Tweet from Automic ONE Autommation

Step 1: Create a Twitter Developer Account

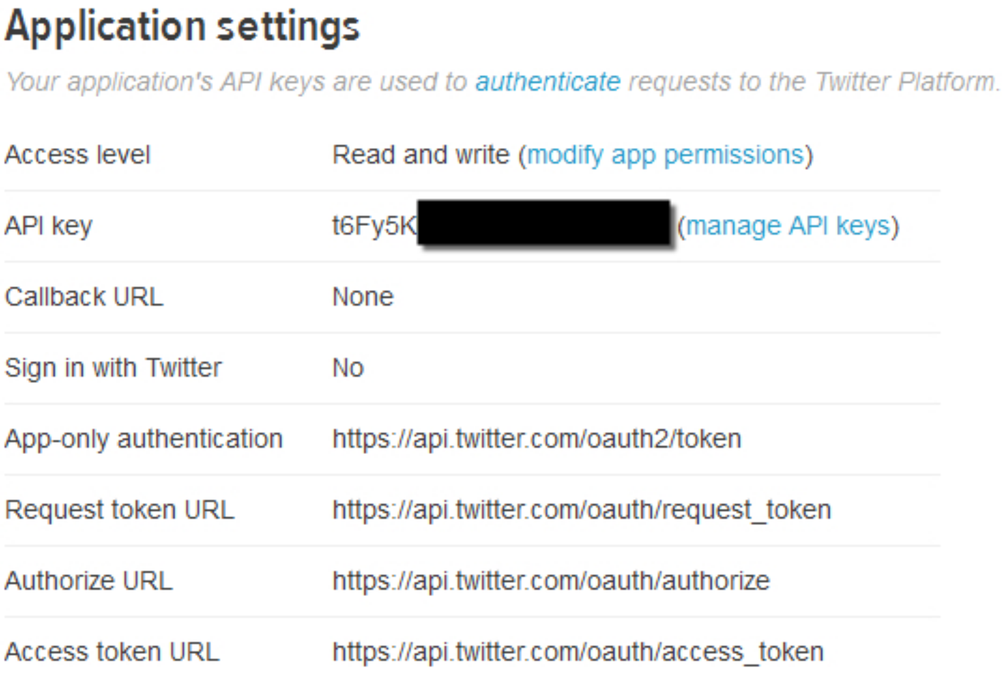
In order to setup the Webservice 3.0 Agent with Twitter, you'll need to create an account on the Twitter developer site and generate credentials for use with the Twitter API. To create a Twitter developer account follow these steps:

a) Go to https://dev.twitter.com/user/login and log in with your Twitter user name and password. If you do not yet have a Twitter account, click the **Sign up** link that appears under the **Username** field.

b) Link a mobile phone to your twitter Account – in order to setup the new App privileges with Read and Write.

c) Go to the Twitter applications page at https://dev.twitter.com/apps and click **Create a new application**.

d) Follow the on-screen instructions. For the application **Name**, **Description**, and **Website**, you can enter any text — you're simply generating credentials to use with this tutorial, rather than creating a real application.

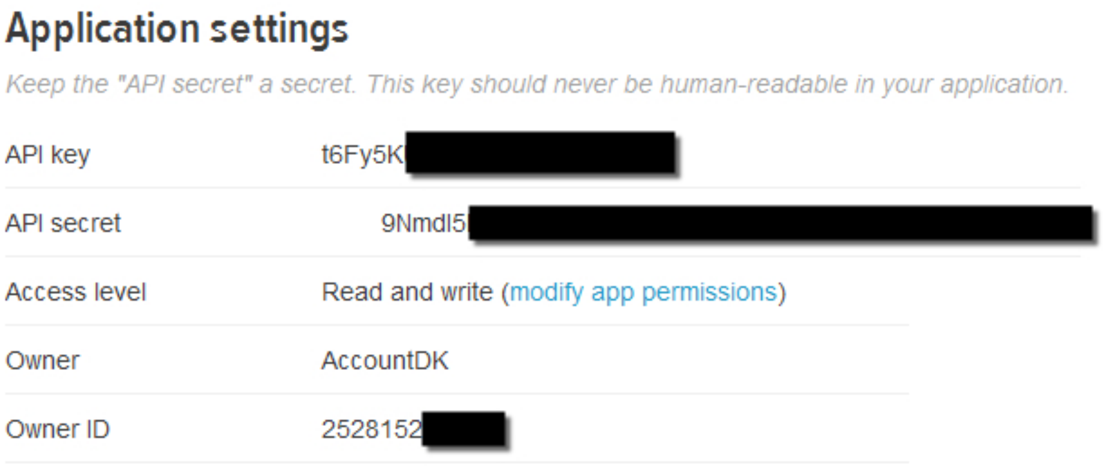


Application settings

Your application's API keys are used to authenticate requests to the Twitter Platform.

Access level	Read and write (modify app permissions)
API key	t6Fy5K[redacted] (manage API keys)
Callback URL	None
Sign in with Twitter	No
App-only authentication	https://api.twitter.com/oauth2/token
Request token URL	https://api.twitter.com/oauth/request_token
Authorize URL	https://api.twitter.com/oauth/authorize
Access token URL	https://api.twitter.com/oauth/access_token

e) On the details page for your new application, you'll see a **API key** and **API secret**. Make a note of these values; you'll need them later in the Connection object. You may want to store your credentials in a text file.

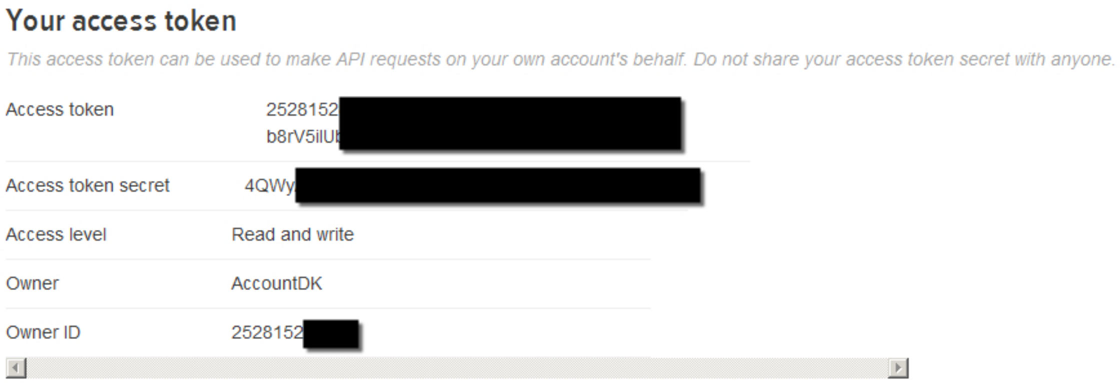


Application settings

Keep the "API secret" a secret. This key should never be human-readable in your application.

API key	t6Fy5K[redacted]
API secret	9Nmdl5[redacted]
Access level	Read and write (modify app permissions)
Owner	AccountDK
Owner ID	2528152[redacted]

f) At the bottom of the application details page, click **Create my access token**. Make a note of the **Access token** and **Access token secret** values that appear, or add them to the text file you created in the preceding step.



Your access token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

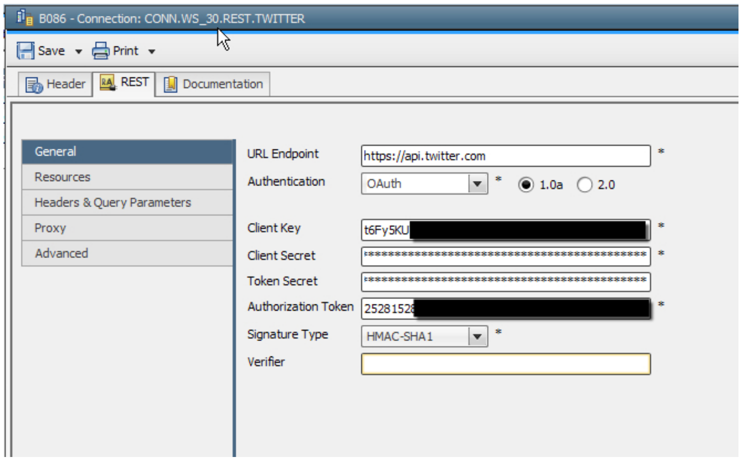
Access token	2528152 b8rV5il[redacted]
Access token secret	4QWy[redacted]
Access level	Read and write
Owner	AccountDK
Owner ID	2528152[redacted]

Step 2: Setup the Webservice 3.0 RA Agent

Nothing special here – standard install according to the documentation

Step 3: Create a Connection Object for Twitter

In order to setup the Webservice 3.0 Agent with Twitter, you'll need to setup a connection object using the OAuth method with data from Step 1:


B086 - Connection: CONN.WS_30.REST.TWITTER
Save
Print
Header
REST
Documentation
General
Resources
Headers & Query Parameters
Proxy
Advanced

URL Endpoint: https://api.twitter.com *

Authentication: OAuth * ● 1.0a ○ 2.0

Client Key: t6Fy5KU[redacted] *

Client Secret: ** *

Token Secret: **

Authorization Token: 2528152[redacted] *

Signature Type: HMAC-SHA1 *

Verifier:

Client Key = API key

Client Secret = API secret

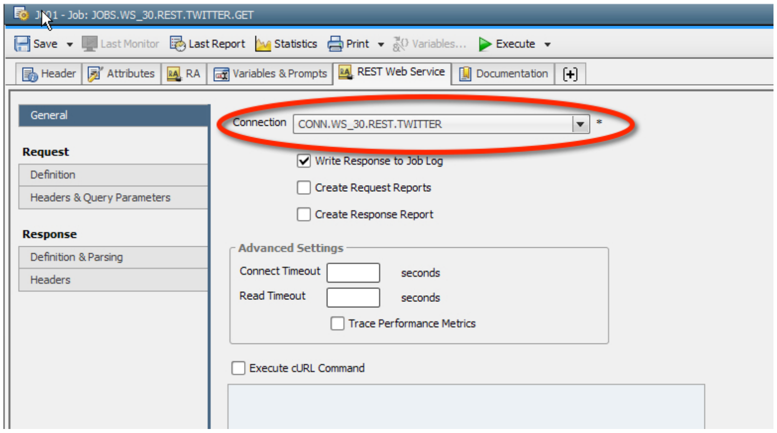
Authorization Token = Access token

Token Secret = Access token secret

Step 4: Create a REST Job to GET tweets

Once you have the connection object defined, you can setup your first Job – this one will GET tweets:

a) Select the connection object



Connection: CONN.WS_30.REST.TWITTER *

☑ Write Response to Job Log

□ Create Request Reports

□ Create Response Report

Advanced Settings

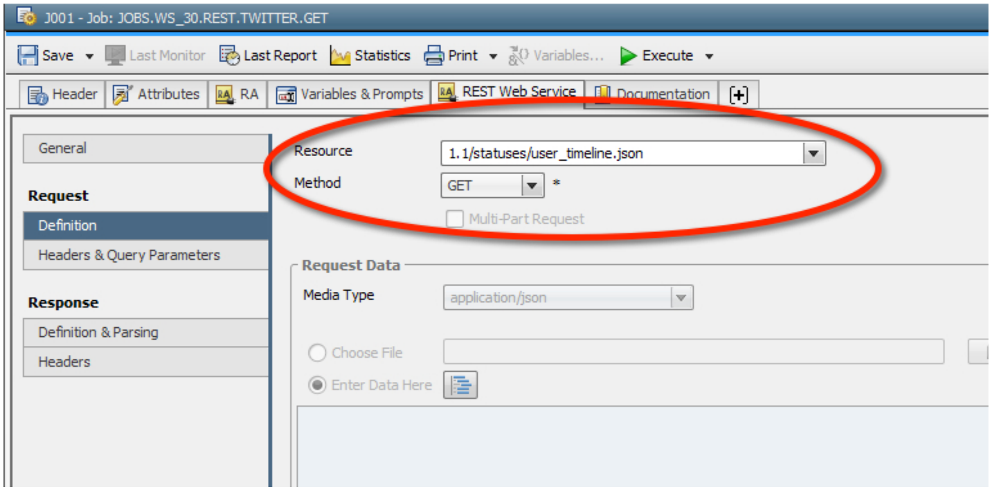
Connect Timeout: seconds

Read Timeout: seconds

□ Trace Performance Metrics

□ Execute cURL Command

b) Type “1.1/statuses/user_timeline.json” into the Resource field – it’s a dropdown, but still accept typing. Select GET in the Method, as you are getting tweets.

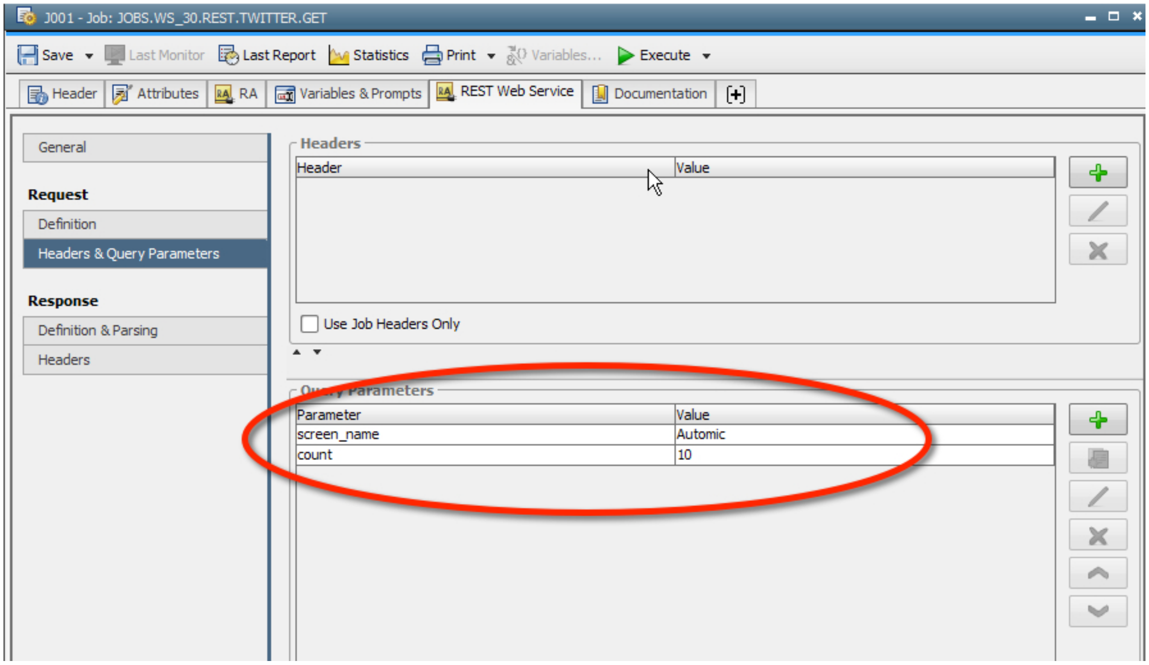


Resource: 1.1/statuses/user_timeline.json

Method: GET *

Media Type: application/json

c) Setup the parameters screen_name and count to bring back only a specific user and only a maximum number of tweets:

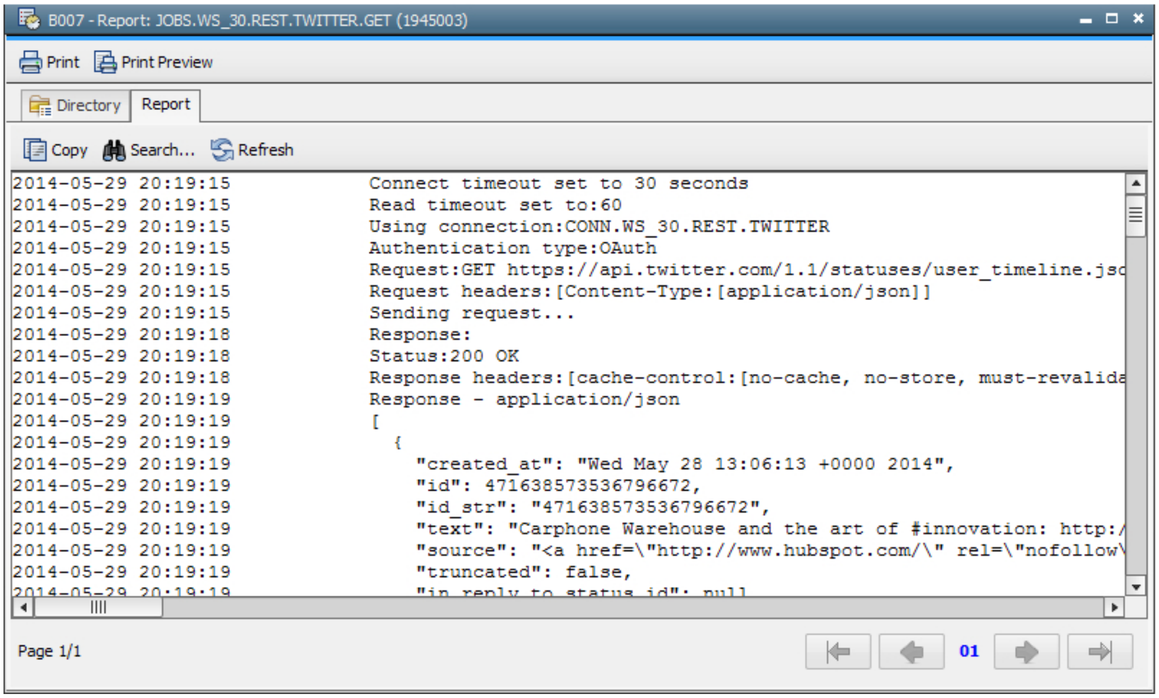


Header	Value

□ Use Job Headers Only

Parameter	Value
screen_name	Automic
count	10

d) Execute the Job – it will bring back the last 10 tweets form Automic



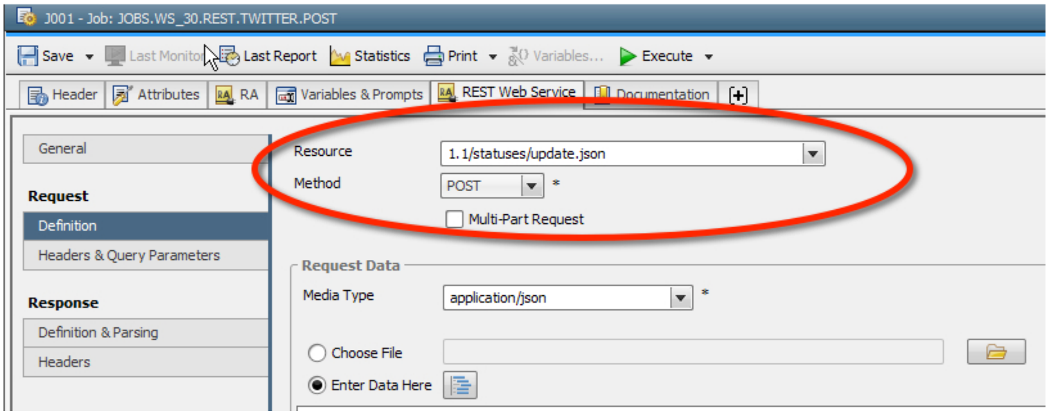
```
2014-05-29 20:19:15            Connect timeout set to 30 seconds
2014-05-29 20:19:15            Read timeout set to:60
2014-05-29 20:19:15            Using connection:CONN.WS_30.REST.TWITTER
2014-05-29 20:19:15            Authentication type:OAuth
2014-05-29 20:19:15            Request:GET https://api.twitter.com/1.1/statuses/user_timeline.jsc
2014-05-29 20:19:15            Request headers:[Content-Type:[application/json]]
2014-05-29 20:19:15            Sending request...
2014-05-29 20:19:18            Response:
2014-05-29 20:19:18            Status:200 OK
2014-05-29 20:19:18            Response headers:[cache-control:[no-cache, no-store, must-revalida
2014-05-29 20:19:19            Response - application/json
2014-05-29 20:19:19            [
2014-05-29 20:19:19              {
2014-05-29 20:19:19                "created_at": "Wed May 28 13:06:13 +0000 2014",
2014-05-29 20:19:19                "id": 471638573536796672,
2014-05-29 20:19:19                "id_str": "471638573536796672",
2014-05-29 20:19:19                "text": "Carphone Warehouse and the art of #innovation: http:/
2014-05-29 20:19:19                "source": "<a href=\"http://www.hubspot.com/\" rel=\"nofollow\
2014-05-29 20:19:19                "truncated": false,
2014-05-29 20:19:19                "in_reply_to_status_id": null
```

Step 4: Create a REST Job to POST tweets

Now that we know authentication works, it's time to POST tweets:

a) Duplicate the GET Job

b) Type "1.1/statuses/update.json" into the Resource field – it's a dropdown, but still accept typing. Select POST in the Method, as you are posting tweets.



Resource: 1.1/statuses/update.json

Method: POST *

□ Multi-Part Request

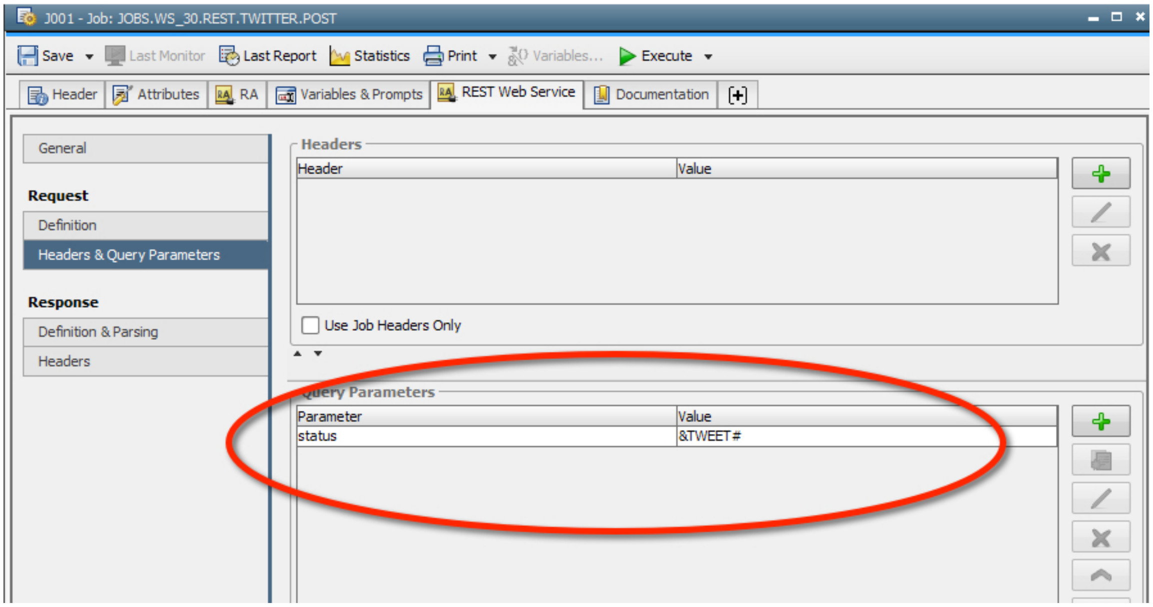
Request Data

Media Type: application/json *

○ Choose File

● Enter Data Here

c) This time the Query Parameters are slightly different:



Headers

Header	Value

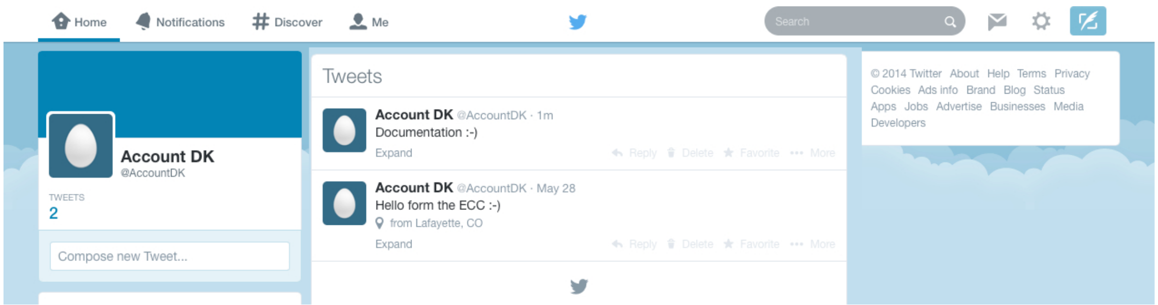
□ Use Job Headers Only

Query Parameters

Parameter	Value
status	&TWEET#

Notice that &TWEET# is a variable – I used a Promptset to make this work more dynamically and show better form the ECC

d) Execute the Job and check your tweets:


Home
Notifications
Discover
Me
Search
Account DK
@AccountDK
TWEETS
2
Compose new Tweet...
© 2014 Twitter About Help Terms Privacy
Cookies Ads info Brand Blog Status
Apps Jobs Advertise Businesses Media
Developers

Tweets

Account DK @AccountDK · 1m
Documentation :-)
Expand · Reply · Delete · Favorite · More

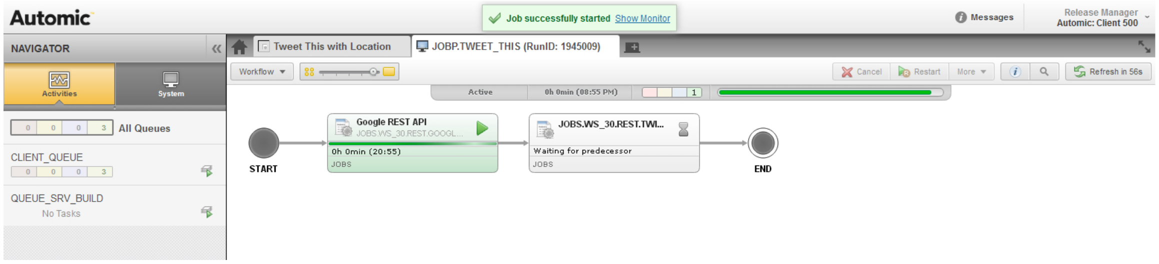
Account DK @AccountDK · May 28
Hello form the ECC :-)
from Lafayette, CO
Expand · Reply · Delete · Favorite · More

Step 5: Explore what else you can do

The Twitter REST API is well documents and you can play with all sorts of things:

https://dev.twitter.com/docs/api/1.1

In addition you can start combining things – in the export you also find a Workflow that combines Google GEO (Maps) to find the coordinates of any address and then passes that information on to a Twitter Job that tweets with location information (you need to enable “Add a location to my Tweets” in the “Security and privacy” settings – you get there by logging on to your “regular” twitter site)



Job successfully started Show Monitor

JOBP.TWEET_THIS (RunID: 1945009)

0	0	0	3	All Queues

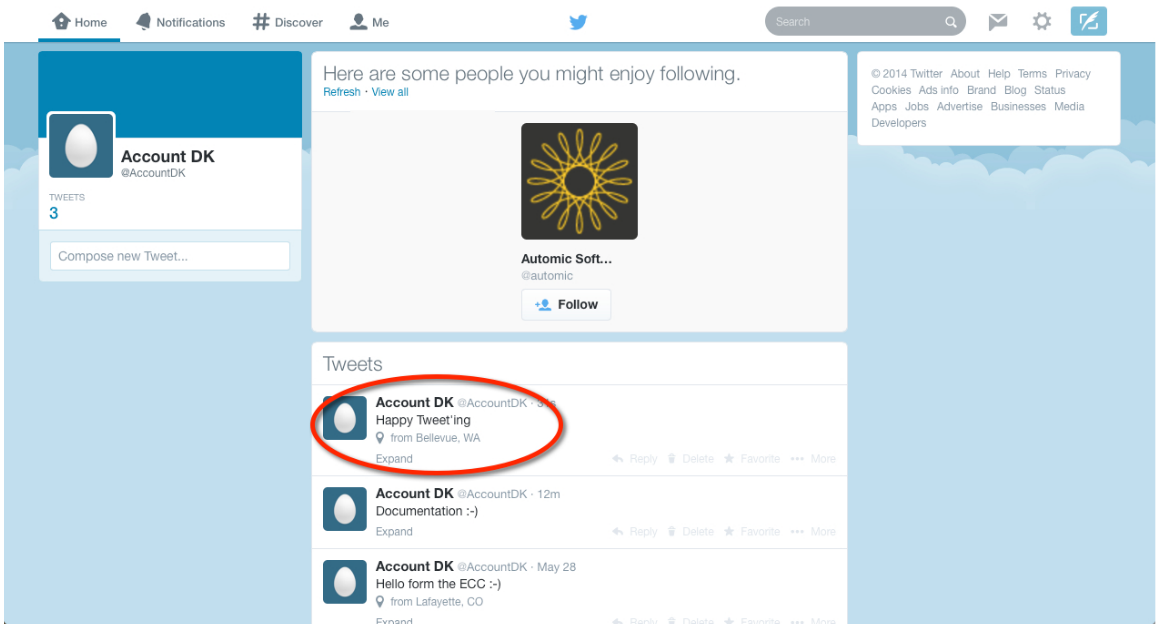
CLIENT_QUEUE 0 0 0 3

QUEUE_SRV_BUILD
No Tasks

Active 0h 0min (08:55 PM) 1

Google REST API
JOBS.WS_30.REST.GOOGL...
0h 0min (20:55)
JOBS

JOBS.WS_30.REST.TWI...
Waiting for predecessor
JOBS



Account DK
@AccountDK
3

Here are some people you might enjoy following.

Automatic Soft...
@automic

© 2014 Twitter About Help Terms Privacy Cookies Ads info Brand Blog Status Apps Jobs Advertise Businesses Media Developers

Tweets

Account DK @AccountDK · 34s
Happy Tweet'ing
from Bellevue, WA

Account DK @AccountDK · 12m
Documentation :-)

Account DK @AccountDK · May 28
Hello form the ECC :-)
from Lafayette, CO