

IBM Watson Knowledge Studio

Building a Machine-Learning Annotator

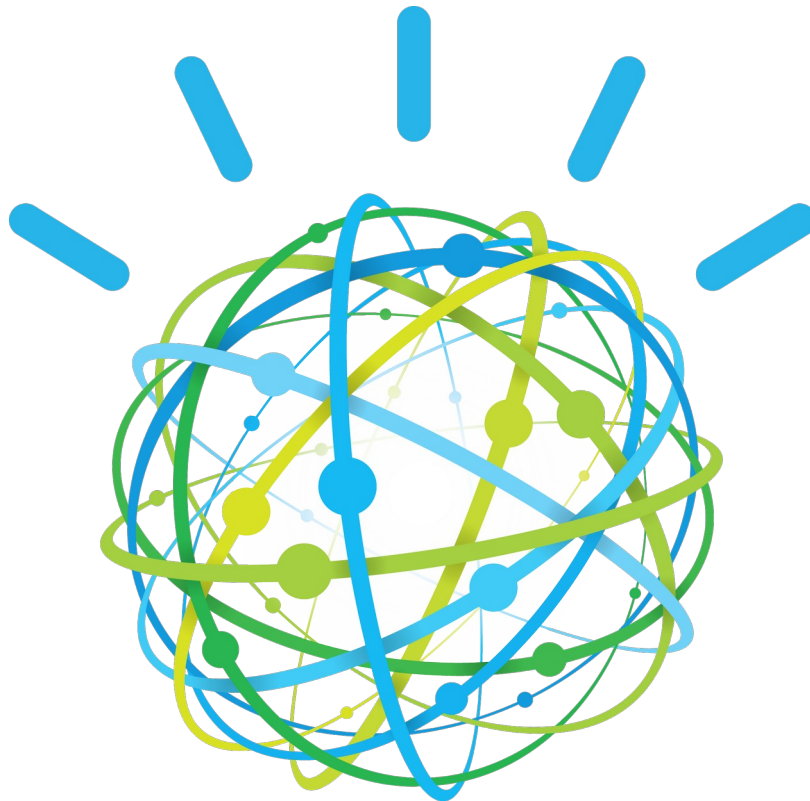
IBM Global Business Partners

Duration: 45 minutes

Updated: May 5, 2020

Klaus-Peter Schlotter

kps@de.ibm.com



IBM

Version 1.0

Overview

The [IBM Watson Developer Cloud](#) offers a variety of services for developing cognitive applications. Each Watson service provides a Representational State Transfer (REST) Application Programming Interface (API) for interacting with the service. Some services, such as the Speech to Text service, provide additional interfaces.

With Watson Knowledge Studio you can teach IBM Watson® the language of your domain with custom models that identify entities and relationships unique to your industry in unstructured text. Build your models in a collaborative environment designed for both developers and domain experts, without needing to write code. Use the models in IBM Watson Discovery, IBM Watson Natural Language Understanding and IBM Watson Explorer.

The features that are available depend on the language. This list may change over time, so always check this in the [online documentation](#).

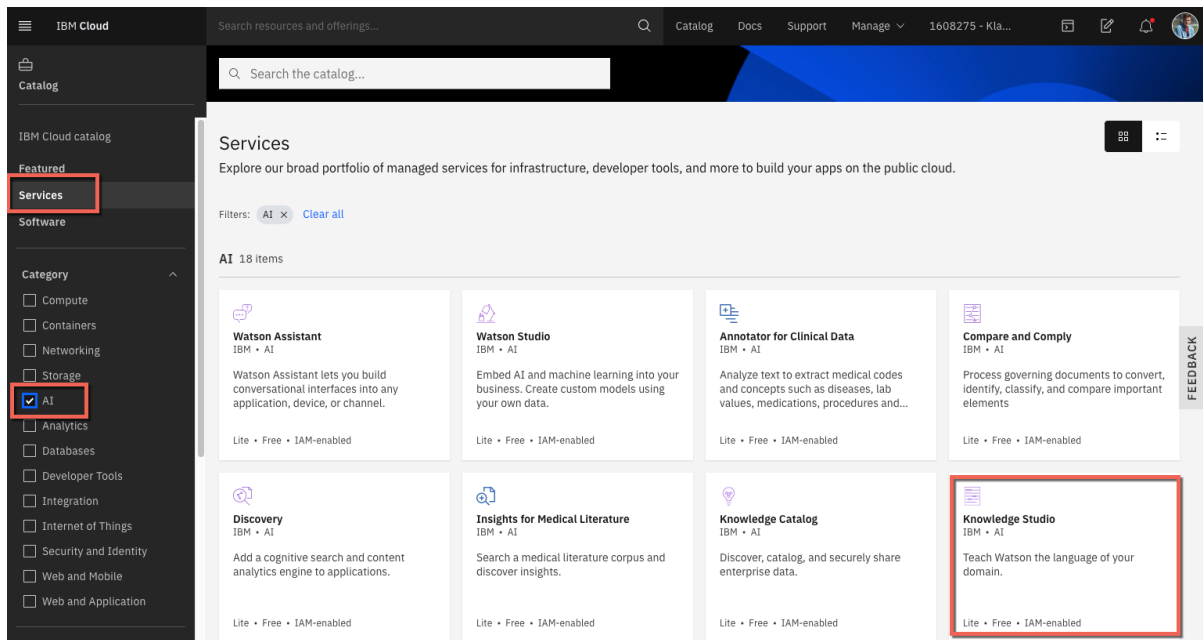
The language model built in this tutorial will be applied Natural Language Understanding app created in Lab 4 that can run on your local workstation and be deployed on the IBM Cloud.

- In a first step, an instance of the Watson Knowledge Studio service will be created.
- In a second step, a Machine Learning annotator will be built and deployed to a Natural Language Understanding service in the IBM Cloud.
- Then the model will be tested with the Postman REST client.
- Finally the Node.js application created in Lab 4 will be used to incorporate the model into an application.

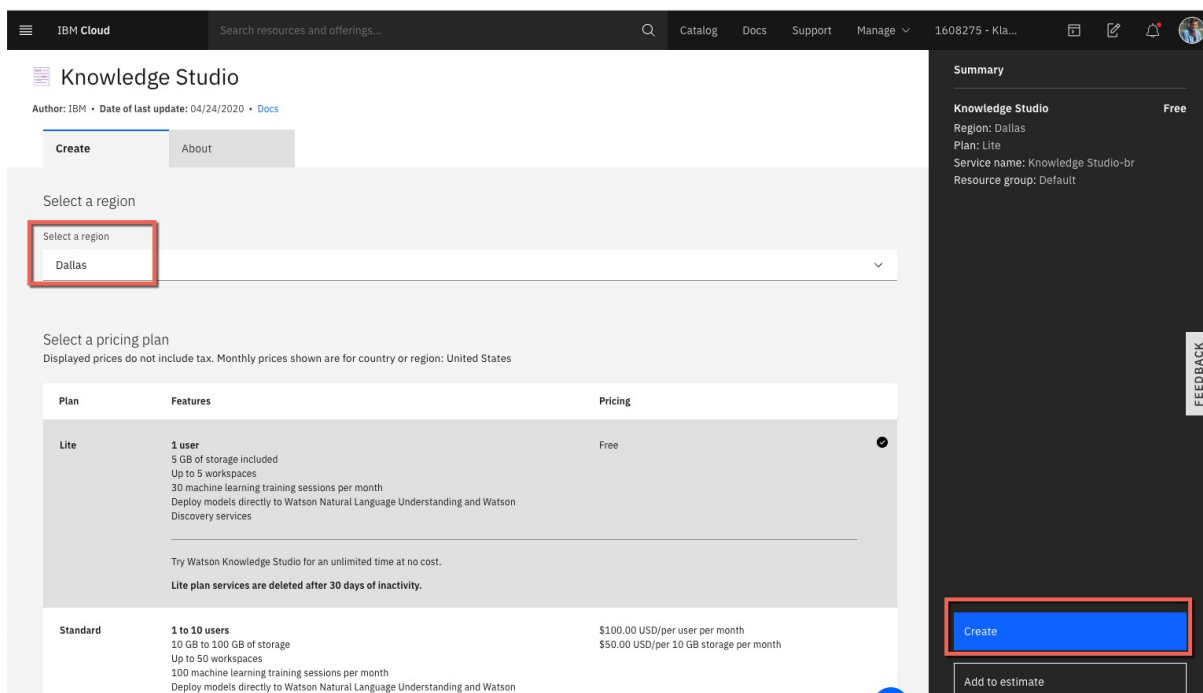
Section 1: Create a Machine Learning Annotator

Step 1 In your IBM Cloud Platform console click [Create resource](#) +

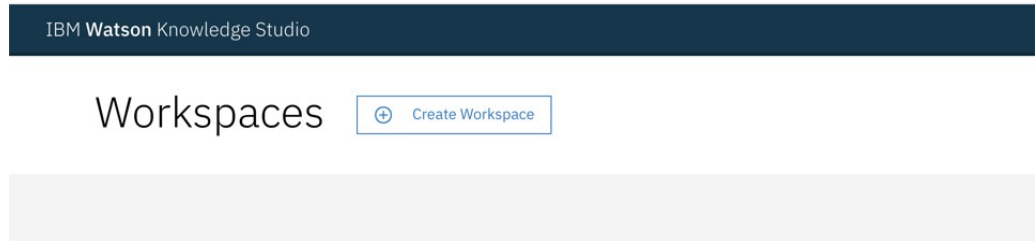
Step 2 Click **Services** then **select** the **AI** category, click on the Knowledge Studio service



Step 3 **Select a region**, enter a unique **name** (you can accept the default) and **select the Lite plan**. Click **Create**.

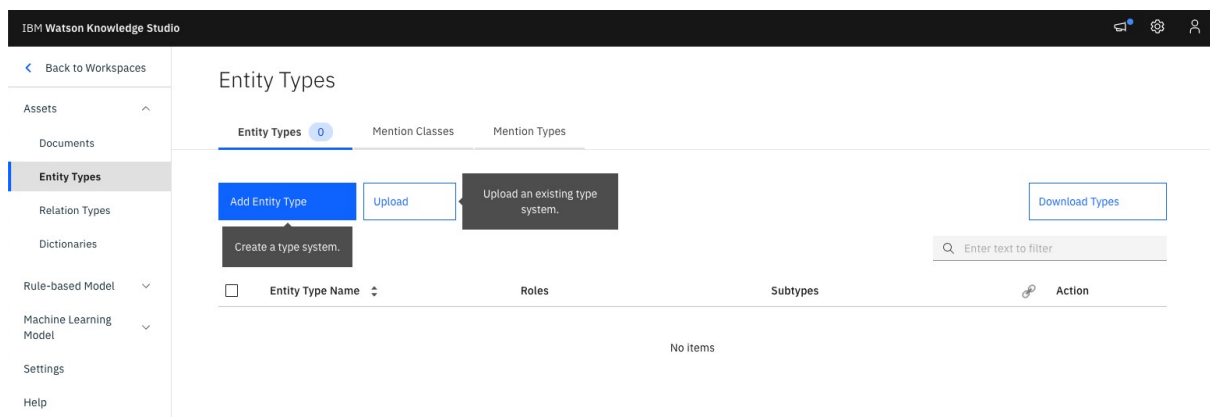


Step 4 In the service instance click [Launch Watson Knowledge Studio](#) to open Knowledge Studio.



Step 5 Click [Create Workspace](#) +, then [Create entities and relations workspace](#). Name the workspace and select *English* as the language. Click *Create*.

Step 6 The workspace opens with *Entity Types* page displayed.



Step 7 On the *Assets* → *Entity Types* click [Add Entity Type](#). As *Entry Type Name* enter PIZZA_TYPE and click [Save](#).

Step 8 Also add an Entity Type NEGATIVE_FEEDBACK

Step 9 On *Assets* → *Relation Types* click [Add Relation Type](#).
name it complaintWith
First Entity Type PIZZA_TYPE
Second Entity Type NEGATIVE_FEEDBACK
Click [Save](#).

Relation Types

Relation Types 1

[Add Relation Type](#)

Enter text to filter

Relation Type	First Entity Type / Role	Second Entity Type / Role	Action
complaintWith	PIZZA_TYPE	NEGATIVE_FEEDBACK	Edit Delete

Step 10 On *Assets* → *Dictionaries* click [Create Dictionary](#),
Name: pizza_types_dict
click [Save](#)

Step 11 [Upload](#) the dictionary entries for the *Pizza Types*
([Dictionary-items-PizzaTypes.csv](#))

Upload Dictionary Entries

Drag a file here, or click to browse for the file that you want to upload.
You can upload any CSV file that contains dictionary terms.
Maximum file size: 1MB

[dictionary-items-organization.csv](#)

[Close](#) [Upload](#)

Step 12 Map this dictionary to the PIZZA_TYPE.

Back to Workspaces

Assets

Documents

Entity Types

Relation Types

Dictionaries

Rule-based Model

Machine Learning Model

Settings

Help

Dictionaries

[Create Dictionary](#)

pizza_dict 24

Language: English | 24 entries

Entity type: PIZZA_TYPE

Rule class: None

[Add Entry](#) [Upload](#) [Download](#)

Enter text to filter

<input type="checkbox"/>	Lemma	Surface Forms	Part of Spee...	Action
<input type="checkbox"/>	Beverly Hills Hotel	Beverly Hills Hotel	Noun	Edit Delete
<input type="checkbox"/>	British Academy	British Academy, BAFTA/LA	Noun	Edit Delete
<input type="checkbox"/>	Golden Globes	Golden Globes	Noun	Edit Delete

Step 13 Click **Add Entry**, enter *four seasons as Surface Form* and select **Noun** as *Part of Speech*. Click **Save**.

Note: For Brazilian Portuguese, English, French, German, Italian, and Spanish, Knowledge Studio does not currently provide an option to specify case-insensitive dictionary-matching, but dictionary entries match text that has a higher case. For example, vehicle in the dictionary matches vehicle, Vehicle or VEHICLE in text, while Sat in the dictionary matches Sat or SAT in text, but not sat.

Step 14 On **Assets** → **Documents** click **Upload Document Sets** to import the training documents ([pizza_reviews.txt](#)).

Step 15 On **Machine Learning Model** → **Pre-annotation** you see the available pre-annotators - dictionary mapping done above is available.

Click **Run Pre-annotators**, select the one for dictionaries and click **Next**.

Run Pre-annotators

Close

Next

Select pre-annotators

Select the pre-annotators that you want to use.

Pre-annotator
<input checked="" type="checkbox"/> Dictionaries

Step 16 Select the documents set imported above and the option *Wipe previous ...* then click **Run**. Click **OK** on the *Confirmation* panel.

Run Pre-annotators

Back

Run

Select document sets

Check if you want to remove previous pre-annotation results from documents before running the pre-annotators. If not checked, all previous annotations are preserved.

* Annotations made by humans outside of the pre-annotation process remain even if you check the wipe option.

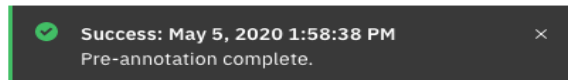
☒ Wipe previous pre-annotation results

Select the document sets or annotation sets that you want to pre-annotate.

Document set	Documents	Pre-annotated documents	Human annotated documents
<input checked="" type="checkbox"/> pizza_reviews.txt_set	1	0	0

The Pre-annotation searches all occurrences of PIZZA_TYPE and marks them accordingly.

The following message should be displayed and the pizza types are annotated in our annotation set.



Step 17 On *Machine Learning Model* → *Annotations* → *Annotations* click [Add Task](#)

Note: You could do this directly on the Ground Truth tab when there is only one person to work on this annotation. But for multiple annotators an Annotation Task is needed to split the work.

Step 18 On *Create Annotation Tasks* click [Create Annotation Sets](#) to create a copy of our training documents for annotation.

Base set: pizza_reviews_text_set

Overlap: 100%

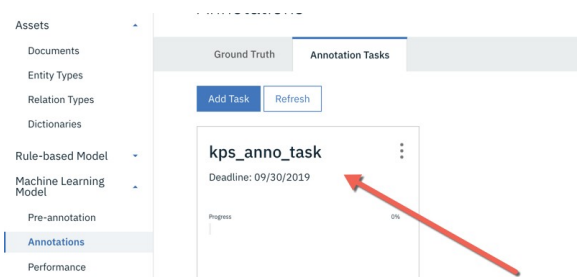
Annotator: <your id>

Set name: pizza_anno_set

Click: [Generate](#)

Step 19 Back on the *Create Annotation Task* page enter a *Task name* pizza_anno_tasks an optional *Deadline* and click [Save](#)

Step 20 On *Machine Learning Models* → *Annotations* → *Annotation Tasks*, click on the task created above.



Step 21 The annotation set is in status In Progress because of the pre-annotation we did before. Now we want to manually annotate the other items.

Click [Annotate](#).

[←](#) | pizza_anno_task [Edit](#)

Deadline: 05/05/2020

Annotations added to annotation sets are not considered ground truth until the annotation sets are submitted and accepted. When an annotation set is accepted, documents that are annotated in only one annotation set immediately become ground truth. Documents that are annotated in two or more annotation sets become overlapping documents that will become ground truth after conflicts are resolved.

[Refresh](#)
[Calculate Inter-Annotator Agreement](#)
[Check Overlapping Documents for Conflicts](#)
[Apply Type System Updates](#)

Annotation Set Name	Annotator Name	Status	Action
pizza_anno_set	kpschlottter	IN PROGRESS	Annotate

Step 22 [Open](#) the *pizza_reviews.txt* document.

[← Back to Workspaces](#)

Assets
Documents
Entity Types
Relation Types
Dictionaries
Rule-based Model

Select Document

Annotation Set: kps_pizza_anno_set
Annotator: kps@de.ibm.com
Task: kps_anno_task

[Submit All Documents](#)

Showing 1-1 of 1


Document Name	Status	Last Modified	Actions
pizza_reviews.txt	Ready	Sep 15, 2019 9:53:53 AM	Open

Here you see that the pre-annotation with the pizza_types_dic dictionary already marked as PIZZA_TYPES.


pizza_reviews.txt


- I ordered a large **mediterranean** pizza from your Mopac location and the pizza was cold.
- Loved the Far West but the **White Pizza** was less desirable.
- I was in your Pflugerville store and had the Brazos with thin crust, which was awesome.
- I thought the Farmer's Market was supposed to have mushrooms on it, but mine did not.
- The Meaty Madness I had on Sunday was burnt, so I want my money back.
- The **Mediterranean** I had the other day was charred on the edges.
- The Brazos really needs thicker crust.

Step 23 Mark the remaining PIZZA_TYPES and the NEGATIVE_FEEDBACK entries.

Step 24 Click the  tab on the document panel. Click on the PIZZA_TYPE and on the associated NEGATIVE_FEEDBACK, and from the *Relations Type* side bar select *complaintWith*.

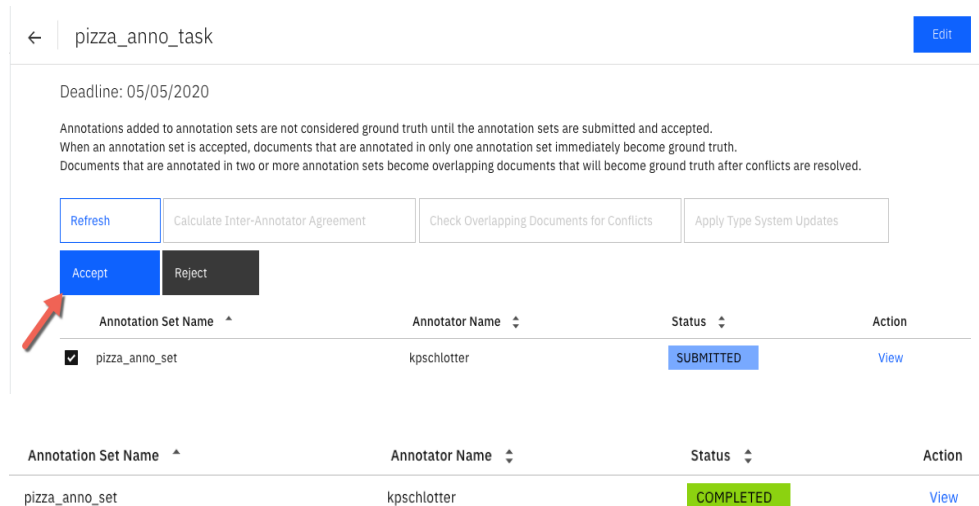
Click .


Step 25 Click  [Open document list](#) to go back to the document list. You see the above document with status in progress.

Step 26 On the *Select Document* panel click . Now all documents are in the  state.

Step 27 Click  to close the *Select Document* panel. Then click .



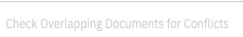

Step 28 On the Annotation Task **Accept** the submitted annotation set. Confirmation with .






← pizza_anno_task 


Deadline: 05/05/2020


Annotations added to annotation sets are not considered ground truth until the annotation sets are submitted and accepted. When an annotation set is accepted, documents that are annotated in only one annotation set immediately become ground truth. Documents that are annotated in two or more annotation sets become overlapping documents that will become ground truth after conflicts are resolved.



   

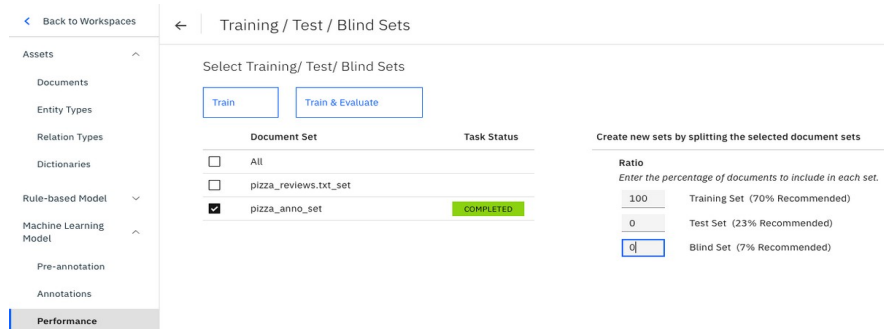
 

Annotation Set Name	Annotator Name	Status	Action
<input checked="" type="checkbox"/> pizza_anno_set	kpschlottner		View

Annotation Set Name	Annotator Name	Status	Action
pizza_anno_set	kpschlottner		View



Step 29 On Machine Learning Model → Performance click  to create the model from our annotations.


Step 30 On the *Training/Test/Blind Sets* page **select** the document set  100% *Training Set*, 0% *Test Set* and 0% *Blind Set*. (we only have 1 document) Then click .



← Back to Workspaces ← Training / Test / Blind Sets

Select Training/ Test/ Blind Sets

Document Set	Task Status
<input type="checkbox"/> All	
<input type="checkbox"/> pizza_reviews.txt_set	
<input checked="" type="checkbox"/> pizza_anno_set	

Create new sets by splitting the selected document sets

Ratio
Enter the percentage of documents to include in each set.

Ratio	Training Set (70% Recommended)	Test Set (23% Recommended)	Blind Set (7% Recommended)
100			
0			
0			

This process takes about 15 minutes.



Step 31 On *Machine Learning Model* → *Versions* click **Create Version** to create a version of your model for deployment. Enter a Description and click **OK**.

Back to Workspaces

Assets

Documents

Entity Types

Relation Types

Dictionaries

Rule-based Model

Machine Learning Model

Pre-annotation

Annotations

Performance

Versions

Versions

Machine Learning Model

Machine learning models, once trained, can be used to annotate new documents prior to human annotation.

[Go to Pre-annotation page](#)

Export the current version of your model to use in other Watson applications such as Watson Explorer.

[Export current model](#)

Version History and Deployment

Version	Base	Creation Date	Entity Scores	Relation Scores	Description	Action
1.1		Current Version	N/A	N/A		Create Version
1.0		05/05/2020	N/A	N/A	First Version for test deployment	Promote Delete Deploy

Step 32 Click [Deploy](#) to deploy your model. Select *Natural Language Understanding* service created in Lab 4. click *Next*.

Deploy Model v1.0

Select a service to deploy to.
All services require a subscription. [Learn more](#)

☒ Natural Language Understanding

☐ Discovery

[Cancel](#) [Next](#)

Step 33 Select your service and **click** *Deploy*.

Deploy Model v1.0

Deploying to Natural Language Understanding
You must have a subscription to the IBM Watson™ Natural Language Understanding service, and know the names of your IBM Cloud space and service instance. [Learn more](#)

IBM Cloud Information
IBM Cloud is the IBM cloud platform. Click [here](#) to open IBM Cloud and create an account or look up details for an existing service.

Region
Dallas

Resource group
Default

Service name
my-nlu

Cancel Deploy

Step 34 The deployment process starts. **Click** *OK*.

Deployment Started.

Deploying to Natural Language Understanding
It might take a few minutes for publishing and deployment to complete, and for this model to be available to your applications.
You can view your deployed models, withdraw a model from deployment, or deploy a newer version.

Model ID: ed303d2

You can [view documentation](#) to learn how to implement the deployed model into your application.

OK

You then have a deployment history. Here you can always see the Id of the deployed model.

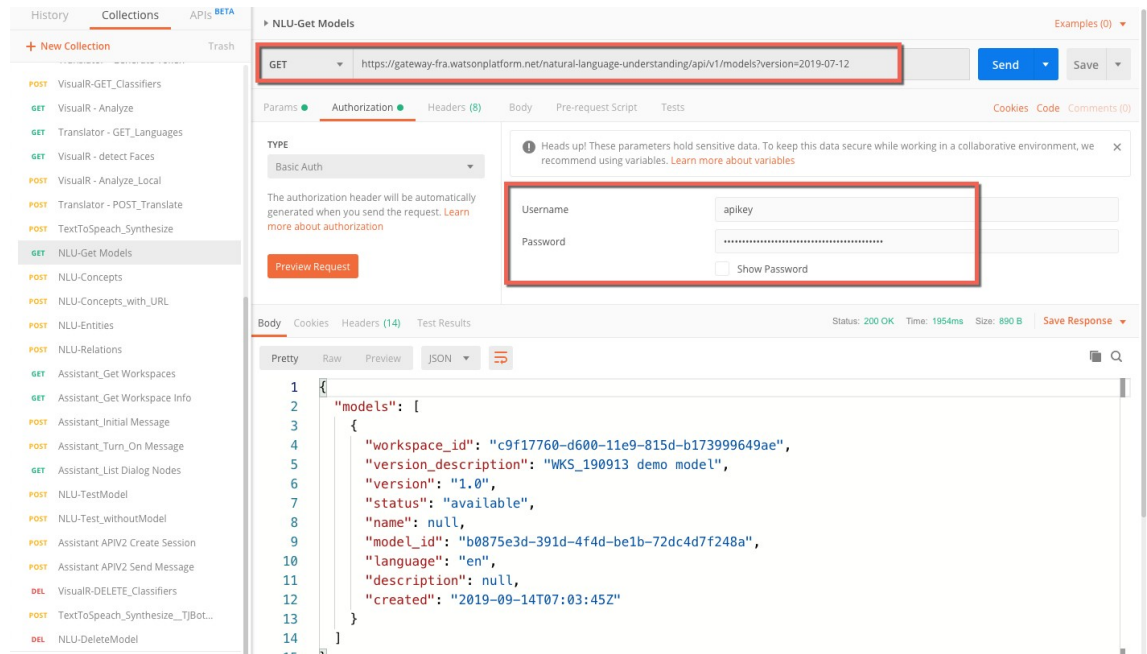
Version History and Deployment

Version	Base	Creation Date	Entity Scores	Relation Scores	Description	Action
1.1	Current Version		N/A	N/A		Create Version
1.0		09/14/2019	N/A	N/A	WKS_190913 demo model	Promote Delete Deploy
Deployed Models (1)						
			Model ID: b0875e3d-391d-4f4d-be1b-72dc4d7f248a	Service ID: d373e60d-dcd1-4efd-b305-2eac19e1c79c	Undeploy Status	

Section 2: Testing the model with [Postman](#) (See [Lab 4](#))

Step 35 List your deployed models. You need Basic Auth (username and password from your Natural Language Understanding Service from your Bluemix account and the following URL: (See Lab 4 Step 6)

url: **<yourUrlFromStep3c>/v1/models**
 params: version 2019-07-12



Step 36 Test the model with the Basic Auth information from previous step and in the *Headers* section add

key: Content-type value: application/json

In the *URL* change models to analyze

In the Body section enter the text you want to analyze and specify the service feature(s) you want to apply.

```

{ "text": "I ordered a Mediterranean from your mopac location on July 25
and the cheese was stuck to the top of the box.",
  "features": { "relations": { "model": "<your model ID goes here>" } } }
  
```

Step 37 With the following result:

POST <https://gateway.watsonplatform.net/natural-language-understanding/api/v1/analyze?version=2019-07-12> Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Cookies Code Comments (0)

none form-data x-www-form-urlencoded raw binary GraphQL BETA JSON (application/json) Beautify

```

1 { "text": "NCR, which counts IBM founder Thomas Watson as one of its early employees, said its products and services account for
2   more than $400 billion in annual commerce and 23 billion consumer serf-service transactions.",
3   "features": {
4     "relations": {
5       "model": "825a30e9-c3ac-45ec-8bf5-311c981db164"
6     },
7     "entities": {
8       "model": "825a30e9-c3ac-45ec-8bf5-311c981db164"
9     }
10  }

```

Body Cookies Headers (19) Test Results Status: 200 OK Time: 501ms Size: 1.13 KB Save Response

Pretty Raw Preview JSON

```

1 {
2   "usage": {
3     "text_units": 1,
4     "text_characters": 210,
5     "features": 2
6   },
7   "relations": [
8     {
9       "type": "founderOf",
10      "sentence": "NCR, which counts IBM founder Thomas Watson as one of its early employees, said its products and services
11        account for more than $400 billion in annual commerce and 23 billion consumer serf-service transactions.",
12      "score": 0.956072,
13      "arguments": [
14        {
15          "text": "Thomas Watson",
16          "location": [
17            30,
18            43
19          ],
20          "entities": [
21            {
22              "type": "PERSON",
23              "text": "Thomas Watson",
24              "disambiguation": {
25                "subtype": [
26                  "NONE"
27                ]
28              }
29            }
30          ]
31        },
32        {
33          "text": "IBM",
34          "location": [
35            18,
36            21

```

Bootcamp

Section 3: Use the Model in the App from Lab 4

In Lab 4 about the Watson Natural Language Understanding service we created a Node.js web application ([Github repository](#)) that retrieves the deployed model at application startup and makes it available for text analysis. In the model we have created entities and relations and these two features can be applied to the text analysis. Test the text from **Step 36** with and without the model selected.

IBM EAG NLU-Tester

Analyze text with IBM Watson Natural Language Understanding service

☐ url ☒ text

☐ categories ☐ concepts ☐ emotion ☒ entities ☐ keywords ☐ metadata ☒ relations ☐ semantic_roles

☒ Model: 9162956-7d6036

☒ Entities

☒ Relations

Enter text or url

Enter text or an URL.

Result:

IBM Ecosystem Advocacy Group – 2018

Here you can test the same text sample as with Postman in Section 2.

IBM EAG NLU-Tester

Analyze text with IBM Watson Natural Language Understanding service

☐ url ☒ text

☐ categories ☐ concepts ☐ emotion ☒ entities ☐ keywords ☐ metadata ☒ relations ☐ semantic_roles

☒ Model: 91629562-a55e-4a99-a409-6a4abb7d6036

☒ Entities

☒ Relations

Enter text or url

I ordered a Mediterranean from your mopac location on July 25 and the cheese was stuck to the top of the box.

Result:

```
{
  "usage": {
    "text_units": 1,
    "text_characters": 109,
    "features": 2
  },
  "relations": [
    {
      "type": "complaintWith",
      "sentence": "I ordered a Mediterranean from your mopac location on July 25 and the cheese was stuck to the top of the box.",
      "score": 0.981304,
      "arguments": [
        {
          "text": "Mediterranean",
          "location": [
            12,
            25
          ],
          "entities": [
            {
              "type": "PIZZA_TYPE",
              "text": "Mediterranean",
              "disambiguation": {
                "subtype": [
                  "NONE"
                ]
              }
            }
          ]
        }
      ]
    }
  ]
}
```