The Tone Analyzer

The first thing we need to do is create the actual Tone Analyzer. Start by registering an IBM Cloud account if you don't already have one. (No credit card needed.)

Next, we're going to create a Tone Analyzer project. Go to the Tone Analyzer page on IBM's website and click Get started free. Log in with your cloud account if necessary.

You should now be on a page to create the new Tone Analyzer project , make sure "Lite" is selected from the drop down menu, and click "Create Project".

From the manage page , click on show credentials and copy the apikey and url values.

Your Username would be “apikey” and password the value of apikey showed in the credentials

To Test Out The Tool We Would Use The Node Project.

Start With A Simple Node Project

We're going to create a basic node project that uses the Tone Analyzer on a string. On your computer, make a folder for our sample project.

Start by making a basic node project. In the terminal:

```
npm init
```

For the package name, enter "tones"(or anything you wish)). The version is "1.0.0". Press enter for the description.The entry point is "index.js". For everything else, just press enter.

Our project now consists of the package.json file. But let's also create the entry point, index.js file, and also a config.js file to store our credentials. In your terminal.

```
touch index.js
touch config.js
```

Open the "config.js" file and enter the following code to it:

```
var config =
        {
        username: "",
        password: ""
        };

module.exports = config;
```

Fill in the username and password that we got from the previous section. Save and close the file.

Watson Has His Own Node Library, And Some Sample Code

So we have a basic node application, and our Tone Analyzer credentials are stored in our config file, but we need to write some code that uses Watson's Tone Analyzing power. Conveniently, Watson has his own Node library that we're going to use. Install it with the following command.

```
npm install --save watson-developer-cloud
```

Preparations are complete, so we can start coding. Before that, though, notice the API reference on IBM's website. It has some useful code that we're going to use

In our index.js file, add the following code:

```
// Use our Watson library.
var ToneAnalyzerV3 = require('watson-developer-cloud/tone-analyzer/v3');

// Require our config variables.
var config = require('./config');

// The text that we want to analyze the tone of.
var text = "My son has high fever since last week.\" No medicines are
working from him, since yesterday he is facing difficulty in breathing.\"
I don't have more money to buy any medicines and we live in a remote
village area , so no proper medical facility available. \"I kindly request
the authorities to provide for my child for I fear if he does not receive
proper medication or treatment soon, he may die.\"";
// Turn our text into valid json.
var input = { "text": text };

// The format that the tone analyzer needs.
var params =
        {
        'tone_input': input,
        'content_type': 'application/json'
        };

// Initialize the Tone Analyzer by giving it our credentials.
var tone_analyzer = new ToneAnalyzerV3(
        {
        username: config.username,
        password: config.password,
        version_date: '2017-09-21'
        });

// Use our Tone Analyzer variable to analyze the tone.
tone_analyzer.tone(params, function(error, response)
        {
        // There's an error.
        if (error)
                {
```

```
38                  console.log('Error:', error);
39                  }
40          // No error, we got our tone result.
41          else
42                  {
43                  // The tone of the text, as determined by watson.
44                  var tone = JSON.stringify(response, null, 2)
45
46                  // Output Watson's tone analysis to the console.
                    console.log("The tone analysis for \'" + text + "\'
  is:\n");
                    console.log(tone);
                    }
            });
```

The code is pretty basic. We use our Watson library and our config file. We format that text as valid json. We then create the tone analyzer, and use it to analyze our text.

Test It Out

Our project is all set, now all we need to do is test it out. To do so, in your terminal:

```
node index.js
```

The output looks something like this:

```
1  {
 2      "document_tone":{
 3          "tones":[
 4              {
 5                  "score":0.789226,
 6                  "tone_id":"fear",
 7                  "tone_name":"fear"
 8              },
 9              {
10                  "score":0.614792 ,
11                  "tone_id":"Tentative",
12                  "tone_name":"Tentative"
13              },
14              {
15                  "score":0.53035,
16                  "tone_id":"sadness",
17                  "tone_name":"sadness"
18              }
19          ]
20      },
21      "sentences_tone":[
22          {
23              "sentence_id":0,
24              "text":"Since Yesterday".
25 .",
```

```
            "tones":[
                {
                    "score":0.588614,
                    "tone_id":"sadness",
                    "tone_name":"sadness"
                }
            ]
        },
        {
            "sentence_id":1,
            "text":"“I kindly request the authorities to provide for my
child for I fear if he does not receive proper medication or treatment
soon “, he may die .\"",
            "tones":[
                {
                    "score":0.687768,
                    "tone_id":"Tentative",
                    "tone_name":"Tentative"
                }
            ]
        }
    ]
}
```


You can see that Watson analyzes the overall text, as well as each individual sentence. It gives a score on what percentage (in decimal form, obviously) the text tone is.