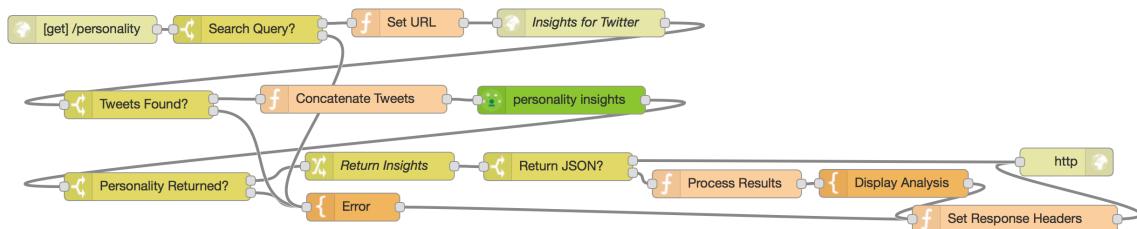
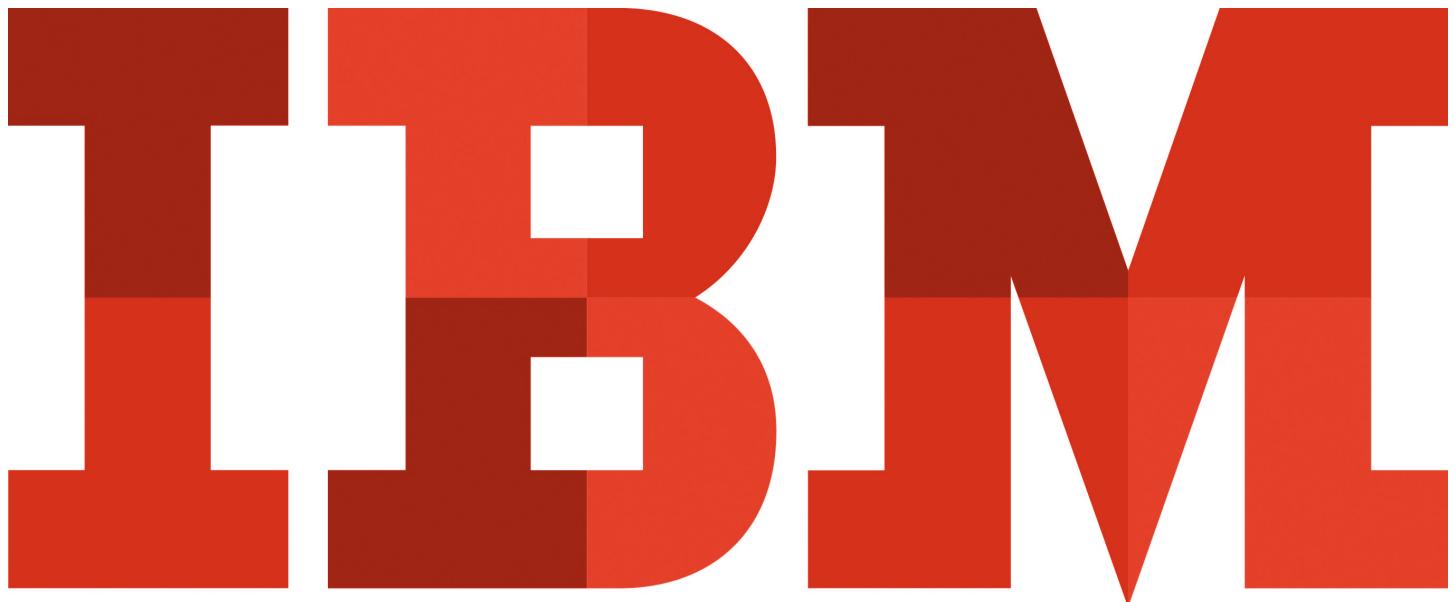


# Analyzing Twitter User's Personality in Node-RED

## Hands-On Lab

JeanCarl Bisson | [jbisson@us.ibm.com](mailto:jbisson@us.ibm.com) | [@dothewww](https://twitter.com/dothewww)



Search tweets from Twitter and gain insights using IBM Watsons Personality Insights



A digital copy of this lab and code snippets can be found at:  
<http://ibm.biz/node-red-twitter-personality>

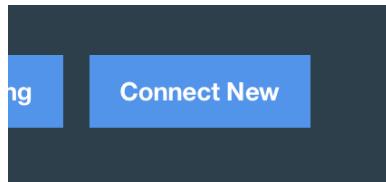


# Add Insights for Twitter service in IBM Bluemix

In this lab, we'll create a Node-RED application that searches Twitter for tweets using the Insights for Twitter Bluemix service, and then use the IBM Watson Personality Insights service to gain insight into how and why people think, act, and feel the way they do.

Our first step is to add the Insights for Twitter service and connect it to the Node-RED application.

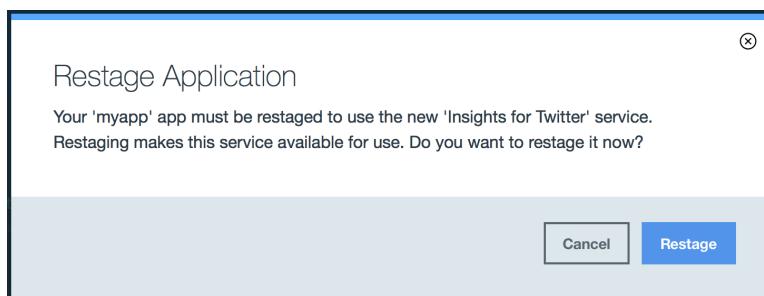
1. Go to the Connections tab under the application overview for your Node-RED application in the IBM Bluemix dashboard and click on **Connect New**.



2. Click on the **Insights for Twitter** tile under the Data & Analytics section. Click on **Create**.

A screenshot of the IBM Bluemix catalog. On the left, there is a sidebar with categories like All Categories, Boilerplates, Compute, Network, Storage, Data &amp; Analytics (which is selected and highlighted in blue), Watson, Internet of Things, APIs, DevOps, Security, and Application Services. The main area displays various service tiles. One of the tiles is for 'Insights for Twitter', which has a blue border around it, indicating it was selected. Other visible tiles include dashDB, DataWorks, Elasticsearch by Compose, Geospatial Analytics, IBM DataStage on Cloud, IBM DB2 on Cloud, IBM Graph, IBM Master Data Management on Cloud, MongoDB by Compose, PostgreSQL by Compose, and Predictive Analytics.

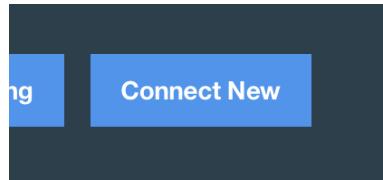
3. IBM Bluemix will prompt to restage the application. Click on **Restage**. The application will restart and include the new service credentials in the environment.



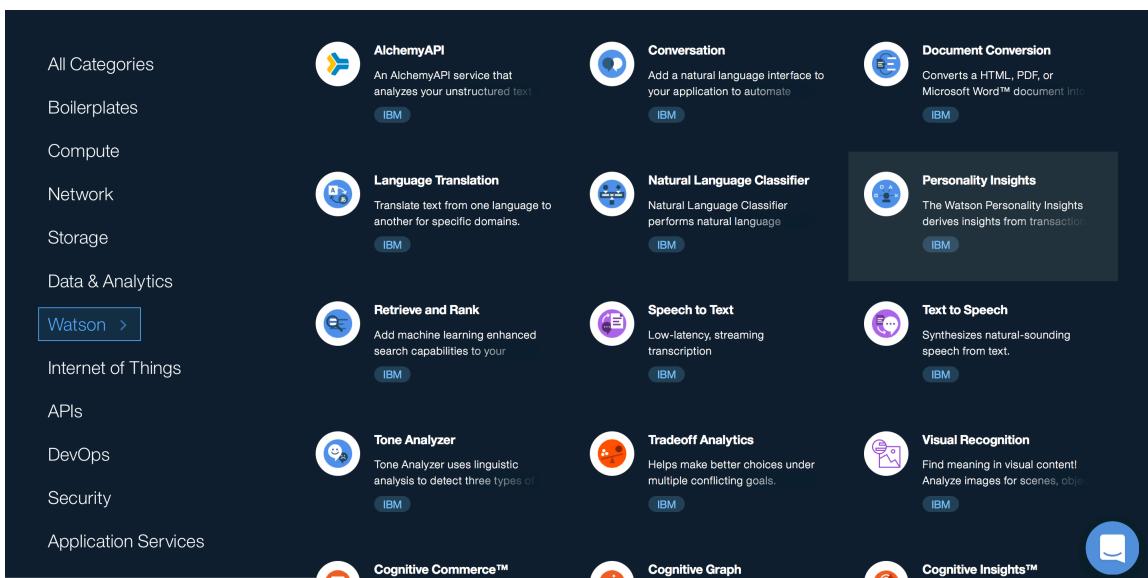
# Add Personality Insights service in IBM Bluemix

Next, let's add IBM Watson Personality Insights service and connect it to the Node-RED application.

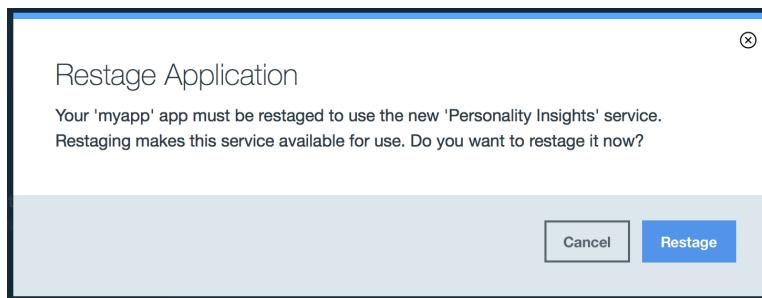
1. Go to the Connections tab under the application overview for your Node-RED application in the IBM Bluemix dashboard and click on **Connect New**.



2. Click on the **Personality Insights** tile under the Watson section. Click on **Create**.



3. IBM Bluemix will prompt to restage the application. Click on **Restage**. The application will restart and include the new service credentials in the environment.

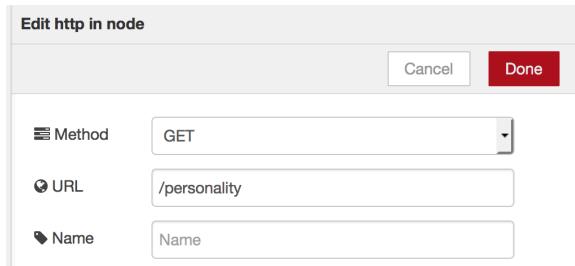


4. When the application has finished restaging, open the Node-RED Flow Editor. If you already have Node-RED open, refresh the page.

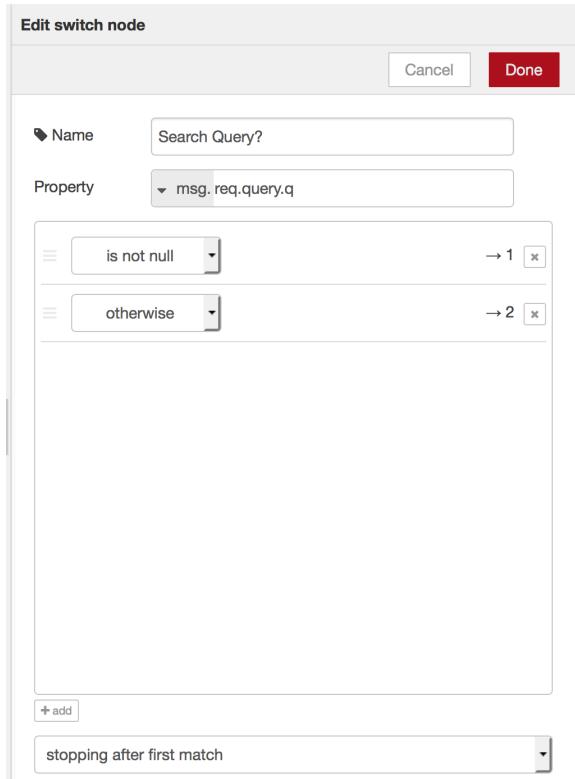
# Retrieve and Analyze Tweets

In this section, we will create a web URL endpoint that takes in a search query in the URL; use the Insights for Twitter service to search for tweets matching this criteria; concatenate the matching tweets together; use the IBM Watson Personality Insights service to gain insight into how and why people think, act, and feel the way they do; and then return the results in JSON and HTML formats.

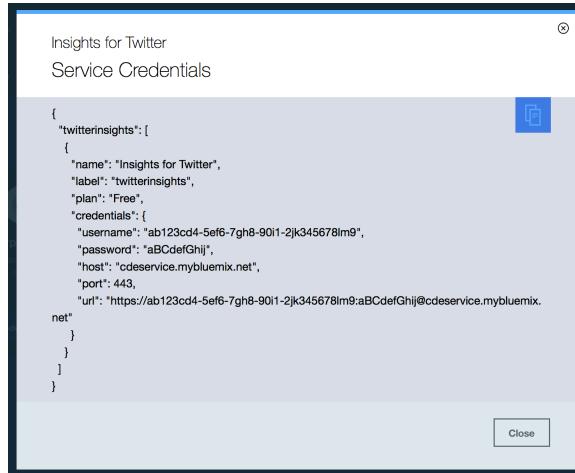
1. Add a  node as shown below to expose a web URL endpoint.



2. Checking for scenarios where errors might happen is a good programming habit to have. We should check whether the query parameter *q* is passed in. If not, we should display an error message. Add a  node as shown below. We'll connect the second output to an error message later on in step #18.



3. Return to the Connections tab in the application overview in the Bluemix console. Under the Insights for Twitter service tile, click on **View Credentials**. Copy the value in the URL property.



4. Add a node as shown below. This will set the request URL for the HTTP request node in the next step.

**Edit function node**

Name

Function

```
1 msg.url = 'https://cdeservice.mybluemix.net/api/v1/messages/search?size=500&q='+msg.payload.q;
2
3 return msg;
4
```



Get the code:  
ibm.biz/Bd457G

5. Add a node as shown below. This will make a HTTP request using the URL from the function node and return tweets that match the search criteria in *msg.payload*. Use the username and password values from step #3, and change the return value to return a parsed JSON object.

**Edit http request node**

Cancel
Done

Method

URL

Enable secure (SSL/TLS) connection

Use basic authentication

Username

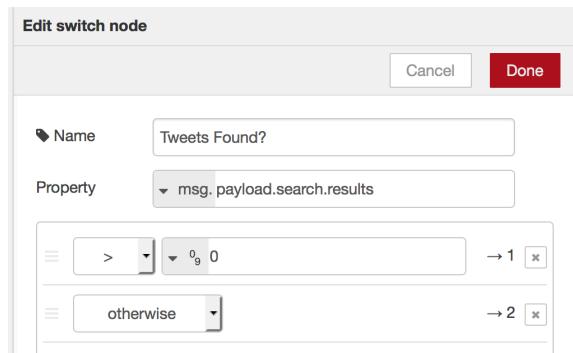
Password

Return

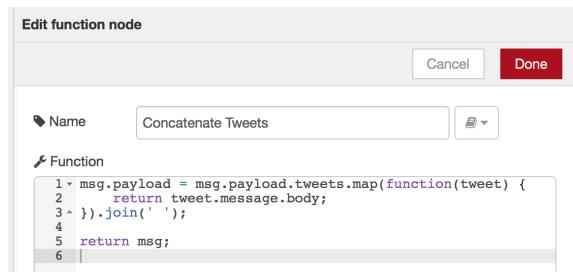
Name

Tip: If the JSON parse fails the fetched string is returned as-is.

6. What happens if there are no tweets found? Add a  node as shown below to check for this scenario. This switch node tests the number of results. If the number of tweets is greater than 0, we'll continue processing normally. Otherwise, we'll proceed to show an error message. We'll connect the second output to an error message later on in step #18.



7. Now that we have an array of tweets, we will concatenate the tweets together so that the IBM Watson Personality Insights service can analyze this content. Add a  node as shown below.

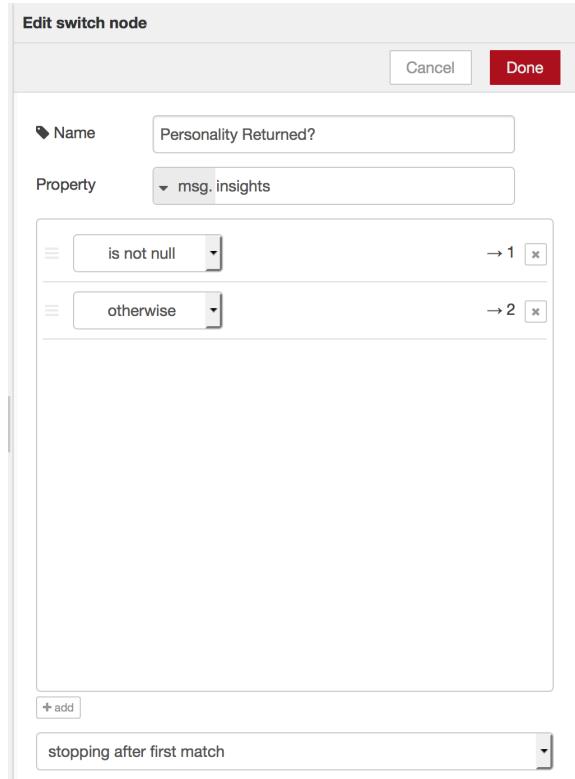


Get the code:  
[ibm.biz/Bd457n](http://ibm.biz/Bd457n)

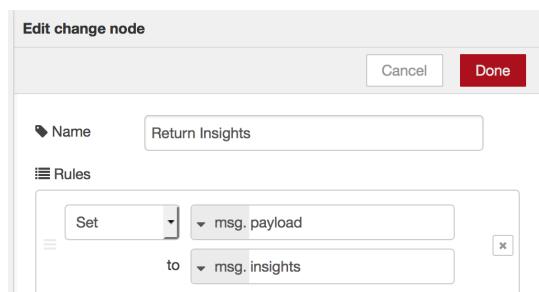
8. Add a  node as shown below. Because we added the IBM Watson Personality Insights service to our application, the credentials will be automatically read from the environment and used by the node. The IBM Watson Personality Insights service can analyze content in English or Spanish. Select English.



9. The IBM Watson Personality Insights service requires a minimum of a hundred words for a basic analysis. For statistically significant results, you need at least 3500 words and ideally 6000 words. Add a  switch node as shown below to check if a result is returned and proceed with the flow. Otherwise, we'll proceed to show an error message. We'll connect the second output to an error message later on in step #18.



10. The results from the IBM Watson Personality Insights node will reside in the *msg.insights*. Add a  change node to move the results to *msg.payload* as shown below.

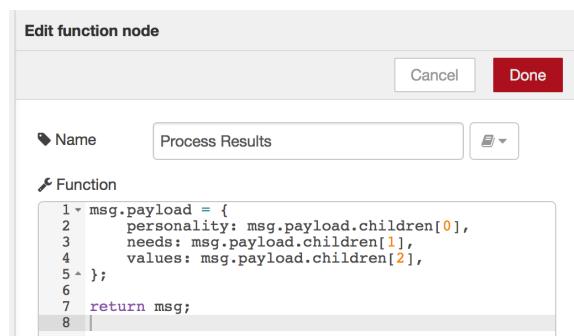


11. To make this application versatile, we'll split the flow so our application can return the results in two formats. The first option will be where the results are returned in JSON format, great for use in applications that can call the web endpoint and consume the JSON. The second option will be a webpage showing the results in a human readable report. Add a  node as shown below.



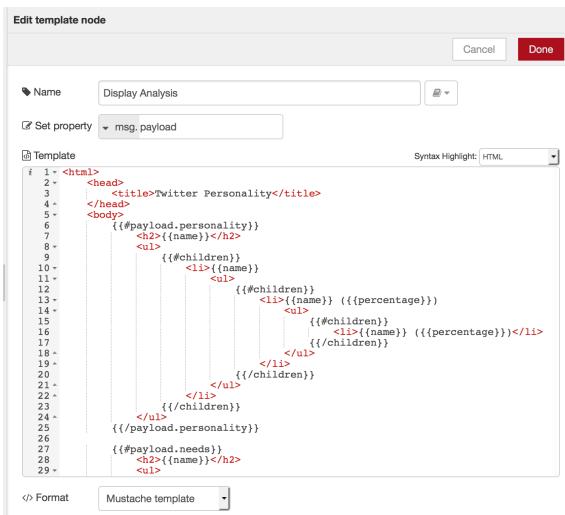
12. For the flow where the JSON should be returned, we can simply return the contents of *msg.payload*. Add a  to complete the flow.

13. For the flow where a webpage should be returned, we need to rearrange the results to simplify accessibility in the next step. Add a  as shown below.



Get the code:  
ibm.biz/Bd457b

14. Add a  node with the HTML in the file named 4-display-analysis.html. This HTML will display the results in a human readable format when viewed in a web browser.



```

1 <html>
2   <head>
3     <title>Twitter Personality</title>
4   </head>
5   <body>
6     {{#payload.personality}}
7       <h2>{{name}}</h2>
8       <ul>
9         {{#children}}
10          <li>{{name}}
11            <ul>
12              {{#children}}
13                <li>{{name}} ({{percentage}})</li>
14              {{/children}}
15            </ul>
16          </li>
17        {{/children}}
18      </ul>
19    {{/payload.personality}}
20    {{#payload.needs}}
21      <h2>{{name}}</h2>
22      <ul>
23        {{#children}}
24          <li>{{name}} ({{percentage}})</li>
25        {{/children}}
26      </ul>
27    {{/payload.needs}}
28    <ul>
29      {{#children}}

```



Get the code:  
ibm.biz/Bd457p

15. Add a  node to change the content-type as shown below. This will inform the browser to return the content as a webpage.



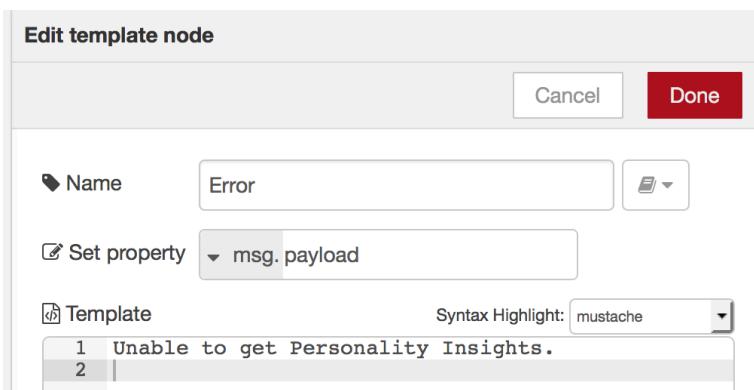
```

msg.headers = {'content-type': 'text/html'};
return msg;

```

16. Connect this node to the  node added in step #12.

17. The last step is to add a  node with the error message shown below.

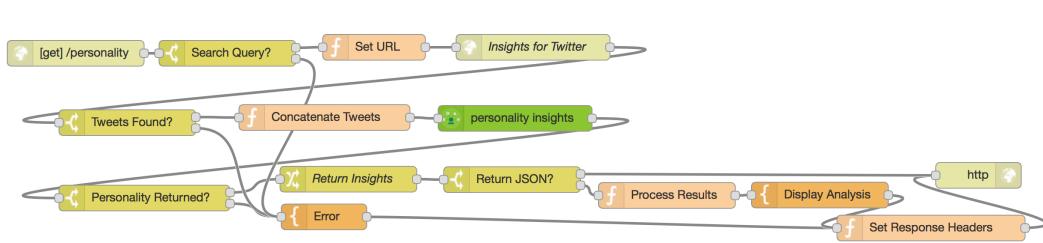


```

1 Unable to get Personality Insights.
2

```

18. Connect to the input of this node from the switch nodes that we created in steps #2, #6, and #9. Connect the output of this node to the node added in step #12. The completed flow is shown below.



Get the code:  
ibm.biz/Bd457A

19. Click on the red button in the upper-right corner of the screen to save and deploy your changes.

20. Congratulations! You've completed the programming part of this lab. Now let's analyze a user's tweets. Copy the following URL into a browser's address bar.

`http://<<MY-APP>>.mybluemix.net/personality?q=from:<<USERNAME>>`

- Replace <<MY-APP>> with the host of the Node-RED application you chose.
- Replace <<USERNAME>> with the Twitter user (with a sufficient number of tweets) to analyze

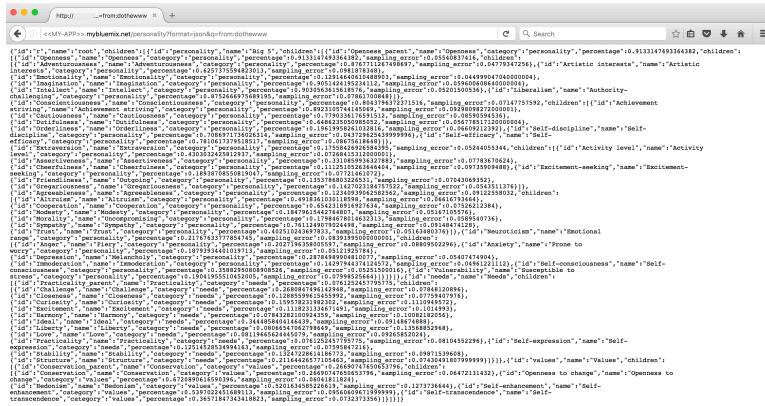
21. Depending on the tweets from the user you choose to analyze, you'll see the individuals' intrinsic personality characteristics, including Big Five, Needs and Values inferred from the tweets.

**Big 5**

- Openness
  - Openness (0.913314749336482)
  - Adventurousness (0.8767711467496697)
  - Curiosity (0.8921460619482303)
  - Emotionality (0.1291464061048800)
  - Imagination (0.9651242195234112)
  - Individuality (0.8921464061048800)
  - Independence (0.8723666975889195)
  - Authority-challenging (0.8723666975889195)
  - Achievement striving (0.892105744145069)
  - Curiosity (0.8921464061048800)
  - Dutifulness (0.648623509085052)
  - Orderliness (0.19619958261032816)
  - Stability (0.8921464061048800)
  - Self-efficacy (0.781061737918517)
  - Extraversion
    - Extraversion (0.4030312425812937)
    - Activity level (0.4030312425812937)
    - Assertiveness (0.8921464061048800)
    - cheerfulness (0.11125105263646404)
    - Exuberance-seeking (0.1002105263646404)
    - Gregariousness (0.142702184717522)
    - Aggressiveness (0.4918361030118598)
    - Arrogance (0.4918361030118598)
    - Boldness (0.8921464061048800)
    - Modesty (0.18739615442764807)
    - Sympathy (0.761124097934498)
    - Trust (0.4025102459793533)
  - Extraversion (0.4030312425812937)
  - Fiery (0.2027196358005597)
  - Fun-loving (0.8921464061048800)
  - Mirthful (0.28744989948109772)
  - Immaturity (0.1482944712741872)
  - Self-centered (0.8921464061048800)
  - Love (0.08119656264445079)
  - Pragmatic (0.0761252457795775)
- Needs
  - Practically (0.0761252457795775)
    - Challenge (0.269847496142948)
    - Curiosity (0.1597821196230)
    - Contentment (0.1597821196230)
    - Excitement (0.118213334671491)
    - Happiness (0.1597821196230)
    - Ideal (0.3444858040146439)
    - Likeable (0.1597821196230)
    - Love (0.08119656264445079)
    - Pragmatic (0.0761252457795775)

22. To see the JSON representation of the results, insert `format=json` in the URL query string:

<http://<<MY-APP>>.mybluemix.net/personality?format=json&q=from:<<USERNAME>>>



The screenshot shows a browser window with the URL `http://<<MY-APP>>.mybluemix.net/personality?format=json&q=from:<<USERNAME>>`. The page displays a large JSON object representing a user's personality traits. The structure includes various categories like Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism, each with multiple facets and their respective percentages and sampling errors.

```
{
  "id": "1",
  "name": "root",
  "children": [
    {
      "id": "personality",
      "name": "Big Five",
      "children": [
        {
          "id": "Openness",
          "name": "Openness",
          "percentage": 0.913314749344682,
          "sampling_error": 0.0594857414,
          "children": [
            {
              "id": "Adventurousness",
              "name": "Adventurousness",
              "percentage": 0.167671136784987,
              "sampling_error": 0.047794742156,
              "children": [
                {
                  "id": "Artistic",
                  "name": "Artistic",
                  "percentage": 0.1514464448854,
                  "sampling_error": 0.0452524448854,
                  "children": [
                    {
                      "id": "Dramatic",
                      "name": "Dramatic",
                      "percentage": 0.1514464448854,
                      "sampling_error": 0.0454860084800001
                    }
                  ]
                },
                {
                  "id": "Challenging",
                  "name": "Challenging",
                  "percentage": 0.1514464448854,
                  "sampling_error": 0.0478170084811
                }
              ]
            },
            {
              "id": "Achievement striving",
              "name": "Achievement striving",
              "percentage": 0.092310274414569,
              "sampling_error": 0.01471759394,
              "children": [
                {
                  "id": "Achievement",
                  "name": "Achievement",
                  "percentage": 0.092310274414569,
                  "sampling_error": 0.01471759394
                }
              ]
            }
          ]
        },
        {
          "id": "Conscientiousness",
          "name": "Conscientiousness",
          "percentage": 0.1514464448854,
          "sampling_error": 0.0477881212050001,
          "children": [
            {
              "id": "Self-discipline",
              "name": "Self-discipline",
              "percentage": 0.1514464448854,
              "sampling_error": 0.0477881212050001,
              "children": [
                {
                  "id": "Dutifulness",
                  "name": "Dutifulness",
                  "percentage": 0.1514464448854,
                  "sampling_error": 0.0477881212050001,
                  "children": [
                    {
                      "id": "Obedience",
                      "name": "Obedience",
                      "percentage": 0.1514464448854,
                      "sampling_error": 0.0477881212050001
                    }
                  ]
                },
                {
                  "id": "Cheerfulness",
                  "name": "Cheerfulness",
                  "percentage": 0.1514464448854,
                  "sampling_error": 0.0477881212050001
                }
              ]
            },
            {
              "id": "Agreeableness",
              "name": "Agreeableness",
              "percentage": 0.1514464448854,
              "sampling_error": 0.0477881212050001,
              "children": [
                {
                  "id": "Cooperation",
                  "name": "Cooperation",
                  "percentage": 0.1514464448854,
                  "sampling_error": 0.0477881212050001
                }
              ]
            }
          ]
        },
        {
          "id": "Extraversion",
          "name": "Extraversion",
          "percentage": 0.1514464448854,
          "sampling_error": 0.0477881212050001,
          "children": [
            {
              "id": "Activity level",
              "name": "Activity level",
              "percentage": 0.1514464448854,
              "sampling_error": 0.0477881212050001
            }
          ]
        }
      ]
    }
  ]
}
```

23. You can also modify the query parameter `q` to search for a search term instead of tweets by a specific user. Use the respective URLs:

<http://<<MY-APP>>.mybluemix.net/personality?q=<<SEARCH-TERM>>>

<http://<<MY-APP>>.mybluemix.net/personality?format=json&q=<<SEARCH-TERM>>>