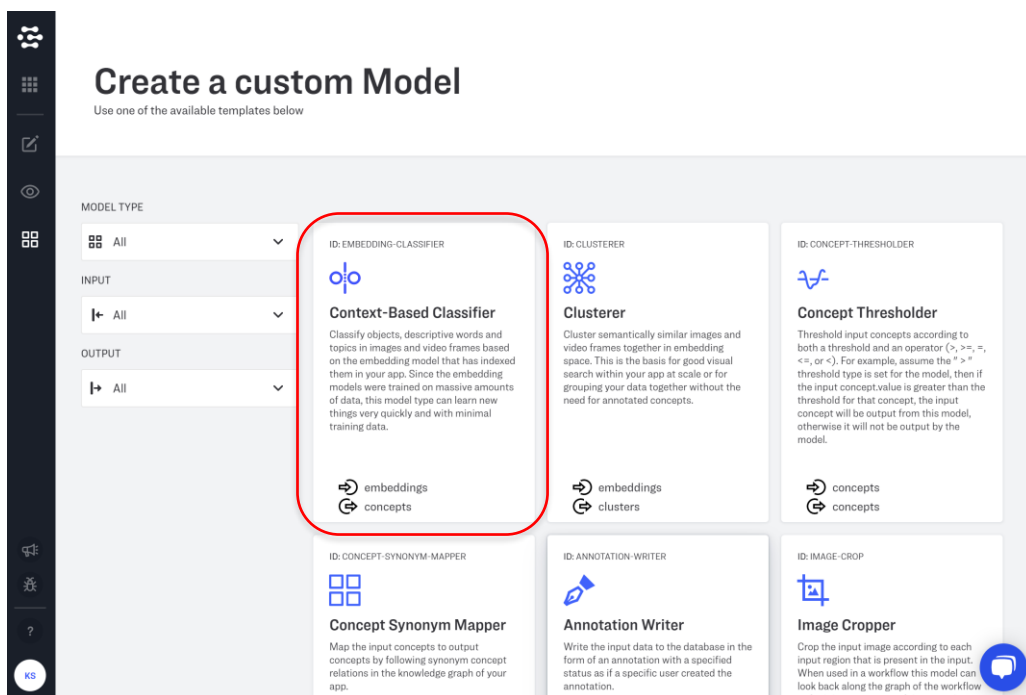
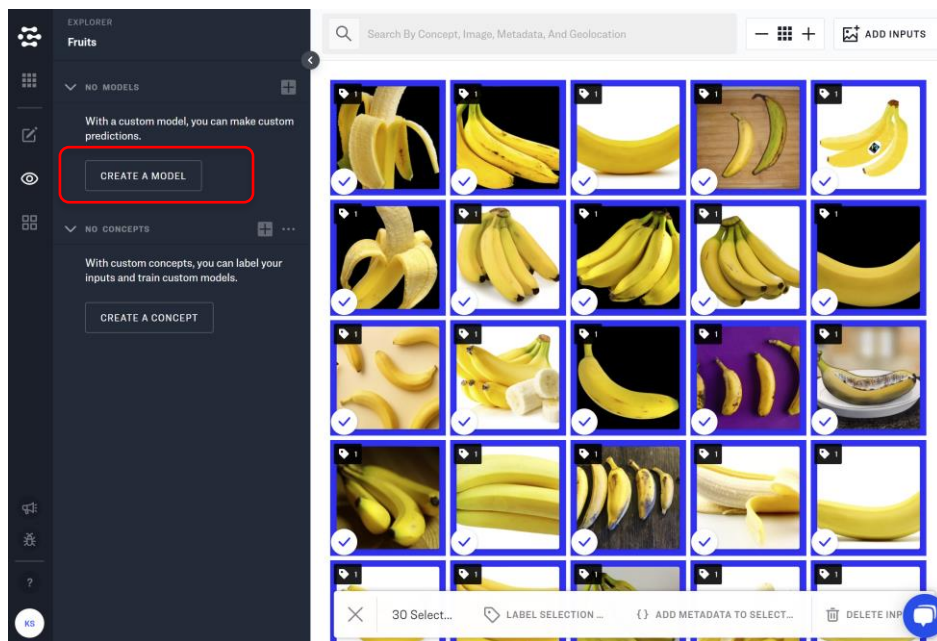
d) Click **LABEL WITH SELECTED CONCEPTS**.

### 13. Add more images and concepts

Repeat step 3 to 5 for the rest of the fruits folders (except test\_images)

### 14. Create and Train the model

Once all of your images are labelled, let's go ahead and create your first model by clicking CREATE A MODEL.



- a) Click on **Context-based Classifier**. Enter a name for the model and other details as shown. Click on **CREATE MODEL**.



## Create a Context-Based Classifier Model

Classify objects, descriptive words and topics in images and video frames based on the embedding model that has indexed them in your app. Since the embedding models were trained on massive amounts of data, this model type can learn new things very quickly and with minimal training data.

MODEL ID (OPTIONAL) ?

user-defined ID

DISPLAY NAME ?

Fruits

OUTPUT\_INFO.DATA.CONCEPTS \* ?

CONCEPT ID	CONCEPT NAME
coconut	coconut
banana	banana

Concept Name

OUTPUT\_INFO.OUTPUT\_CONFIG.CONCEPTS\_MUTUALLY\_EXCLUSIVE ?

If you expect multiple concepts for this model to be present per image, set 'Concepts Mutually Exclusive' to false.

NO

OUTPUT\_INFO.OUTPUT\_CONFIG.CLOSED\_ENVIRONMENT ?

If you will be training with images that do not contain any concepts for this model, set 'Closed Environment' to false.

NO

OUTPUT\_INFO.OUTPUT\_CONFIG.EMBED\_MODEL\_VERSION\_ID ?

bb186755eda04f9cbb6fe32e816be104

CREATE MODEL

Let's go ahead and train your first model. You can either do this via the little 3-dot menu next to the model name or you can click on the model name and then click the Train Model button on the ensuing page.

- b) Click on the application (ie Fruits). You should see the following. Click on **TRAIN MODEL**.

MODEL ID: Fruits

CREATED: Jul 1, 2020 at 11:18 AM

ACTIVE MODEL

TRAIN MODEL

Concepts 2 Versions

VERSION ID	STATUS	CONCEPTS	INPUTS	ROC AUC	METRICS	CREATED
684a1eb9ec1d4d4ac8c28d180c57d21d8	Model not yet trained	2				Jul 1, 2020 11:18am

### 15. Evaluate your model

- a) Click on the model name that appears on the left panel to bring you to the screen above.
- b) Click on **Versions**. It will take you to the following screen:

MODEL ID: Fruits

CREATED  
Aug 24, 2019 at 01:38 PM

ACTIVE MODEL TRAIN MODEL

EDIT MODEL DETAILS

Concepts 8 **Versions**

VERSION ID	STATUS	CONCEPTS	INPUTS	ROC AUC	METRICS	CREATED
4f8d32fe4fb442d289bd8cd39af575f7	Model trained successfully	8	288		EVALUATE	Aug 24, 2019 2:04pm
2cb2122a05da42af835fa1d81293583d	Model not yet trained	8				Aug 24, 2019 2:01pm
e657b1c2ae1149e29da9198132f41499	Model not yet trained	7				Aug 24, 2019 2:00pm
ec01cf662cc4483a898afd3853727d8a	Model not yet trained	6				Aug 24, 2019 1:58pm
0b82be31828c4fa2b5087eb938599acf	Model not yet trained	5				Aug 24, 2019 1:57pm
fe9dd420bfa94a4ea0d0a7457e9c9150	Model not yet trained	4				Aug 24, 2019 1:55pm
959924fa385e4c2fa972825fd5337a3e	Model not yet trained	3				Aug 24, 2019 1:53pm

- c) To evaluate your model, click the **evaluate** option under Metrics. This will take a short amount of time depending on the number of images added to your model. Our simple model should be evaluated in seconds. Once the evaluation is completed, the "Evaluate" option will be changed to a "View" option. Click it, and you will see the evaluation results.

Concepts 8 **Versions**

**Evaluation Summary Table**

CONCEPT	K-SPLIT AVG ACCURACY SCORE (ROC AUC)	I-SPLIT						
		TOTAL LABELED	TOTAL PREDICTED	TRUE POSITIVES	FALSE NEGATIVES	FALSE POSITIVES	RECALL RATE	PRECISION RATE
durian	1.000	6	6	6	0	0	1.000	1.000
apple	1.000	7	8	7	0	1	1.000	0.875
strawberry	1.000	9	9	9	0	0	1.000	1.000
banana	1.000	7	7	7	0	0	1.000	1.000
dragon fruits	1.000	8	8	8	0	0	1.000	1.000
pineapple	1.000	5	5	5	0	0	1.000	1.000
orange	1.000	9	9	9	0	0	1.000	1.000
coconut	1.000	7	7	7	0	0	1.000	1.000
TOTAL	AVG: 1.000	58	59	58	0	1	AVG: 1.000	AVG: 0.984

**Version Details**

ID: 4f8d32fe4fb442d289bd8cd39af575f7

CREATED AT: Aug 24, 2019 2:04 pm

UNIQUE CONCEPTS: 8

CONCEPTS MUTUALLY EXCLUSIVE: false

CLOSED ENVIRONMENT: false

**Selection Details**

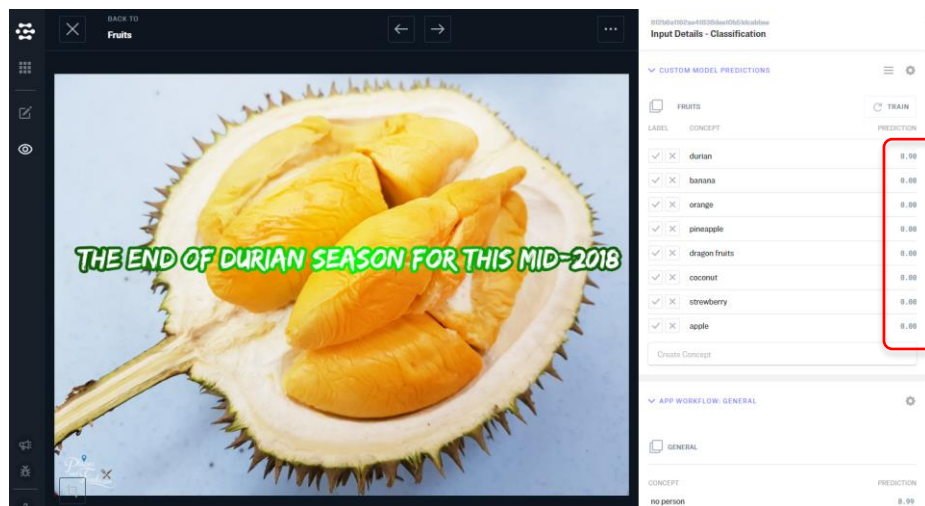
IMAGES LABELED	PREDICTION PROBABILITY FOR
apple	apple
	1.0000
	1.0000
	1.0000
	1.0000
	1.0000
	0.9998
	0.9992

**Concept by Concept Results**

**Co-occurrence**

- d) Evaluation results will be categorized under 4 headings: **Evaluation Summary Table**, **Concept by Concept Results**, **Co-occurrence**, **Precision-Recall**, and **ROC Curves**. Clarifai does the evaluation using a method called K-split cross-validation on the data.



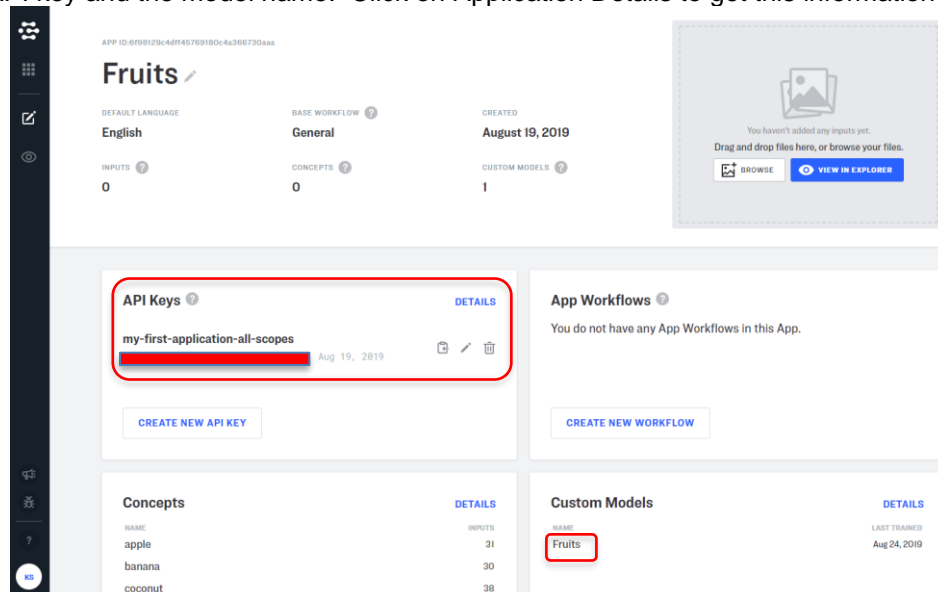


### 17. Adding negative examples when the model is getting confused (optional)

Up until this point we have only added positive examples to the model (e.g. saying "this is definitely concept X"), but we haven't added any negatives that say the opposite (i.e. "this is not concept X"). A robust and well performing concept is typically made up of both positive and negative examples with a 3 or 4:1 ratio, respectively, but adding too many will be counterproductive so be careful. If you see in one instance that you're getting a false positive for "apple", for example, you may want to teach the system that it's wrong.

### 18. Model Deployment and usage

- a) Once you trained a model, an API key is automatically created for you. To use this model in an application, you will need this API key and the model name. Click on Application Details to get this information from the screen.



**19. Use the model via a web app**

- Launch your web browser and go to <http://kwseow.pythonanywhere.com/forms>. This is a demo site created to utilise your model to perform image classification.
- Key in your Clarifai API key and the name of the model you trained in the previous step.
- Select a photo you want to test with and click on submit. After a short while you should see the result of the classification of your model. See the following screen capture:

**AI4E Clarifai image classification demo**

Clarifai API Key

Clarifai custom model name

No file chosen

**Success!** durian|0.9999454  
pineapple|0.0007022619  
coconut|3.5732985e-05  
dragon fruits|1.692772e-05  
orange|6.3478947e-06  
banana|3.1590462e-06  
strewberry|1.7881393e-07  
apple|2.9802322e-08

Using model:  
Fruits|Fruits



d) The following code snippet provide gives you an ideal of how this is done.

```

01 @app.route("/forms", methods=['GET', 'POST'])
02 def myforms():
03     form = ReusableForm(request.form)
04
05     #print(form.errors)
06     if request.method == 'POST':
07         form = ReusableForm()
08
09         if form.validate_on_submit():
10             #write_to_disk(name, surname, email)
11             f = form.photo.data
12             filename = secure_filename(f.filename)
13             f.save(os.path.join('./mysite/static/photos', filename))
14             try:
15                 #api_key = ""
16                 api_key = form.api_key.data
17                 os.environ["CLARIFAI_API_KEY"] = api_key
18                 clarifai_app = ClarifaiApp()
19                 #custom model
20                 #model_name="Fruits"
21                 model_name=form.model_name.data
22                 model = clarifai_app.models.get(model_name=model_name)
23                 response = model.predict_by_filename(os.path.join('./mysite/static/photos',
24                 filename))
25
26                 msg = ""
27                 for concept in response['outputs'][0]['data']['concepts']:
28                     msg += "%s\n"%(concept['name'],concept['value'])
29                 tmp_model = response['outputs'][0]['model']
30                 msg += "\nUsing model:\n%s\n"%(tmp_model['id'],tmp_model['name'])
31                 msg = msg.replace('\n', '<br>')
32             except ApiError as e:
33                 msg = 'Error status code: %d\n' % e.error_code
34                 msg += 'Error description: %s\n' % e.error_desc
35                 if e.error_details:
36                     msg += 'Error details: %s' % e.error_details
37
38                 if e.error_code == 21200:
39                     api_key=form.api_key.data
40                     os.environ["CLARIFAI_API_KEY"] = api_key
41                     clarifai_app = ClarifaiApp()
42                     msg += "\nAvailable models:"
43                     for model in clarifai_app.models.get_all():
44                         msg += "\n%s"%model.model_name
45
46                 msg = msg.replace('\n', '<br>')
47
48                 #flash('Hello: {}'.format(filename))
49                 flash('{}\n'.format(msg))
50                 #filename = secure_filename(form.file.data.filename)
51                 #form.file.data.save('uploads/' + filename)
52                 #flash('Hello: {} {} {}'.format(name, surname,filename))
53
54             else:
55                 flash('Error: All Fields are Required')
56
57         return render_template('form.html', form=form)

```

#### Activity wrap-up:

We learn how to:

- ☐ Train and evaluate an image classifier
- ☐ Deploy the model for use in a chatbot and a web browser