Collect images

How to connect to services

```
twitter = ServiceConnect["Twitter"];
```

Twitter

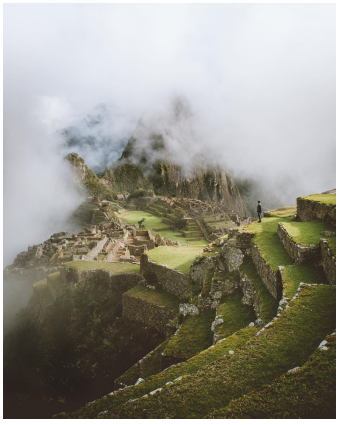
Important: In case there's more than one image per tweet, only the first one will be capture

Sample Tweets ID :

- With photo :
  - 1123373534070353920
  - 1123271151751446530 (2 photos)
- No photo:
  - 1123265083792957445

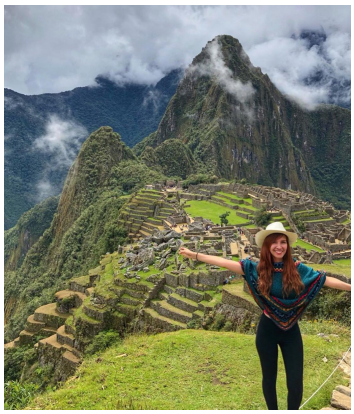
Final code:

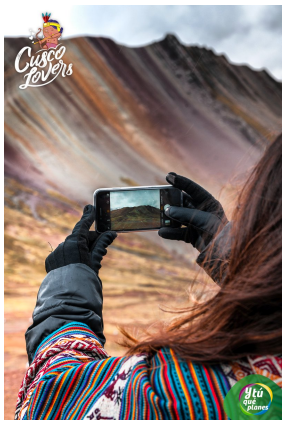
```
rawTweets = twitter["TweetSearch", "Query" → "#cusco", MaxItems → 50];
imageTweets = Select[rawTweets, MemberQ[Keys[#["Entities"]], "media"] &];
Function[list, list[[1]] → list[[2]]] /@
 Normal[imageTweets[All, {"ID", #["Entities"]["media"][[1]]["media_url"] &}][
    DeleteDuplicatesBy[Last]][All, {2 → Import}]]
```

Out[•]= {1 123 484 461 377 757 190 →  ,

1 123 381 570 029 670 400 → 

,

1 123 364 585 128 353 792 → 
,

1 123 363 915 625 164 800 → 

,

1 123 361 630 765 686 784 → 

,

1 123 351 688 046 366 721 →

}

Step by step

- **Note:** twitter[“TweetSearch”] returns a dataset. For more information about datasets, check https://reference.wolfram.com/language/ref/Dataset.html

Step 1: Get the last 3 tweets with the string “#cusco”

In[•]:=
```
lastTweets = twitter["TweetSearch", "Query" → "#cusco", MaxItems → 50];
```

- **Note:**
- The column Entities includes a list of datasets with additional information about the tweet.
- If the tweet includes an image (or imageS), you will find it in url in **Entities > media > media_url.** “media” is one of the datasets inside Entities
- If there are no images in the tweet, the dataset “media” won’t exist

Step 2: Erase all rows with tweets without image

Subtask 2.1: Get all Entities datasets (keys) in first tweet

```
firstTweet = First[lastTweets];
keysEntitiesFT = Keys[firstTweet["Entities"]]
```

Out[•]=

hashtags
symbols
user_mentions
urls

Substep 2.2: Check if “media” is inside this datasets (keys)

```
MemberQ[keysEntitiesFT, "media"]
```

Out[•]= True

Substep 2.3: In a dataset of tweets, select the ones that return True when checking if “media” exists

- **Notes:** *Select* selects the elements of a list (can be also a dataset) that return True when pass through a function. For more information, check <insert webpage>

- In this Step, the first parameter is the full dataset with tweets and the second one is a function (HasMedia) that returns True if the tweet has "media" inside "Entities"

```
HasMedia[tweet_] := MemberQ[Keys[tweet["Entities"]], "media"]
```

```
tweetsWithImage = Select[lastTweets, HasMedia]
```

Out[•]=

ID	Text
1 123 484 461 377 757 190	RT @BonfirePictures: Machu Picchu ˙·.
1 123 390 618 477 191 168	RT @BonfirePictures: Machu Picchu ˙·.
1 123 381 570 029 670 400	Thanks for the dining tip Kevin @CachiLife ... we found Chicha and just go
1 123 373 534 070 353 920	RT @BonfirePictures: Machu Picchu ˙·.
1 123 364 585 128 353 792	Machu Picchu ˙·.
1 123 363 915 625 164 800	¡Nunca dejamos de estar enamorados de ⌗Cusco♥ ! ˙·.
1 123 361 630 765 686 784	Aqui no sul não se confia nem nos bicho. ▯▯▯▯▯▯▯▯▯▯
1 123 351 688 046 366 721	.@DEVIDAPERU capacitó a agricultores de los distritos de San Gabán, ⌗Pur

- Instead of declaring a function, *Select* supports pure functions (the # and &). The next function gives the same result:

```
tweetsWithImage = Select[lastTweets, MemberQ[Keys[#["Entities"]], "media"] &]
```

Out[•]=

ID	Text
1 123 484 461 377 757 190	RT @BonfirePictures: Machu Picchu ˙·.

Step 3: Get the ID and media_url of each tweet in the dataset

SubStep 3.1: Get ID and Entities

```
tweetsIDAndEntities = tweetsWithImage[All, {"ID", "Entities"}]
```

Out[•]=

ID	Entities
1 123 484 461 377 757 190	\[LeftAssociation] …5 \[RightAssociation]
1 123 390 618 477 191 168	\[LeftAssociation] …5 \[RightAssociation]
1 123 381 570 029 670 400	\[LeftAssociation] …5 \[RightAssociation]
1 123 373 534 070 353 920	\[LeftAssociation] …5 \[RightAssociation]
1 123 364 585 128 353 792	\[LeftAssociation] …5 \[RightAssociation]
1 123 363 915 625 164 800	\[LeftAssociation] …5 \[RightAssociation]
1 123 361 630 765 686 784	\[LeftAssociation] …5 \[RightAssociation]
1 123 351 688 046 366 721	\[LeftAssociation] …5 \[RightAssociation]

Substep 3.2: Get ID and media_url(importing media_url) of the first row (tweet)

```
ExtractImage[entityrow_] := entityrow["media"][[1]]["media_url"]

oneTweet = First[tweetsIDAndEntities]; (* First gets the first element *)
URL : ExtractImage[oneTweet["Entities"]]
ID : oneTweet["ID"]
```

Out[•]= URL : http://pbs.twimg.com/media/D5ZmRSlWwAADN45.jpg

Out[•]= ID : 1 123 373 534 070 353 920

Substep 3.3: Get ID and media_url of all rows (tweets)

- Note: To apply functions to a specific column in a dataset (ds) use ***ds[All, {"columnname" -> function}]***. *All specifies that all rows will be affected.*

```
temp = tweetsIDAndEntities[All, {"Entities" → ExtractImage}];
temp[All, <|"TweetID" → "ID", "Image" → "Entities"|>]
(* Just if we want to change column names *)
```

Out[•]=

TweetID	Image
1 123 484 461 377 757 190	http://pbs.twimg.com/media/D5ZmRSlWwAADN45.jpg
1 123 390 618 477 191 168	http://pbs.twimg.com/media/D5ZmRSlWwAADN45.jpg
1 123 381 570 029 670 400	http://pbs.twimg.com/media/D5cNjEeXoAAmxI_.jpg
1 123 373 534 070 353 920	http://pbs.twimg.com/media/D5ZmRSlWwAADN45.jpg
1 123 364 585 128 353 792	http://pbs.twimg.com/media/D5b–IZ9VUAA5WD2.jpg
1 123 363 915 625 164 800	http://pbs.twimg.com/media/D5b9guUUEAAr_CF.jpg
1 123 361 630 765 686 784	http://pbs.twimg.com/media/D5b7cyTXoAA5EvU.jpg
1 123 351 688 046 366 721	http://pbs.twimg.com/media/D5byKDrXsAAL8IM.jpg

- Another way of doing it with pure functions (without the need of declaring a function)...

```
temp = tweetsWithImage[All, {"ID", #["Entities"]["media"][[1]]["media_url"] &}];
IDAndImageTweets = temp[All, <|"TweetID" → 1, "Image" → 2|>]
 (* Just if we want to add names in columns *)
```

Out[•]=

TweetID	Image
1 123 484 461 377 757 190	http://pbs.twimg.com/media/D5ZmRSlWwAADN45.jpg
1 123 390 618 477 191 168	http://pbs.twimg.com/media/D5ZmRSlWwAADN45.jpg
1 123 381 570 029 670 400	http://pbs.twimg.com/media/D5cNjEeXoAAmxI_.jpg
1 123 373 534 070 353 920	http://pbs.twimg.com/media/D5ZmRSlWwAADN45.jpg
1 123 364 585 128 353 792	http://pbs.twimg.com/media/D5b-IZ9VUAA5WD2.jpg
1 123 363 915 625 164 800	http://pbs.twimg.com/media/D5b9guUUEAAr_CF.jpg
1 123 361 630 765 686 784	http://pbs.twimg.com/media/D5b7cyTXoAA5EvU.jpg
1 123 351 688 046 366 721	http://pbs.twimg.com/media/D5byKDrXsAAL8IM.jpg

Substep 3.4: Erase duplicate media_urls

```
noDuplicatesTweets = IDAndImageTweets[DeleteDuplicatesBy["Image"]]
```

Out[•]=

TweetID	Image
1 123 484 461 377 757 190	http://pbs.twimg.com/media/D5ZmRSlWwAADN45.jpg
1 123 381 570 029 670 400	http://pbs.twimg.com/media/D5cNjEeXoAAmxI_.jpg
1 123 364 585 128 353 792	http://pbs.twimg.com/media/D5b-IZ9VUAA5WD2.jpg
1 123 363 915 625 164 800	http://pbs.twimg.com/media/D5b9guUUEAAr_CF.jpg
1 123 361 630 765 686 784	http://pbs.twimg.com/media/D5b7cyTXoAA5EvU.jpg
1 123 351 688 046 366 721	http://pbs.twimg.com/media/D5byKDrXsAAL8IM.jpg

Substep 3.5: Import images from media_urls

```
cleanTweets = noDuplicatesTweets[All, {"Image" → Import}]
```

Out[•]=

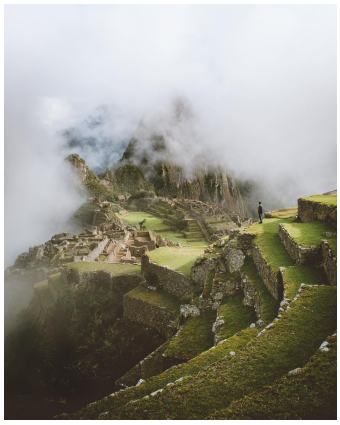
TweetID	Image
1 123 484 461 377 757 190	
1 123 381 570 029 670 400	
1 123 364 585 128 353 792	
1 123 363 915 625 164 800	
1 123 361 630 765 686 784	
1 123 351 688 046 366 721	

Step 4 [Optional]: Convert to list of associations (AKA Python dictionary) or to a list of rules (*[key->value, key->value, ...]*)

- **Note:** Normal turns a dataset into a list of associations (dictionaries in Python)

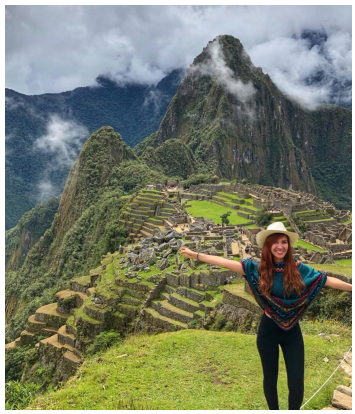
List of associations: *{ <| TweetID-> ..., Image->... |>, <| TweetID-> ..., Image->... |>, <| TweetID-> ..., Image->... |>, ... }*

```
Normal[cleanTweets]
```

Out[•]= { ⟨| TweetID → 1 123 484 461 377 757 190, Image →  |⟩ ,

⟨| TweetID → 1 123 381 570 029 670 400, Image → 

|⟩,

⟨| TweetID → 1 123 364 585 128 353 792, Image →  |⟩,

⟨| TweetID → 1 123 363 915 625 164 800, Image → 

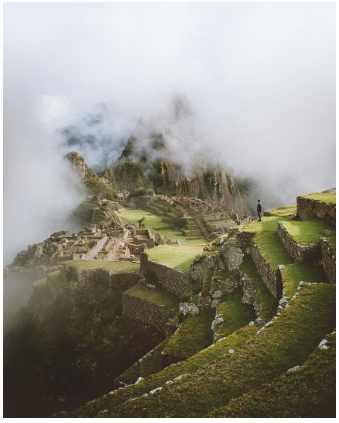
|⟩,

⟨| TweetID → 1 123 361 630 765 686 784, Image →  |⟩,

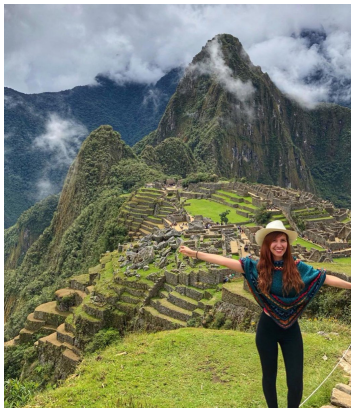
⟨| TweetID → 1 123 351 688 046 366 721, Image →  |⟩ }

List of rules : *{ ID->image, ID->image, ... }*

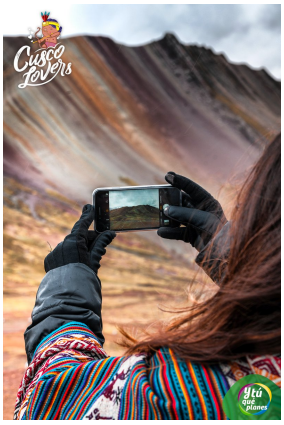
```
Function[list, list["TweetID"] → list["Image"]] /@ Normal[cleanTweets]
```

Out[•]= { 1 123 484 461 377 757 190 →  ,

1 123 381 570 029 670 400 →  ,

1 123 364 585 128 353 792 →  ,

1 123 363 915 625 164 800 →



,

1 123 361 630 765 686 784 →

,

1 123 351 688 046 366 721 →

}