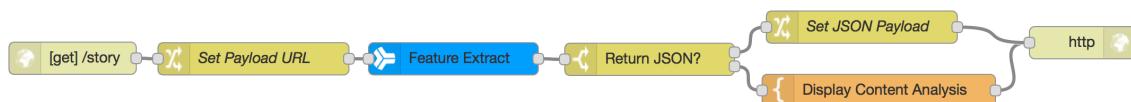


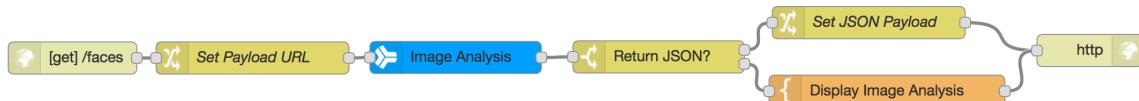
# AlchemyAPI in Node-RED

## Hands-On Lab

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



Extract keywords, entities, concepts, sentiment and more from a news article  
(see *Analyze a News Article in Node-RED*)



Extract gender, name, categories, and age-range of faces in an image  
(see *Analyzing Faces in an Image*)



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

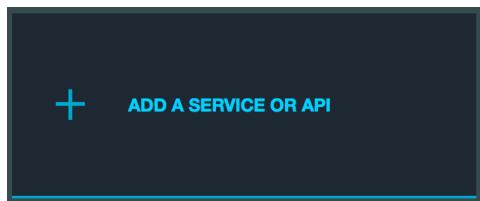


# Add Alchemy API Service in IBM Bluemix

The AlchemyAPI ([alchemy.ai](http://alchemy.ai)) offers three services: Language, Vision, and Data news. In this tutorial, we'll use the Language API to analyze content from a webpage URL. The Language API returns keyword, entities, and concepts, sentiment, and other information about the content being analyzed. We'll also use the Vision API to locate faces and attributes about people in an image. This tutorial uses the Node-RED boilerplate in IBM Bluemix with the AlchemyAPI service.

To get started using the Alchemy API, you'll need to create an AlchemyAPI service to get an API key.

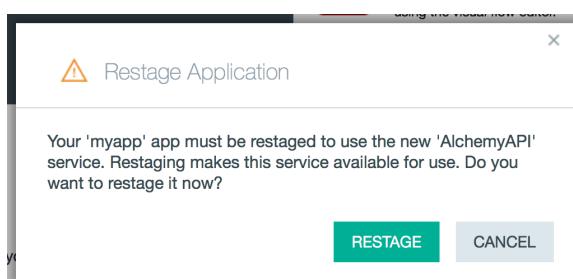
1. Go to the application overview for your Node-RED application in the IBM Bluemix dashboard and click **Add a service or API**.



2. Click the **AlchemyAPI** node 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.

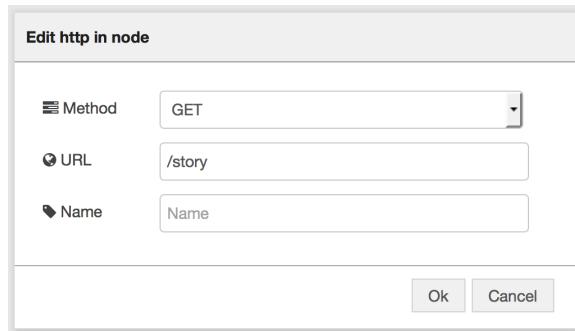


4. When the application has finished restaging, open the Node-RED Flow Editor. If you already have Node-RED open, refresh the page.

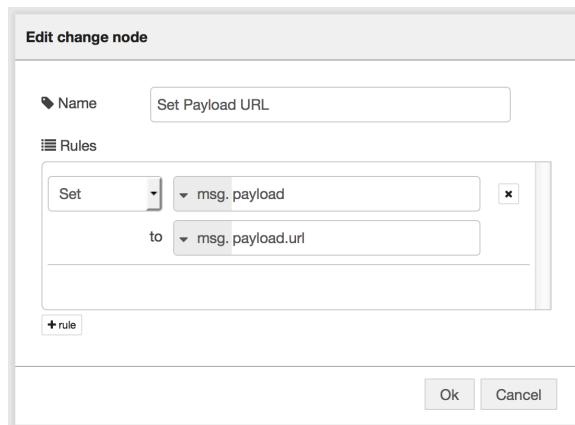
# Analyze a News Article in Node-RED

The AlchemyLanguage API analyzes unstructured text and provides a handful of attributes including keywords, entities, concepts, taxonomy, document sentiment, author, publication date, title and more. You can either provide a URL where the content resides, or a body of text to analyze. In this section, we will analyze a news article accessible via an URL.

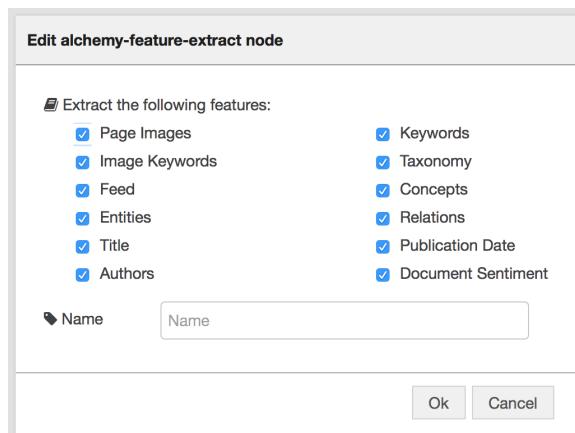
1. Add a  node as shown below.



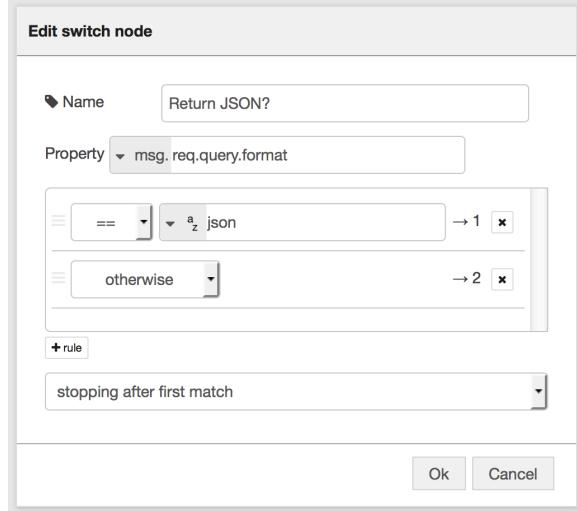
2. Add a  node as shown below. This will take the *url* query parameter and place it in the message *payload* to be passed to the Feature Extract node.

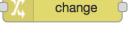


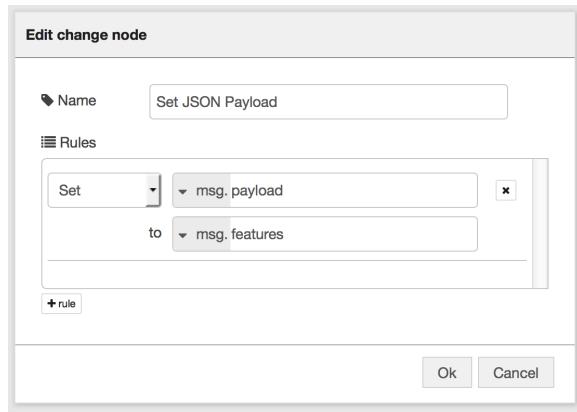
3. Add a  node. Check the options that you want to be included in the response.



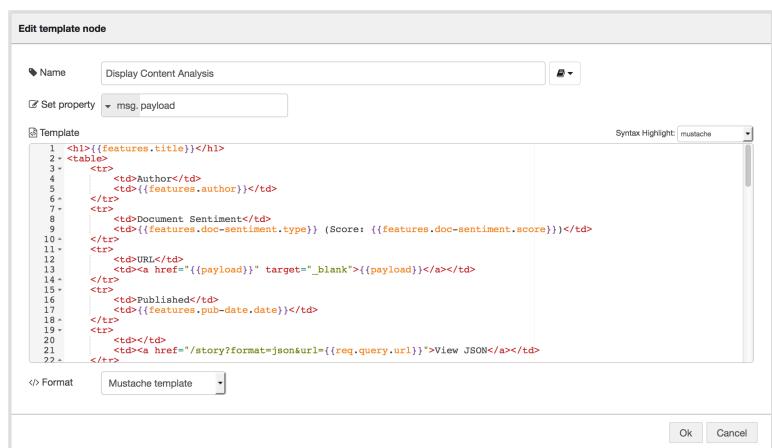
4. To make this application versatile, we'll split the flow so our application will return the results in two formats. The first option will be where the results are returned in JSON format, great for use in applications that can call the web endpoint and consume the JSON. The second option will be a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



5. For the flow where the JSON should be returned, we can simply return the contents of `msg.features`. Add a  node as shown below. This will move the results from the `features` property to the message payload.

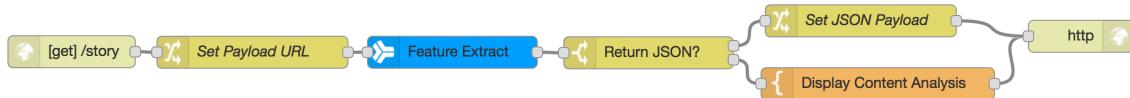


6. For the flow where a webpage should be returned, add a  node with the HTML in the file named `1-display-content-analysis.html`.



Get the code:  
ibm.biz/Bd48EZ

7. Add a  node. Connect the nodes together as shown below.

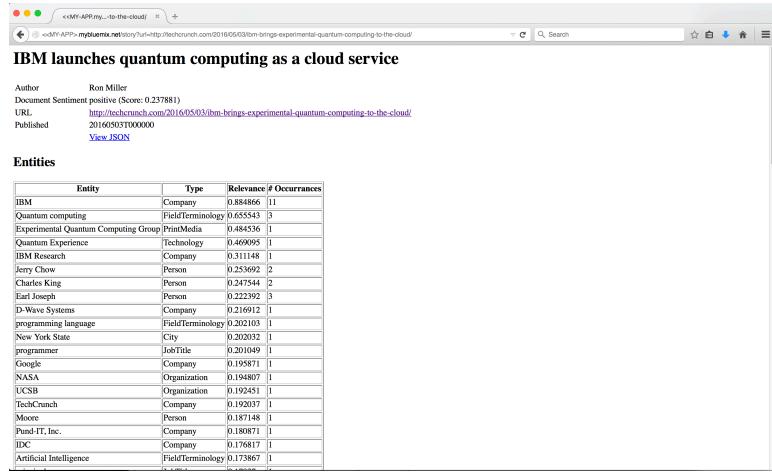


8. Click on the red  Deploy button in the upper-right corner of the screen to save and deploy your changes.
  9. Visit a news website and copy the URL of a publicly accessible news article. Make sure the content isn't behind an authentication wall where you have to sign in to access the content.
  10. Open a browser tab and visit your application's endpoint, passing in the URL to the content:

<http://<<MY-APP>>.mybluemix.net/story?url=<<URL-TO-STORY>>>

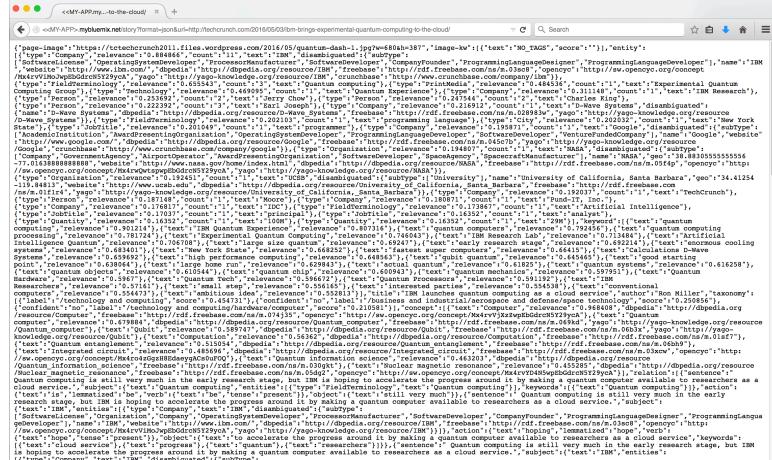
- Replace <>MY-APP<> with the host of the Node-RED application you chose.
  - Replace <>URL-TO-STORY<> with the URL of the content.

11. Depending on the content located at the URL, you may see a list of attributes including keywords, entities, concepts, taxonomy, document sentiment, author, publication date, title and more mentioned within the text.



12. To see the JSON representation of the content insert `format=json` in the URL query string:

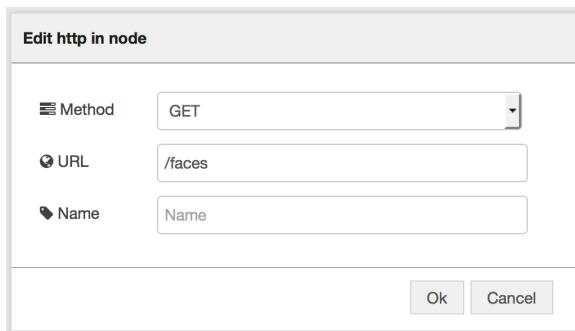
`http://<<MY-APP>>.mybluemix.net/story?format=json&url=<<URL-TO-STORY>>`



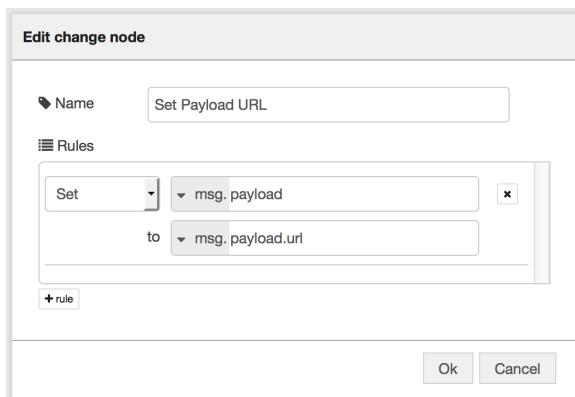
# Analyzing Faces in an Image

The AlchemyVision API analyzes images and provides a handful of attributes including gender, age-range, and if known, the name and type, of each person in the given image. You can either provide a URL or the binary data of the image to analyze. In this section, we will faces in an image, and provide an age-range, gender, and if known, the name and type of the person.

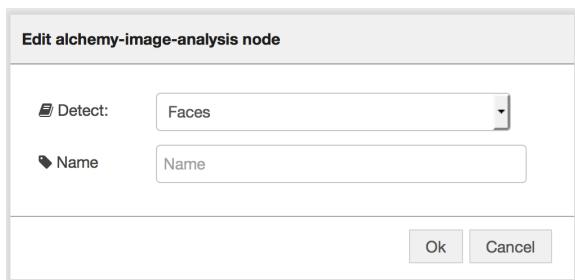
1. Add a  node as shown below.



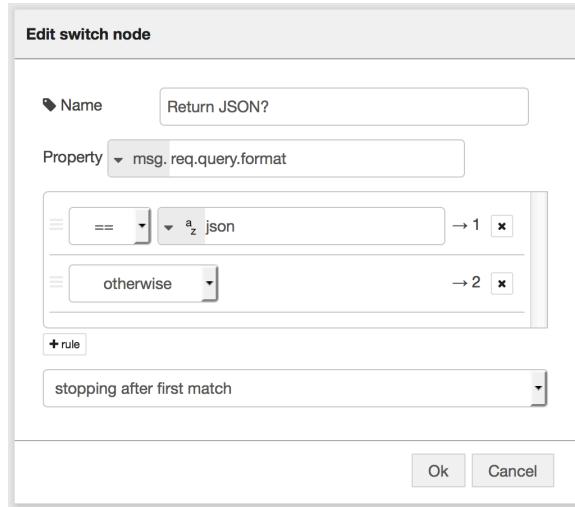
2. Add a  node as shown below. This will take the *url* query parameter and place it in the message *payload* to be passed to the Image Analysis node.

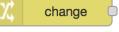


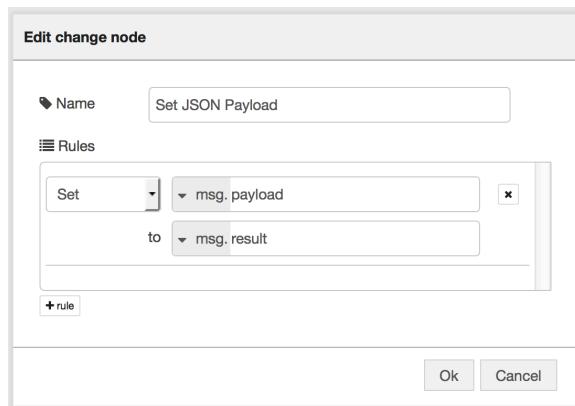
3. Add a  node. Select **Faces** from the Detect menu as shown below.

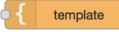


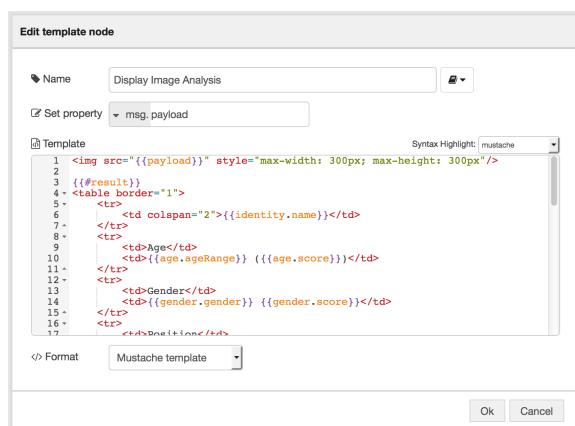
4. To make this application versatile, we'll split the flow so our application will return the results in two formats. The first option will be where the results are returned in JSON format, great for use in applications that can call the web endpoint and consume the JSON. The second option will be a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



5. For the flow where JSON should be returned, we can simply return the contents of `msg.result`. Add a  node as shown below. This will move the results from the `result` property to the message `payload`.

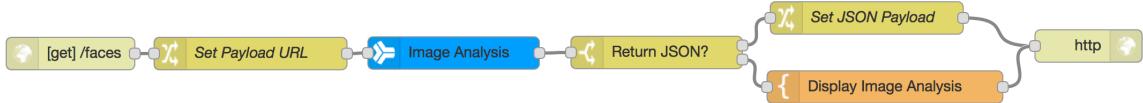


6. For the flow where a webpage should be returned, add a  node with the HTML in the file named `2-display-image-analysis.html`.



Get the code:  
[ibm.biz/Bd48Ey](http://ibm.biz/Bd48Ey)

7. Add a  node. Connect the nodes together as shown below.

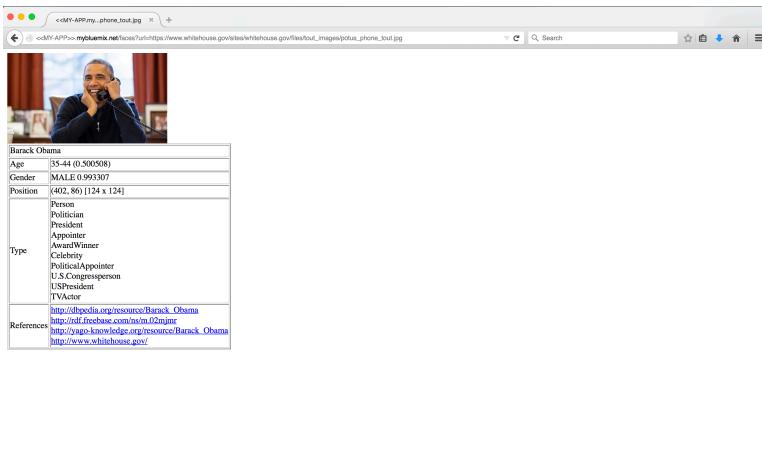


13. Click on the red  button in the upper-right of the screen to save and deploy your changes.
14. Locate an image of at least one person and copy the URL of the image. Make sure the content isn't behind an authentication wall where you have to sign in to access the content.
15. Open a browser tab and visit your application's endpoint, passing in the URL to the image:

`http://<<MY-APP>>.mybluemix.net/story?url=<<URL-TO-IMAGE>>`

- Replace <<MY-APP>> with the hostname of the Node-RED application you chose.
- Replace <<URL-TO-IMAGE>> with the URL of the image.

16. Depending on the content located at the URL, you may see a list of attributes for each face.



17. To see the JSON representation of the content insert `format=json` in the URL query string:

`http://<<MY-APP>>.mybluemix.net/story?format=json&url=<<URL-TO-IMAGE>>`

