

Smart Bridge Internship

Project Report

on

AI Powered News Search App

by

Vishal Vishwakarma

Presidency University
Benagluru, India

Table of Contents

1. Introduction.....	3
a. Overview	
b. Purpose	
2. Literature Survey.....	4
a. Existing Problem	
b. Proposed Solution	
3. Theoretical Analysis.....	5
a. Block Diagram	
b. Hardware/Software Designing	
4. Experimental Investigations.....	24
5. Flowchart.....	24
6. Result.....	25
7. Advantages and Disadvantages.....	26
8. Applications.....	26
9. Conclusion.....	26
10. Future Scope.....	27
11. Bibliography.....	27
Appendix.....	28
a. Source code	

Introduction

1. Overview

The web is home to massive amounts of data, with more being created every day. Organizations can harness this constant stream of information to gain understanding, plan strategies, and find opportunities. Enriched news data can help your application make dynamic connections across current events faster.

2 .Purpose

The purpose of this project is to develop a web application that meets our need to find the most relevant and recent new articles worldwide and update them regularly.

Since the discovery service is integrated with Slack, it gives a bot to search news with a keyword. In addition, the web application also analyzes the sentiment present in this news article and extracts key words and concepts to make it easier for the user to decipher what is important and what is not.

Literature Survey

a. Existing Problem

All current news applications can be confusing for the average person, with multiple functions and exaggerated design, these applications still do not meet the demand of the latest news users and often get results from the past days, weeks and months, which confuses the user only.

Also, there is no way in these apps to know what the approximate feeling of the audience is regarding the article or news topic, which makes it less interactive and easy to use.

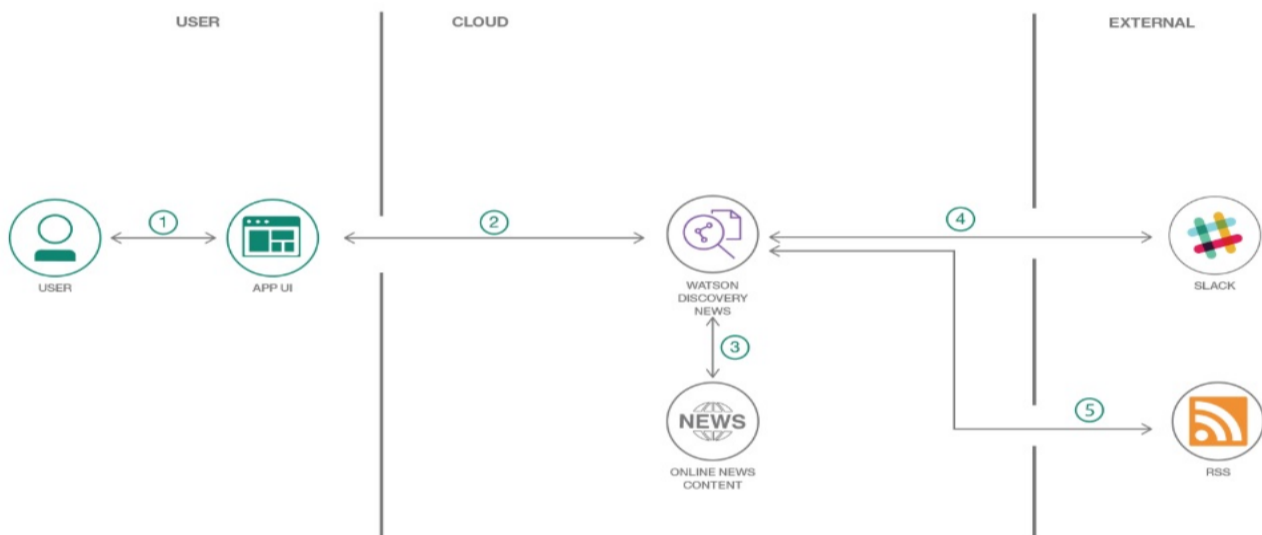
b. Proposed Solution

Discovery service available in the IBM cloud, creating a web app to get the latest news results is fast and hassle-free. When integrated with Red Node Flow, the IBM Discovery Service can create a simple, engaging, organized user interface that provides users with relevant news articles as Discovery Service continuously crawls the web for the latest news to provide.

By adding emotional analysis, we make the user interface more interactive and easier to understand.

Theoretical Analysis

a. Block Diagram



The above block diagram tells user interacts with the user interface created through the red node application that investigates a relevant topic about which he would like to receive news.

The user interface integrated with the discovery service will use the service to present the most recent and relevant news articles, corresponding to what the user searched for.

Since slack is also integrated with discovery, we can use our own bot and ask it to search for news by typing in the keyword and the bot searches for news articles related to it.

b. Software Designing

The following Task are required to make the complete application with slack bot integration.

Project Tasks

1. Creating and deploying Watson discovery news app locally.
2. Integrating Slack-bot with Watson Discovery.
3. Creating node-red user Interface.
4. Integrating node-red UI with Watson Discovery.

Setting up the development Environment

1. Installing Git

Install Git on Windows. The most official build is available for download on the Git website. Just go to <https://git-scm.com/download/win> and the download will start automatically. Note that this is a project called Git for Windows, which is separate from Git itself.

2. Installing Node.js

Introduction

Node.js is a run-time environment which includes everything you need to execute a program written in JavaScript. It's used for running scripts on the server to render content before it is delivered to a web browser.

2.1.1 Download Node.js Installer

In a web browser, navigate to <https://nodejs.org/en/download/>

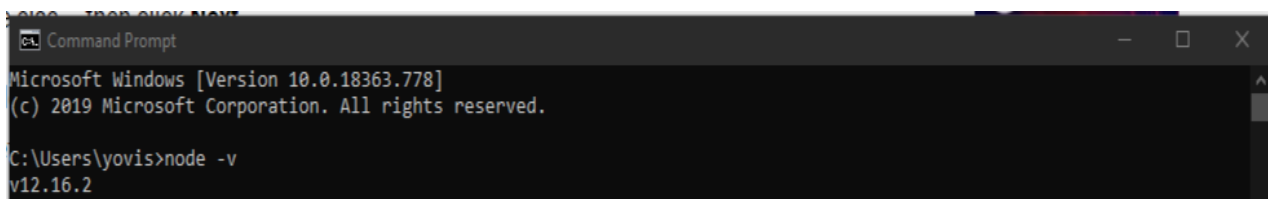
The installer will prompt you for the installation location. Leave the default location, unless you have a specific need to install it somewhere else – then click Next.

Verify Installation

Open a command prompt (or PowerShell), and enter the following:

```
1 node -v
```

The system should display the Node.js version installed on your system



```
Microsoft Windows [Version 10.0.18363.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\yovis>node -v
v12.16.2
```

3. Installing Node-RED

Introduction

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

Installing Node-RED as a global module adds the command node-red to your system path. Execute the following at the command prompt:

```
1 npm install -g --unsafe-perm node-red
```

Adding A collection of Node-RED nodes for IBM Watson services:

```
1 npm install node-red-node-watson
```

fig 2 shows the nodes that will be added

Adding node-red-dashboard module:

This module provides a set of nodes in Node-RED to create a live data dashboard quickly.

```
1 npm install node-red-dashboard
```

fig 1 shows the nodes that will be added



Task 1 Creating and deploying Watson discovery news app locally.

1.1 Creating Watson Discovery service on IBM cloud.

- Pre-requisites : IBM cloud account.
- About Watson Discovery: IBM Watson Discovery, we can ingest, normalize, enrich, and search your unstructured data (JSON, HTML, PDF, Word, and more) with speed and accuracy. It packages core Watson APIs such as Natural Language Understanding and Document Conversion along with UI tools that enable you to easily upload, enrich, and index large collections of private or public data.

Step 1: Login to IBM cloud account.

Step 2: Click 'Create resource' on your IBM Cloud dashboard.

Step 3: Search the catalog for Discovery.

Step 4: Click Discovery to launch the create panel.

Step 5: From the panel, enter a unique name, a region and resource group, and a plan type (select the default **lite** plan). Click **Create** to create and enable your service..

The next step is to configure your Watson Discovery service.

IBM Cloud

Search resources and offerings...

Catalog Docs Support Manage Richard Hagarty's Account

View all

Discovery

Lite • IBM

Add a cognitive search and content analytics engine to applications to identify patterns, trends and actionable insights that drive better decision-making. Securely unify structured and unstructured data with pre-enriched content, and use a simplified query language to eliminate the need for manual filtering of results.

[View Docs](#) [View API Docs](#) [Terms](#)

AUTHOR IBM
PUBLISHED 05/07/2019
TYPE Service

Service name:
Discovery-ur

Choose a region/location to deploy in:
Dallas

Select a resource group:
default

Tags:
Examples: env:dev, version-1

Features

- **Rapid results**
Spend less time struggling with your data. Automated ingestion and integrated natural language processing in a fully managed cloud service removes the complexity from dealing with natural language content.
- **Domain Intelligence**
Adapt Discovery's understanding of your corpus with integrated machine learning to find the most relevant answers. Teach Discovery to apply knowledge of unique entities and relations in your industry or organization with Watson Knowledge Studio.
- **AI ready for business**
Uncover deep connections throughout your data by using advanced AI functions out of the box, such as natural language queries, passage retrieval, relevancy training, relationship graphs and anomaly
- **Cloud Native, Enterprise Grade and Secure**
Take advantage of enterprise-grade security and support, such as data isolation, end to end encryption, GDPR compliance and support for non-regulated PII data, built to scale on the cloud.

Need Help?
[Contact IBM Cloud Support](#)

Add to estimate Create

1.2 Configuring Watson Discovery service on IBM cloud.

Step 1: Find the Discovery service in your IBM Cloud Dashboard.

Step 2: Click on the service and then click **Launch tool**

The Watson Discovery News data collection is already associated with your service. You'll use this collection as the data source for your app. To access the collection, you must find the **COLLECTION_ID** and **ENVIRONMENT_ID**. To find these values:

Step 3: Click on the collection from the Manage Data panel. In this case, it is named Watson Discovery News.

Step 4: Click on the drop-down icon located in the top right corner of the panel.

IBM Watson Discovery

Cookie Preferences Instance: Discovery-mz

Watson Discovery News (English)

15,711,550 documents

About Watson Discovery News

Watson Discovery News is updated continuously with new articles.

Language	New articles per day
English	300,000
Spanish	60,000
German	40,000
Japanese	17,000
Korean	10,000

The news sources vary by language, so the query results for each collection will not be identical.

Added 5 enrichments to your data

Entity Extraction

United States (1057k) | Twitter (874k) | Facebook (863k) | U.S. (854k) | official (774k)

Sentiment Analysis

51% positive | 3% neutral | 45% negative

Concept Tagging

United States (1489k) | Stock (1128k) | Stock market (1128k) | English-language films (933k) | Share (700k)

Category Classification

law, govt and politics...

Now you're ready to query!

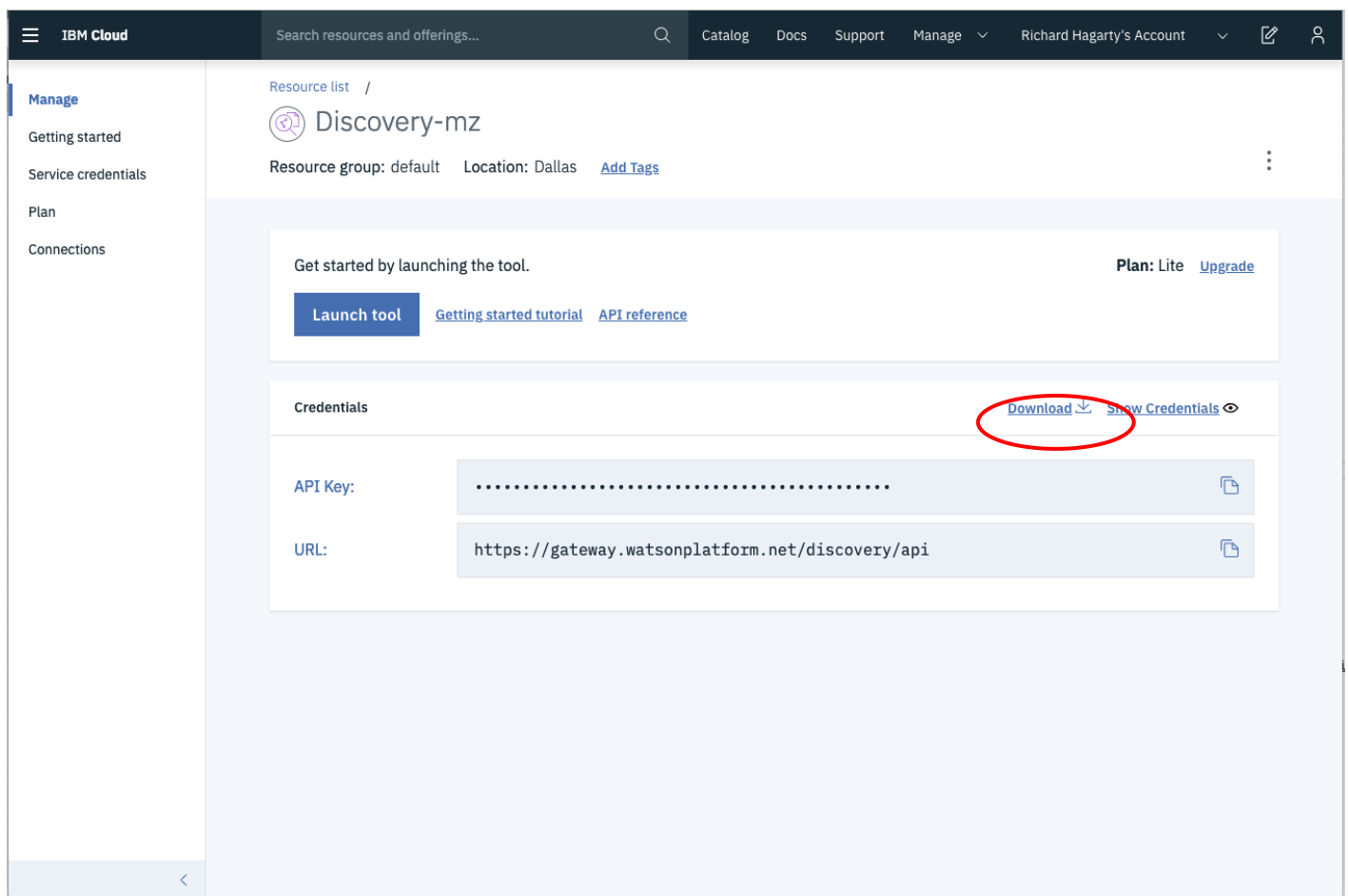
- Company acquisitions relating to artificial intelligence [Run](#)
- Top 20 recently acquired artificial intelligence companies [Run](#)
- Top 10 companies with positive sentiment [Run](#)
- 50 most mentioned people in the Tech industry [Run](#)
- Sentiment of articles about IBM over the last 60 days [Run](#)

1.3 Saving Watson Discovery credentials.

you'll need to add the Watson Discovery credentials to the .env file Later.

Step 1: Locate the service credentials listed on the home page of your Discovery service.

Step 2: click on the download link and save the file.

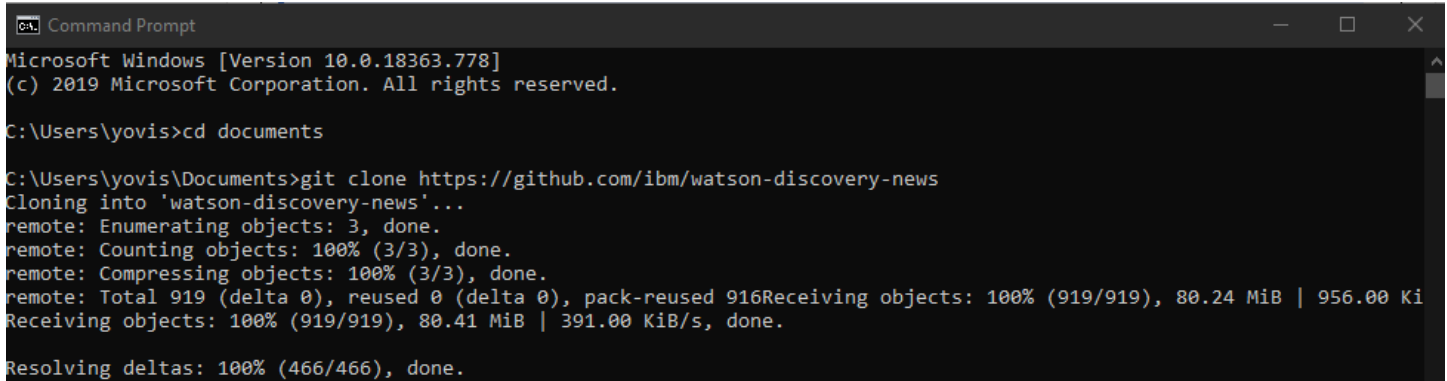


1.4 Deploying Watson-discovery-news App locally.

Note: Since deploying App on IBM cloud is not possible with Lite account so we deploy locally on your machines.

Step 1: Use the following command to clone the Watson-discovery-news GitHub repository.

```
1 git clone https://github.com/ibm/watson-discovery-news
```



```
Command Prompt
Microsoft Windows [Version 10.0.18363.778]
(c) 2019 Microsoft Corporation. All rights reserved.

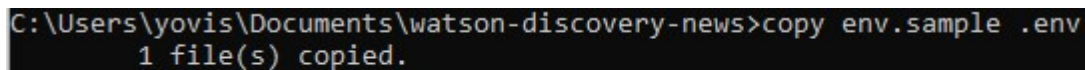
C:\Users\yovis>cd documents

C:\Users\yovis\Documents>git clone https://github.com/ibm/watson-discovery-news
Cloning into 'watson-discovery-news'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 919 (delta 0), reused 0 (delta 0), pack-reused 916Receiving objects: 100% (919/919), 80.24 MiB | 956.00 KiB/s, done.
Receiving objects: 100% (919/919), 80.41 MiB | 391.00 KiB/s, done.

Resolving deltas: 100% (466/466), done.
```

Step 2 : Use the following command to create .env file

```
1 copy env.sample .env
```



```
C:\Users\yovis\Documents\watson-discovery-news>copy env.sample .env
1 file(s) copied.
```

Step 3: Use the following command to Enter Watson discovery service credentials in .env file

step to download credentials [here](#)

Copy paste the credentials from downloaded file to .env.

```
1 Notepad .env
```

Step 4: Installing packages and dependencies

A package is a folder containing a program described by a package.json file

This command installs a package, and any packages that it depends on.

```
1 npm install
```

```
Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

> core-js@2.6.10 postinstall C:\Users\user\Desktop\smartbridge\watson-discovery-news\node_modules\babel-runtime\node_modules\core-js
> node postinstall || echo "ignore"

> nodemon@1.18.9 postinstall C:\Users\user\Desktop\smartbridge\watson-discovery-news\node_modules\nodemon
> node bin/postinstall || exit 0

Love nodemon? You can now support the project via the open collective:
> https://opencollective.com/nodemon/donate

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

added 1496 packages from 818 contributors in 617.718s
```

Step 5: Running the deployed app.

```
1 npm start
```

```
C:\> npm

Microsoft Windows [Version 10.0.18363.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\yovis>cd C:\Users\yovis\Desktop\watson-discovery-news
C:\Users\yovis\Desktop\watson-discovery-news>npm start

> watson-discovery-news@1.0.0 start C:\Users\yovis\Desktop\watson-discovery-news
> node --max_old_space_size=512 app.js

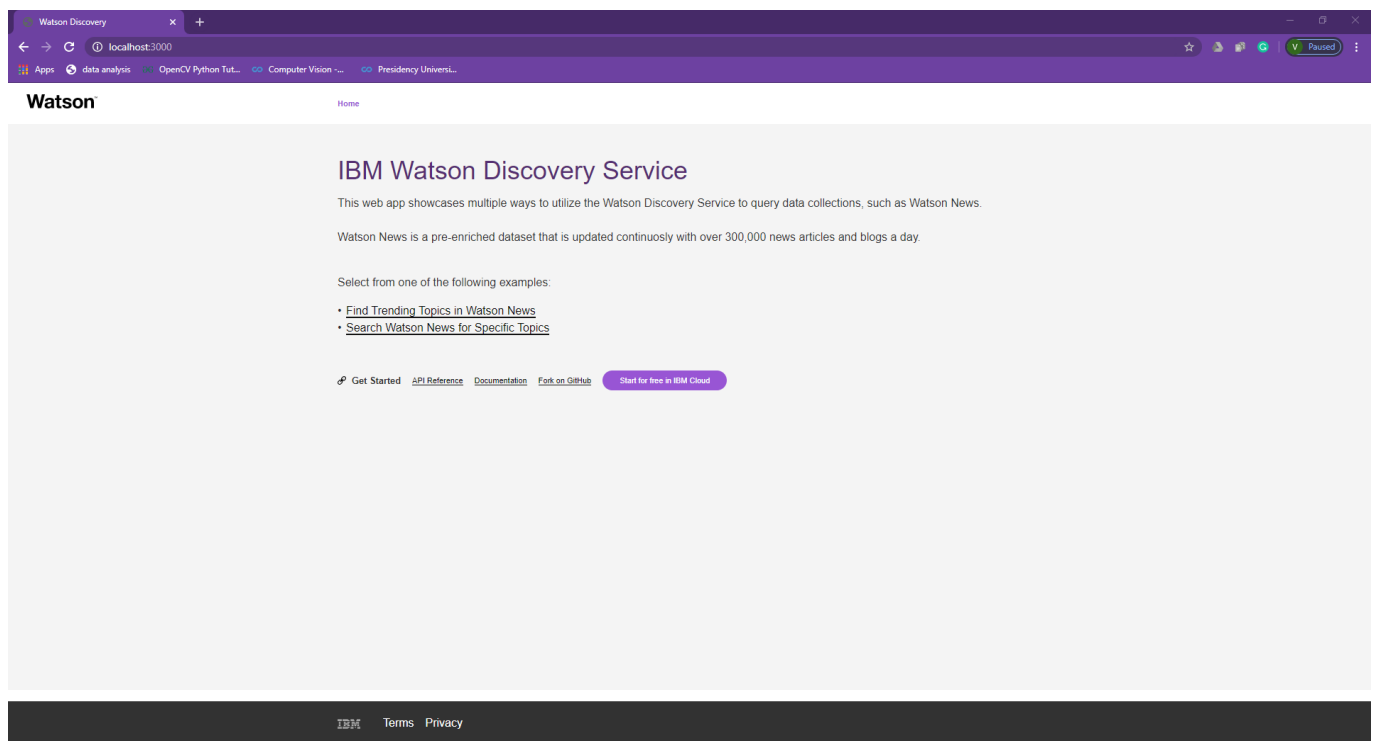
Initializing Botkit v0.7.5
info: ** No persistent storage method specified! Data may be lost when process shuts down.

*****
* WARNING: Your bot is operating without recommended security mechanisms in place.      *
* Initialize your Botkit controller with a clientSigningSecret paramter to enable      *
* verification that all incoming webhooks originate with Slack:                      *
*                                                                                       *
* var controller = new Botkit.slackbot({clientSigningSecret: <my secret from slack>}); *
*                                                                                       *
*****
>> Botkit docs: https://botkit.ai/docs/readme-slack.html#securing-outgoing-webhooks-and-slash-commands
>> Slack docs: https://api.slack.com/docs/verifying-requests-from-slack

Watson Discovery News Server running on port: 3000
```

The application will be available in your browser at `http://localhost:3000`

sample output:

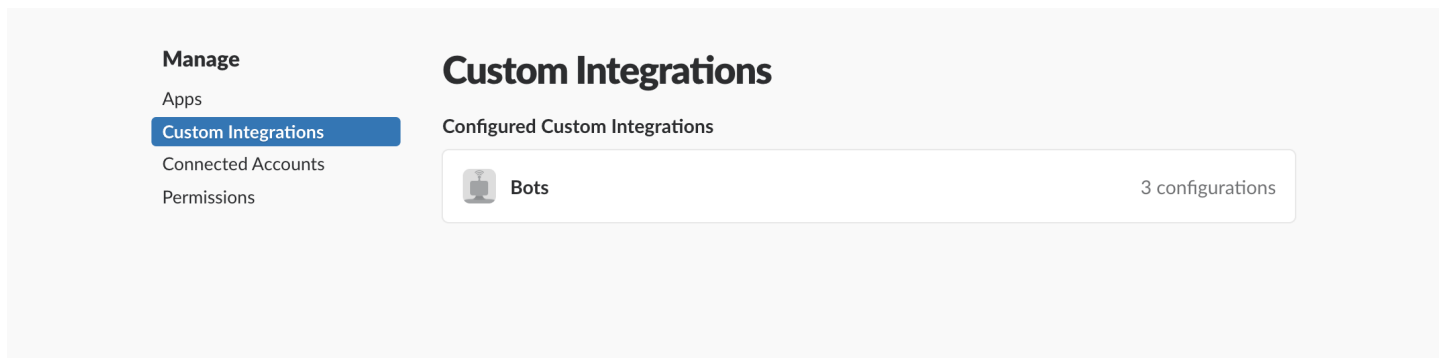


Task 2 Integrating Slack bot with Watson discovery news app.

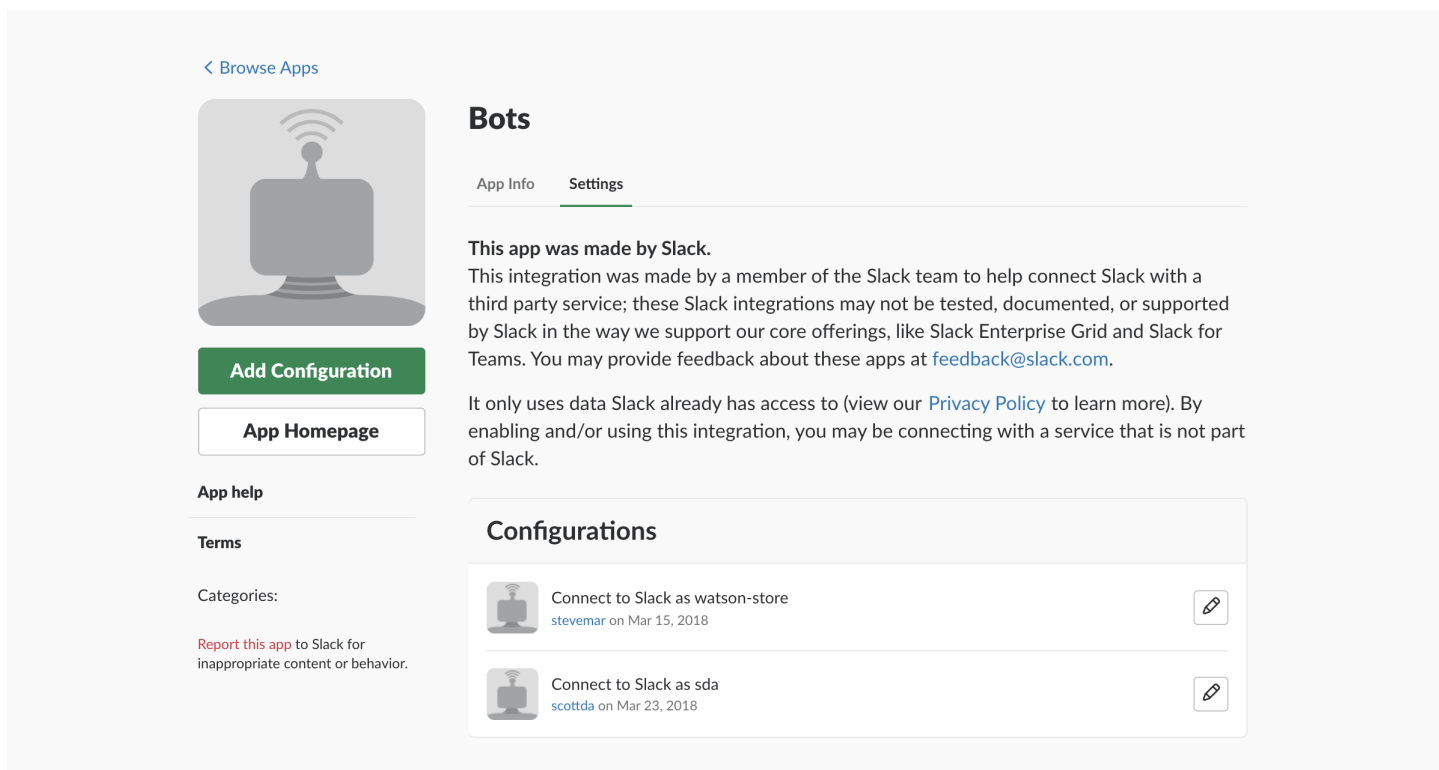
To integrate a new Slack Bot into your existing Slack team, navigate to <https://<my.slack.com>/apps/manage/custom-integrations>, where <my.slack.com> is the Slack workspace you want to customize.

2.1 Creating slack bot

Step 1: From the **Custom Integrations** page, select the **Bots** option.




Step 2: To add a new bot, select the **Add Configuration** button.



Step 3: Enter a username for the bot and click **Add bot integration**.

[Browse apps](#) > [Custom Integrations](#) > [Bots](#) > New configuration



Bots

Connect a bot to the Slack Real Time Messaging API.

Run code that listens and posts to your Slack team just as a user would.

Username

Start by choosing a username for your bot


Username must be all lowercase. They cannot be longer than 21 characters and can only contain letters, numbers, periods, hyphens, and underscores. Most people choose to use their first name, last name, nickname, or some combination of those with initials.

Add bot integration

By creating a bot integration, you agree to the [Slack API Terms of Service](#).

Step 4: Once created, save the **API Token** that is generated.

[Browse Apps](#) > [Custom Integrations](#) > [Bots](#) > Edit configuration



Bots

Added by stevemar on December 17th, 2018

[Disable](#) • [Remove](#)

Run code that listens and posts to your Slack team just as a user would.

Setup Instructions


Please refer to our [bot user API documentation](#), which tells you everything you need to know about setting up a bot integration.

Integration Settings

API Token

The library you are using will want an API token for your bot.

[Regenerate](#)

 Be careful when sharing bot user tokens with applications. Do not publish bot user tokens in public code repositories. [Review token safety tips](#).


Customize Name

Choose the username for this bot.

Username must be all lowercase. They cannot be longer than 21 characters and can only contain letters, numbers, periods, hyphens, and underscores. Most people choose to use their first name, last name, nickname, or some combination of those with initials.

Customize Icon

Change the icon used for this bot.

 or

2.2 Configuring slack bot

Step 1: Edit the .env file and enter the Slack Bot **API Token** saved in the previous step.

```
# Slack
SLACK_BOT_TOKEN=<slack_bot_token>
#note paste token without the brackets
```

Step 2: Restart application

```
1 npm start
```

Sample output on cmd:

```
C:\Users\yovis\Desktop\watson-discovery-news>npm start

> watson-discovery-news@1.0.0 start C:\Users\yovis\Desktop\watson-discovery-news
> node --max_old_space_size=512 app.js


Initializing Botkit v0.7.5
info: ** No persistent storage method specified! Data may be lost when process shuts down.


*****
* WARNING: Your bot is operating without recommended security mechanisms in place.      *
* Initialize your Botkit controller with a clientSigningSecret paramter to enable      *
* verification that all incoming webhooks originate with Slack:                      *
*                                                                                       *
* var controller = new Botkit.slackbot({clientSigningSecret: <my secret from slack>}); *
*                                                                                       *
*****
>> Botkit docs: https://botkit.ai/docs/readme-slack.html#securing-outgoing-webhooks-and-slash-commands
>> Slack docs: https://api.slack.com/docs/verifying-requests-from-slack


Watson Discovery News Server running on port: 3000
Browserslist: caniuse-lite is outdated. Please run next command `npm update`
notice: ** BOT ID: watson_news ...attempting to connect to RTM!
notice: RTM websocket opened
Slack Bot host route: http://localhost:3000
```


Bot on slack:


#discovery ☆
60 | Add a topic

 **watson_news** APP 11:49 AM
was added to #discovery by Vishal Vishwakarma.


 **Vishal Vishwakarma** 11:49 AM
[@watson_news](#) hello
👤 1 😊


 **watson_news** APP 11:49 AM
Hello.


 **Vishal Vishwakarma** 11:50 AM
[@watson_news](#) news please


 **watson_news** APP 11:50 AM
Hi there!


What news are you interested in?

 **Vishal Vishwakarma** 11:50 AM
olympic

 **watson_news** APP 11:50 AM
You want me to search for news articles about [olympic](#)?

 **Vishal Vishwakarma** 11:50 AM
yes

 **watson_news** APP 11:50 AM
OK searching...

<https://www.insidethegames.biz/articles/1093545/olympic-solidarity-increases-noc-funding> (edited)
 **insidethegames.biz**
Olympic Solidarity increases funding to NOCs following Tokyo 2020 postponement
Olympic Solidarity have increased funding to National Olympic Committees following the postponement of the Tokyo 2020 Olympic and Paralympic Games.
Apr 24th (1 MB) ▾

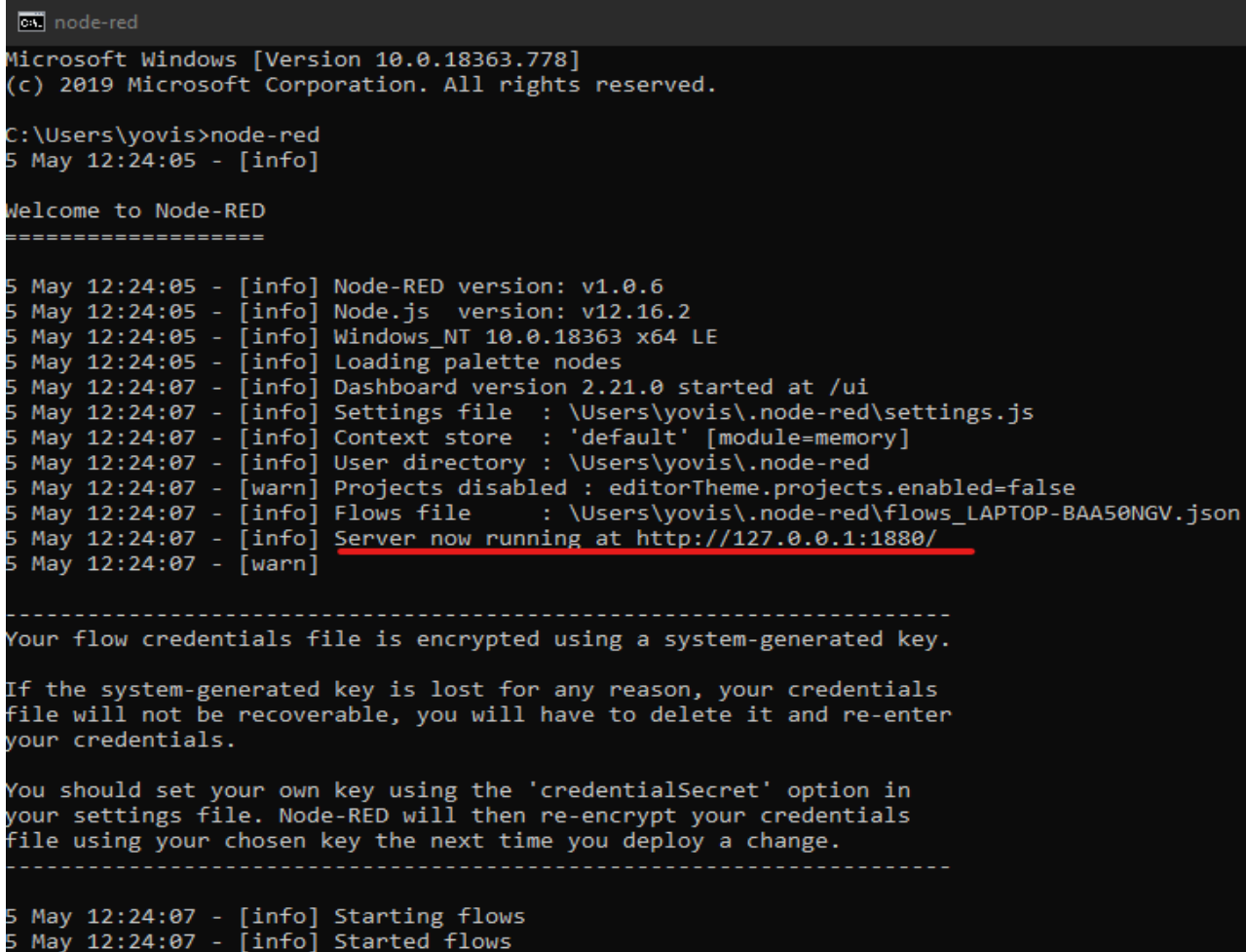
Task 3 Creating App user interface using node-red

Installation node-red has been done earlier to refer [click-here](#).

Step 1: Running node-red

Once installed as a global module you can use the node-red command to start Node-RED in your terminal. You can use Ctrl-C or close the terminal window to stop Node-RED.

```
1 node-red
```



```
node-red
Microsoft Windows [Version 10.0.18363.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\yovis>node-red
5 May 12:24:05 - [info]

Welcome to Node-RED
=====

5 May 12:24:05 - [info] Node-RED version: v1.0.6
5 May 12:24:05 - [info] Node.js version: v12.16.2
5 May 12:24:05 - [info] Windows_NT 10.0.18363 x64 LE
5 May 12:24:05 - [info] Loading palette nodes
5 May 12:24:07 - [info] Dashboard version 2.21.0 started at /ui
5 May 12:24:07 - [info] Settings file : \Users\yovis\.node-red\settings.js
5 May 12:24:07 - [info] Context store : 'default' [module=memory]
5 May 12:24:07 - [info] User directory : \Users\yovis\.node-red
5 May 12:24:07 - [warn] Projects disabled : editorTheme.projects.enabled=false
5 May 12:24:07 - [info] Flows file : \Users\yovis\.node-red\flows_LAPTOP-BAA50NGV.json
5 May 12:24:07 - [info] Server now running at http://127.0.0.1:1880/
5 May 12:24:07 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

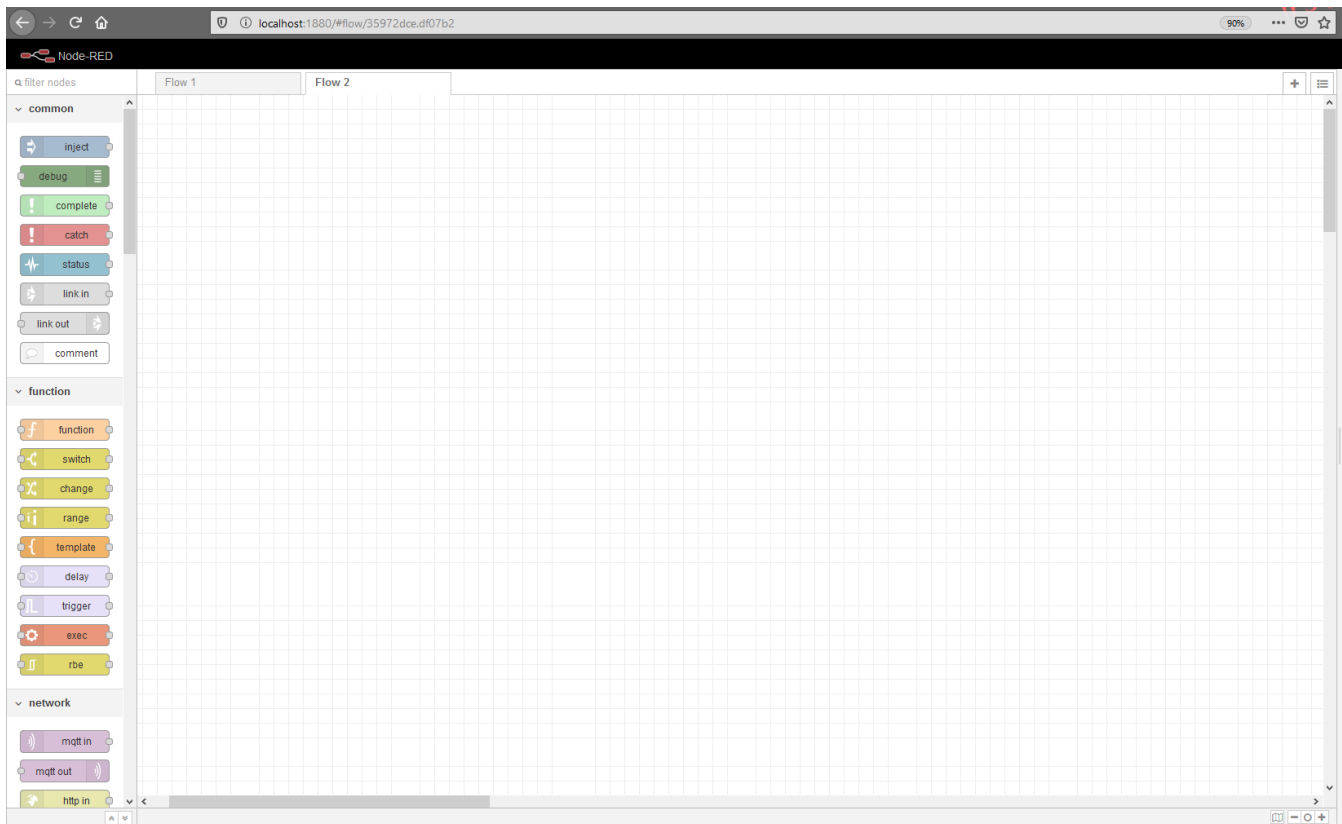
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

5 May 12:24:07 - [info] Starting flows
5 May 12:24:07 - [info] Started flows
```

You can then access the Node-RED editor by pointing your browser at <http://localhost:1880>.

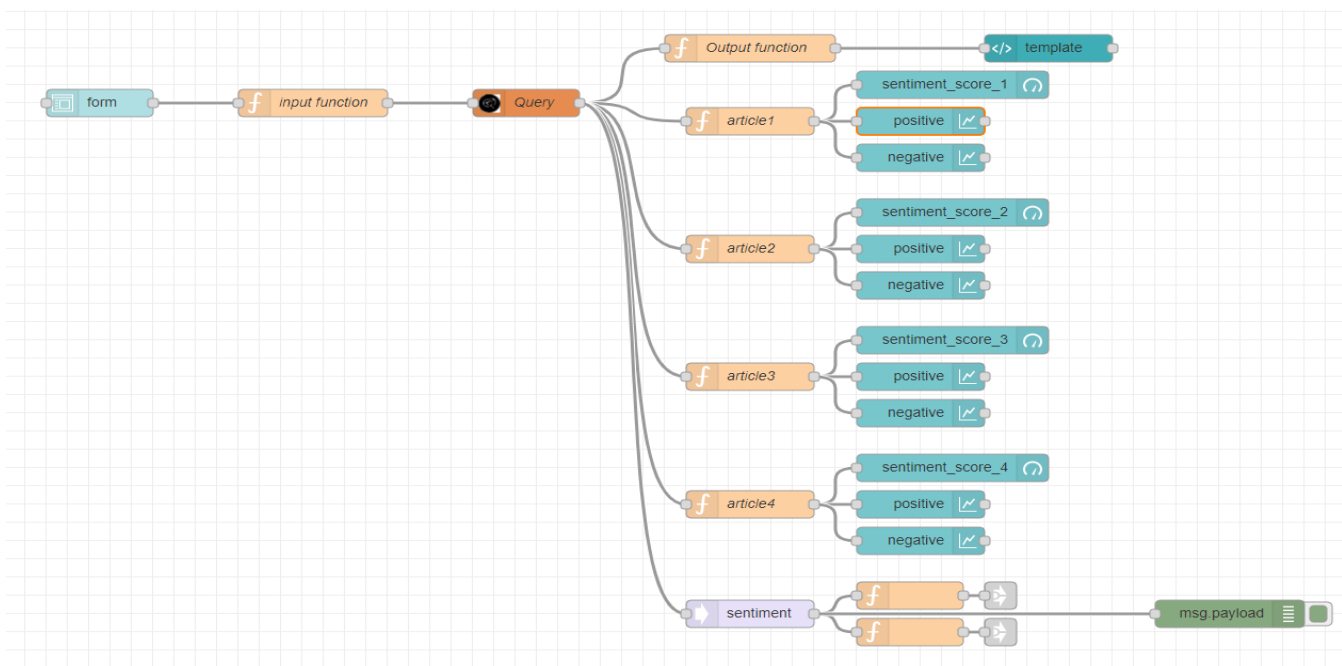
Sample Output at <http://localhost:1880>. :



Step 2: Node-red flow for Watson discovery news

A **flow** is represented as a tab within the editor workspace and is the main way to organize **nodes**. **Flow** tabs.

Drag and drop the nodes from the nodes from palette then connect using wire



Nodes used:

1. Form node
2. Function node
3. Query node
4. Template node
5. gauge node
6. chart node

Step 3: Configuring the nodes

► Form node

The screenshot shows the 'Edit form node' configuration window. At the top, there are 'Delete', 'Cancel', and 'Done' buttons. Below is a 'Properties' section with a sidebar on the left containing icons for Group, Size, Label, and Form elements. The main area contains the following fields:

- Group:** A dropdown menu showing '[Home] Ai NEWS' with an edit icon.
- Size:** A text input field with the value 'auto'.
- Label:** A text input field with the value 'optional label'.
- Form elements:** A table with columns: Label, Name, Type, Required, Rows, and Remove.

Label	Name	Type	Required	Rows	Remove
e.g. Name	input	Text	<input checked="" type="checkbox"/>		

Below the table is a '+ element' button. At the bottom, there is a 'Buttons' section with a 'submit' button and a 'cancel' button.

► Input function node

The screenshot shows the 'Edit function node' dialog box. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below these is a 'Properties' tab with a gear icon. The 'Name' field is set to 'input function'. The 'Function' field contains a JSON object and a return statement. The code is as follows:

```
1 msg={
2   discoveryparams:
3   {
4     "environment_id":"system",
5     "query":msg.payload.input
6   }
7 }
8 return msg;
```

Query node configuration is shown in the next [Task](#)

► Output function node

The screenshot shows the 'Edit function node' dialog box. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below these is a 'Properties' tab with a gear icon. The 'Name' field is set to 'Output function'. The 'Function' field contains a JavaScript snippet. The code is as follows:

```
1
2 msg.payload=msg.search_results.results;
3 return msg;
4
```

Task 4 Integrating Watson discovery service with UI

► Configuring discovery node

Edit discovery node

Delete Cancel Done

Properties

Name Query

Username apikey

Password

API Key

Service Endpoint https://api.eu-gb.discovery.watson.cloud.ibm.com

Method: Search in collection

Environment ID system

Collection ID news-en

Number of documents 10

☐ Passages

☐ Query in Plain Language

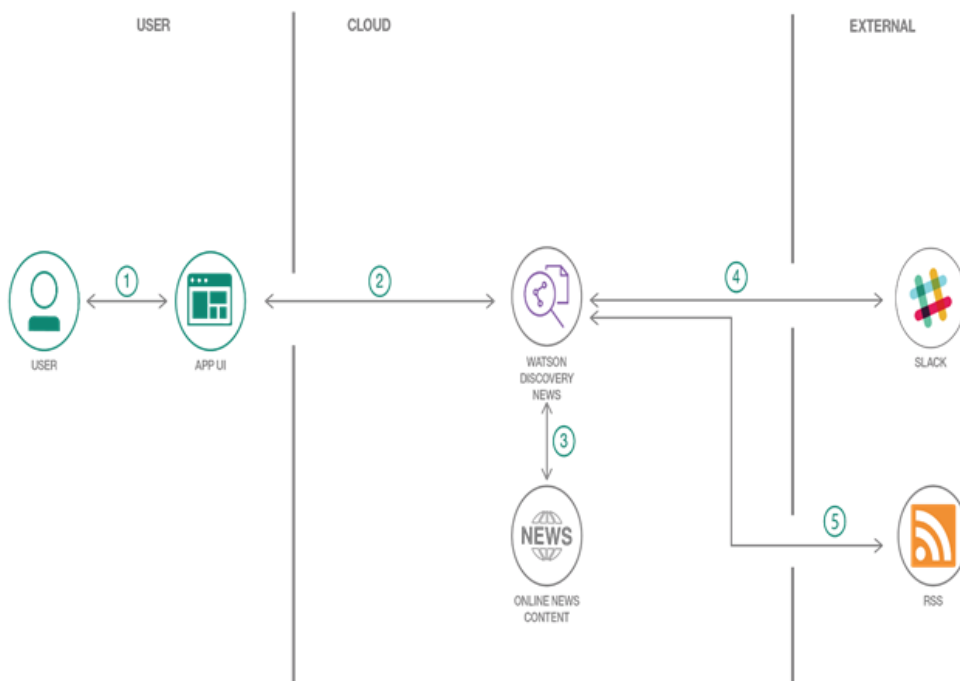
Password , API key , Service Endpoint can be found in the IBM credentials file downloaded [here](#)

Experimental Investigation

First we use the discovery service to configure and query adding our collection. A red node application is created in which the discovery is integrated and a simple flow of 5 nodes is created to enter the news topic and the results show related news.

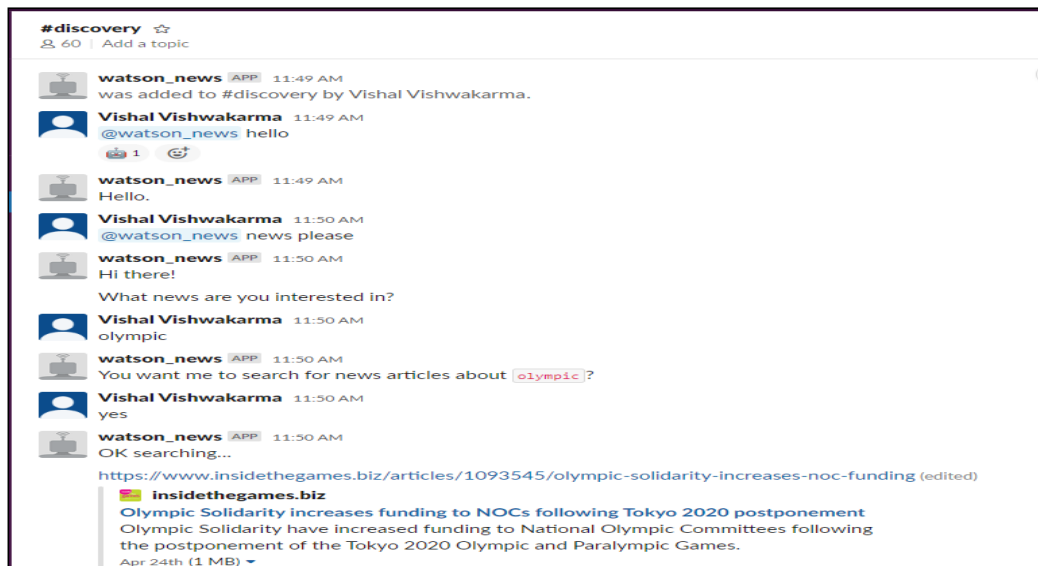
Slack then integrates with Watson's discovery service so that news articles can be searched on more than one platform, and finally sentiment analysis is performed on the data / news articles being searched.

Flowchart



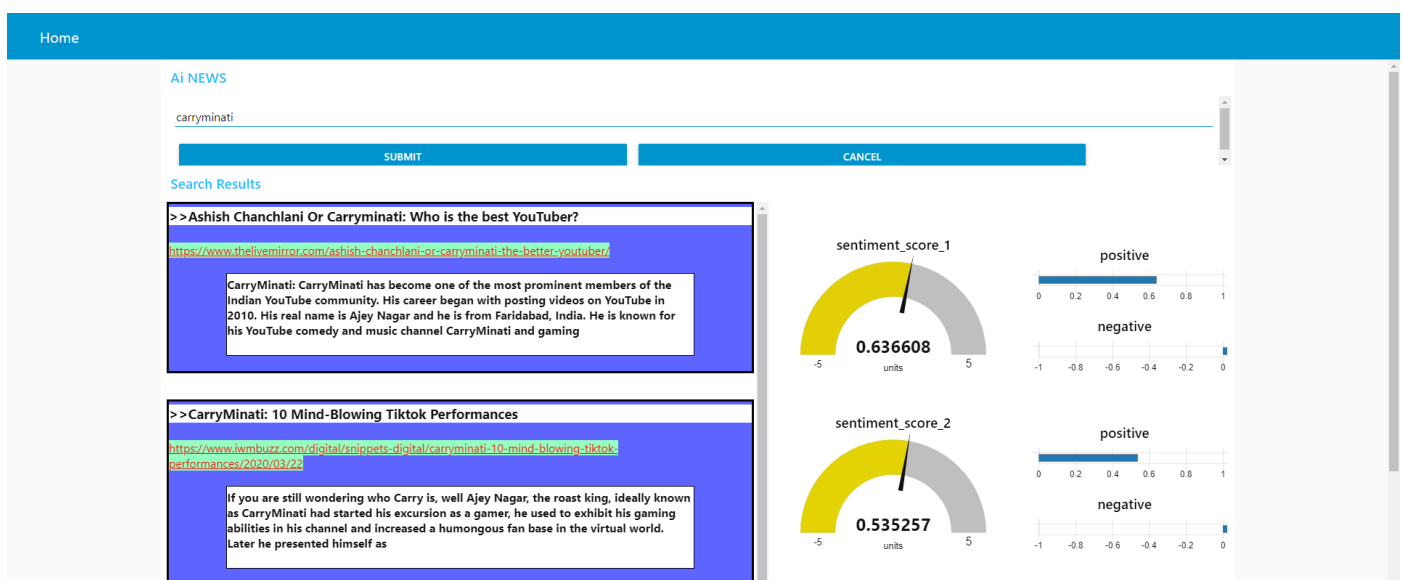
Result

Bot on slack:



The Watson discovery application will be able to fetch relevant news articles within milliseconds, provide sentiment analysis on the fetched data, extract features, concepts, and keywords in the form of a simple UI.

User Interface can be found at <http://localhost:1880/ui/>



Output URL: <http://localhost:1880/ui/#!/0?socketid=iiYB3Q5322Xrtq3KAAAC>

Advantages and Disadvantages

- The web application provides interactive sentiment analysis.
- It can be accessed through more than one platform that is through slack.
- It collects and delivers the most recent data.
- It does not have additional features like storing news history.
- It does not provide a stand-alone app rather uses a web application.

Applications

- This web applications can be used by any user in need of accurate and fast results.
- Can be used by firms and organizations.
- Can be used in stock market to make predictions.

Conclusion

This project gives some basic working knowledge of the Watson Discovery Service and showed you how to use Discovery along with JavaScript and Node.js to build your own news mining web application.

It also gives insight into real-world applications of AI and helps us understand Slack better.

Future Scope

- The web application can be integrated with cloud and made into a mobile app to use it on-the-go.
- Additional sentiments can be added in the UI.
- Related and trending news topics can be shown to the user.

Bibliography

- <https://github.com/IBM/watson-discovery-news/>
- <https://nodered.org/>
- <https://slack.com/intl/en-in/help/articles/360002079527-A-guide-to-Slack%E2%80%99s-Discovery-APIs>
- <https://www.w3schools.com/>

Appendix

Source code

► Template node

CSS part

```
1 <head>
2   <style>
3     .cont{
4       background-color:#5e64ff;
5       outline-style:groove;
6     }
7     .link{
8       color:red;
9       background-color:#94ffbf;
10    }
11    .link:hover{
12      font-size:large;
13    }
14  div.inner{
15    margin:auto;
16    width:80%;
17    background-color:white;
18    border:1px solid black;
19  }
20  p{
21    margin: 5%;
22    font-weight: bold;
23    color: #000000;
24  }
25  </style>
26 </head>
```

HTML part

```
1 <body>
2
3 <div class="cont">
4 <h3 class="title">>>{{msg.payload[0].title}}</h3>
5 <br/>
6 <a target="__blank" class="link" href="{{msg.payload[0].url}}">{{msg.payload[0].url}}</a>
7 <br/>
8 <br/>
9 <div class="inner"><p><b>{{msg.payload[0].text}}</b></p> <br/></div>
10 <br/>
11 </div>
12 <br>
13 <div class="cont">
14 <h3 class="title">>>{{msg.payload[1].title}}</h3>
15 <br/>
16 <a target="__blank" class="link" href="{{msg.payload[1].url}}">{{msg.payload[1].url}}</a>
17 <br/>
18 <br/>
19 <div class="inner"><p><b>{{msg.payload[1].text}}</b></p> <br/></div>
20 <br/>
21 </div>
22 <br>
23 <div class="cont">
24 <h3 class="title">>>{{msg.payload[2].title}}</h3>
25 <br/>
26 <a target="__blank" class="link" href="{{msg.payload[2].url}}">{{msg.payload[2].url}}</a>
27 <br/>
28 <br/>
29 <div class="inner"><p><b>{{msg.payload[2].text}}</b></p> <br/></div>
30 <br/>
31 </div>
32 <br>
33 <div class="cont">
34 <h3 class="title">>>{{msg.payload[3].title}}</h3>
35 <br/>
36 <a target="__blank" class="link" href="{{msg.payload[3].url}}">{{msg.payload[3].url}}</a>
37 <br/>
38 <br/>
39 <div class="inner"><p><b>{{msg.payload[3].text}}</b></p> <br/></div>
40 <br/>
41 </div>
42 <br>
43 </body>
```