

# Watson Hackathon Training for Morgan Stanley Oct 4 2018

David Carew  
[carew@us.ibm.com](mailto:carew@us.ibm.com)

Javier Torres  
[jrtorres@us.ibm.com](mailto:jrtorres@us.ibm.com)

IBM  
**CODE**  
\_\_\_\_\_

# Agenda

- Introductions
- Watson Studio
- Watson Natural Language Understanding and Watson Discovery
- Watson Assistant
- Watson Visual Recognition
- Wrap up and Q&A

Watson Studio

IBM  
CODE

## **What is Watson Studio ?**

Watson Studio is a suite of tools for data scientists, application developers and subject matter experts to collaboratively and easily work with data and use that data to build, train and deploy models at scale.

# Watson Studio

Built for AI teams – enabling team productivity and collaboration



**Tanya**  
Domain Expert

**Her Job:**

To transfer knowledge to Watson for a successful user experience.

**What she does:**

- Range of domain knowledge and uses that to teach Watson and develop custom models
- As Tanya gains more experience she optimizes her knowledge to teach Watson to design better end-user experiences.

**Sometimes known as:**

Subject matter expert, content strategist.



**Mike**  
Data Scientist

**His Job:**

Transform data into knowledge for solving business problems.

**What he does:**

- Runs experiments to build custom models that solve business problems.
- Use techniques such as Machine Learning or Deep Learning and works with Tanya to validate success of trained models.

**Sometimes known as:**

ML/DL engineer, Modeler, Data Miner



**Ed**  
Data Engineer

**His Job:**

Architects how data is organized and ensures operability

**What he does:**

- Builds data infrastructure and ETL pipelines. Works with Spark, Hadoop, and HDFS.
- Works with data scientist to transform research models into production quality systems.

**Sometimes known as:**

Data infrastructure engineer



**Deb**  
The Developer

**Her Job:**

Builds AI application that meet the requirements of the business.

**What she does:**

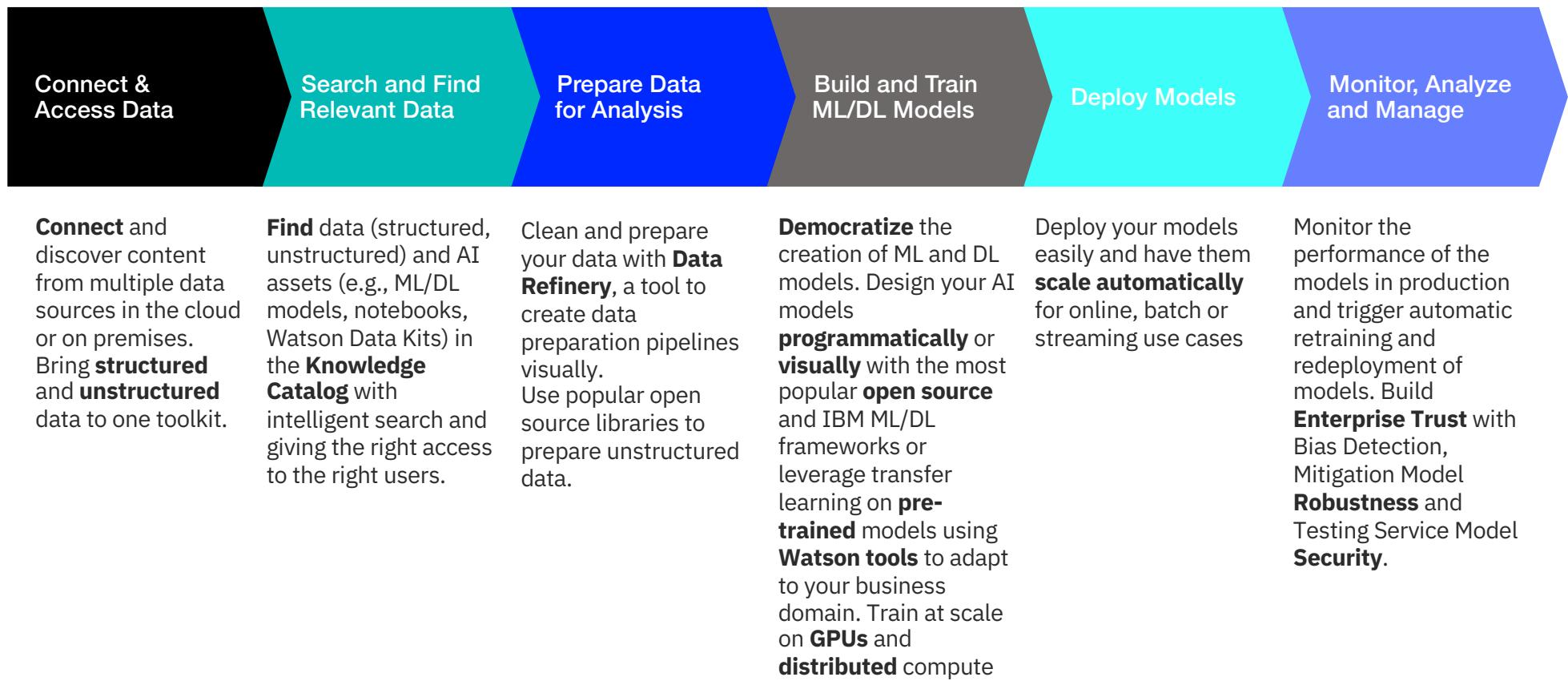
- Starts PoCs which includes gathering content, dialog building and model training
- Focus is on app building for the team or company to use. Will handle ML Ops as needed

**Sometimes known as:**

Front-end, back-end, full stack, mobile or low-code developer

# Watson Studio

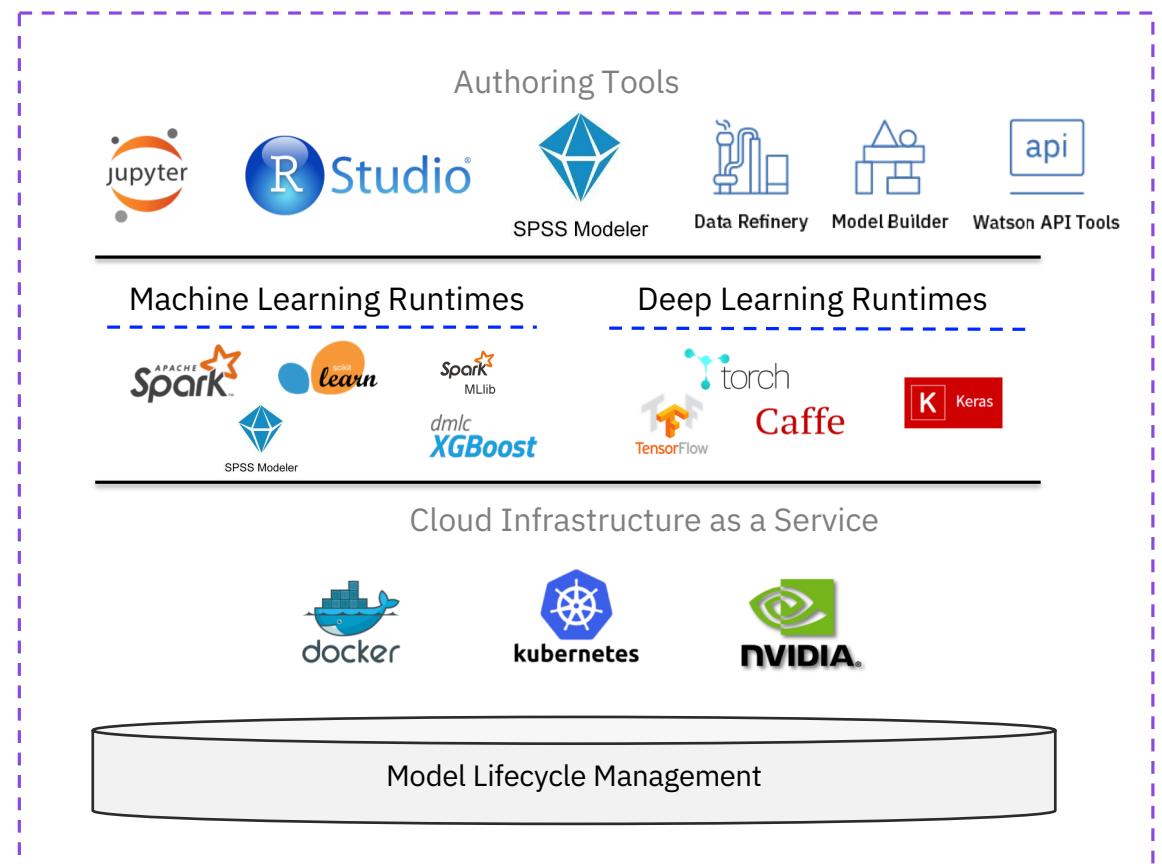
## Supporting the end-to-end AI workflow



# Watson Studio

## Tools for supporting the end-to-end AI workflow

- Create, collaborate, deploy, and monitor
  - Best of breed open source & IBM tools
  - Code (R, Python or Scala) and no-code/visual modeling tools
- 
- Most popular open source frameworks
  - IBM best-in-class frameworks
- 
- Fully managed service
  - Container-based resource management
  - Elastic pay as you go cpu/gpu power



# Resources

- The MNIST Flow Editor demo
  - <https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-canvas-nnd-mnist-tutorial.html?audience=wdp>
- MNIST Demo with Python Notebook
  - <https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-python-mnist-tutorial.html?audience=wdp>
- Credit card fraud detection hands on exercise
  - [Predicting Credit Card Using Keras in Watson Studio](#)
- Oil price prediction hands on exercises
  - [Predicting Oil Prices Using an RNN with Watson Studio](#)
  - [Hyperparameter optimization, model deployment and scoring with Watson Studio](#)
- IBM Code Patterns with Watson Studio
  - [Create and deploy a scoring model to predict heartrate failure](#)
  - [Determine trending topics with clickstream analysis](#)
  - [Analyze traffic data from the city of San Francisco](#)

# Demo

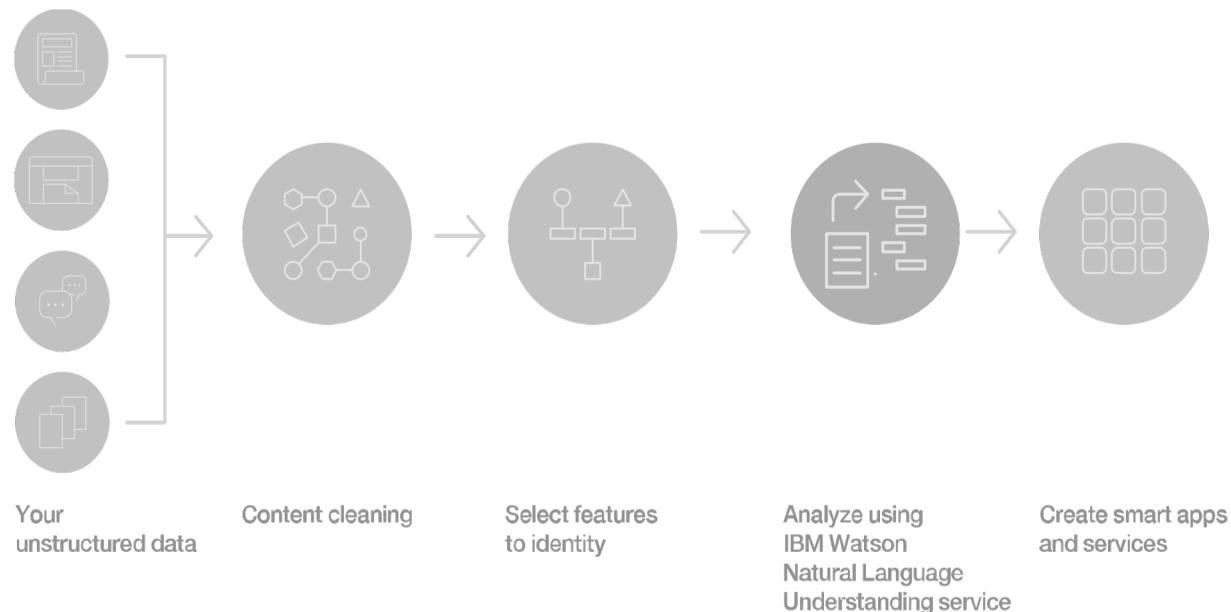
Watson Natural  
Language  
Understanding

IBM  
CODE

# Watson Natural Language Understanding [NLU]



A suite of natural language processing capabilities to analyze text and extract meta-data from content with options for customization to specific industries and domains.



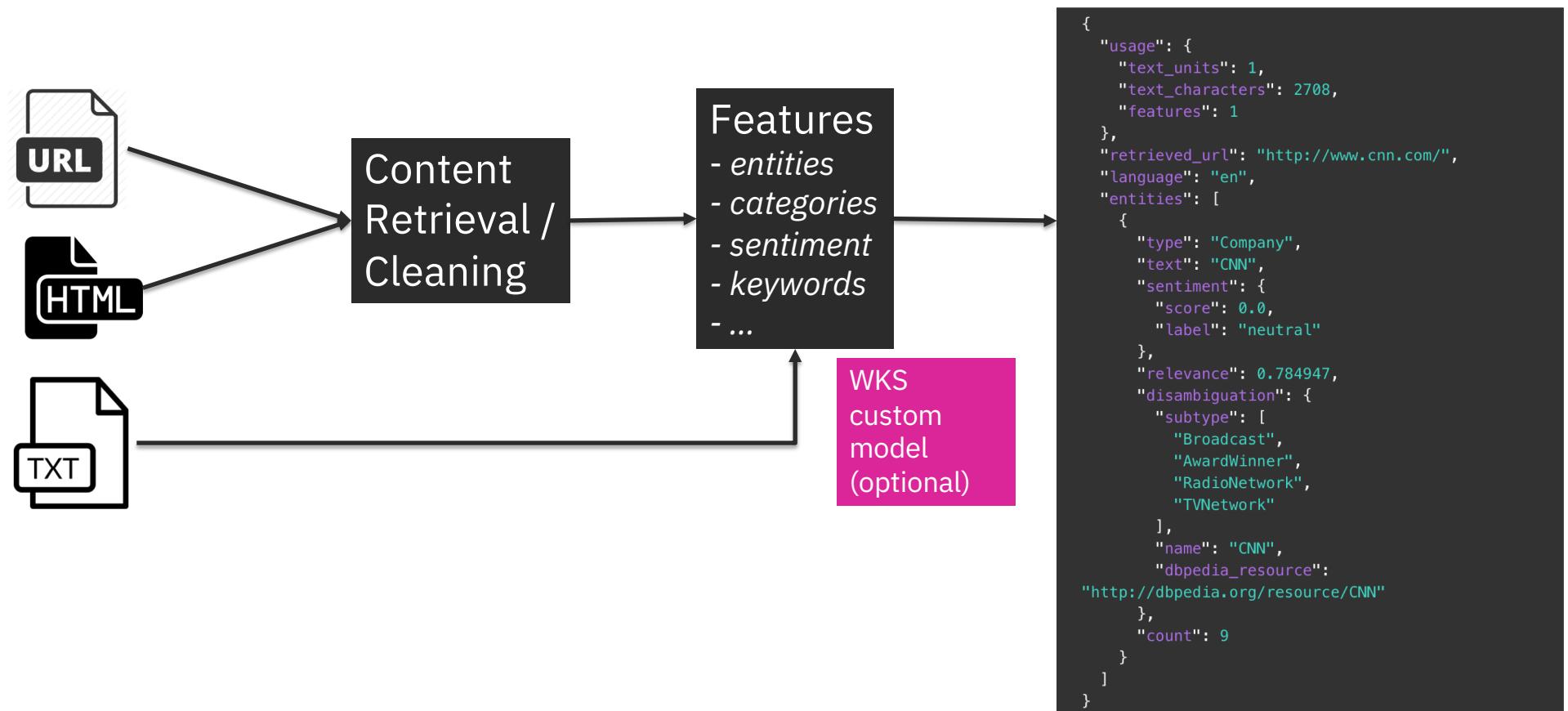
**Content Recommendation:** How can I recommend content that readers might like?

**Advertising Optimization:** Where and how should I place my ads to the right audience?

**Audience Segmentation:** How can I segment customers to tailor strategies for different audiences?

**Voice of Customer Analysis:** How can I address customer feedback and improve customer experience?

# NLU Flow



# NLU Features

Concepts	Identify general concepts in your content. Useful for Topic Detection.
Entities*	Detect important people, locations, companies and other types of entities in your content.
Keywords	Determine the most important keywords in your content.
Categories	Categorize your content into a hierarchical 5-level taxonomy.
Sentiment	Determine whether your content conveys positive or negative sentiment.
Emotion	Detect emotions such as anger, disgust, fear, joy or sadness that are conveyed by your content.
Relations*	Identify relationships between entities in your content.
Semantic role	Identify the subject-action-object pairs in sentences. Useful for event or generic relation detection.
Metadata	Get author information, publication date, title, etc. of your URL or HTML content.

(\*) Entities and Relations can be customized using custom annotation models developed in IBM Watson Knowledge Studio.

Entity types				
Entity type	LEVEL 1	LEVEL 2	LEVEL 3	LEV
Anatomy	art and entertainment	books and literature	best-sellers	Hashtag
Award	art and entertainment	books and literature	children's books	IPAddress
Broadcaster	art and entertainment	books and literature	e-books	.5
Company	art and entertainment	books and literature	fan fiction	JobTitle
Crime	art and entertainment	books and literature	magazines	Location
Drug	art and entertainment	books and literature	mythology	
EmailAddress	art and entertainment	books and literature	non-fiction	
Facility	art and entertainment	books and literature	poetry	
GeographicFeature				
HealthCondition				
Hashtag				
IPAddress				
JobTitle				
Location				

IBM Code

# Watson Discovery



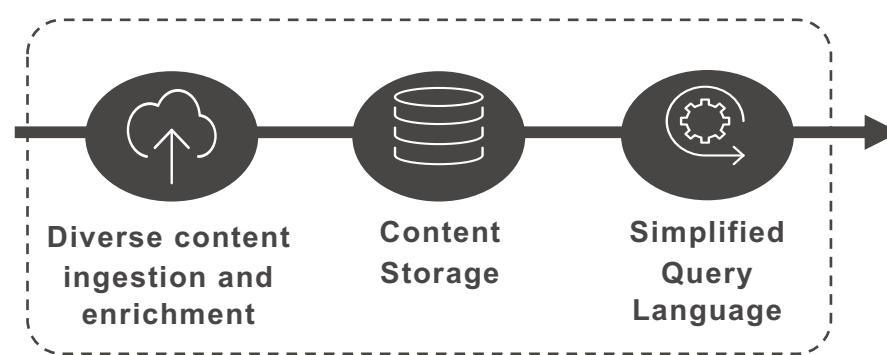
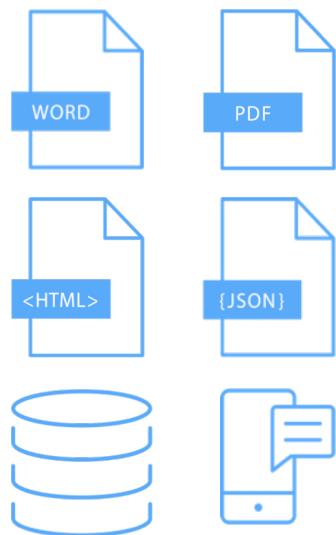
**Discovery**

Lite • IBM

Add a cognitive search and content analytics engine to applications.

# Turn Data into Insights

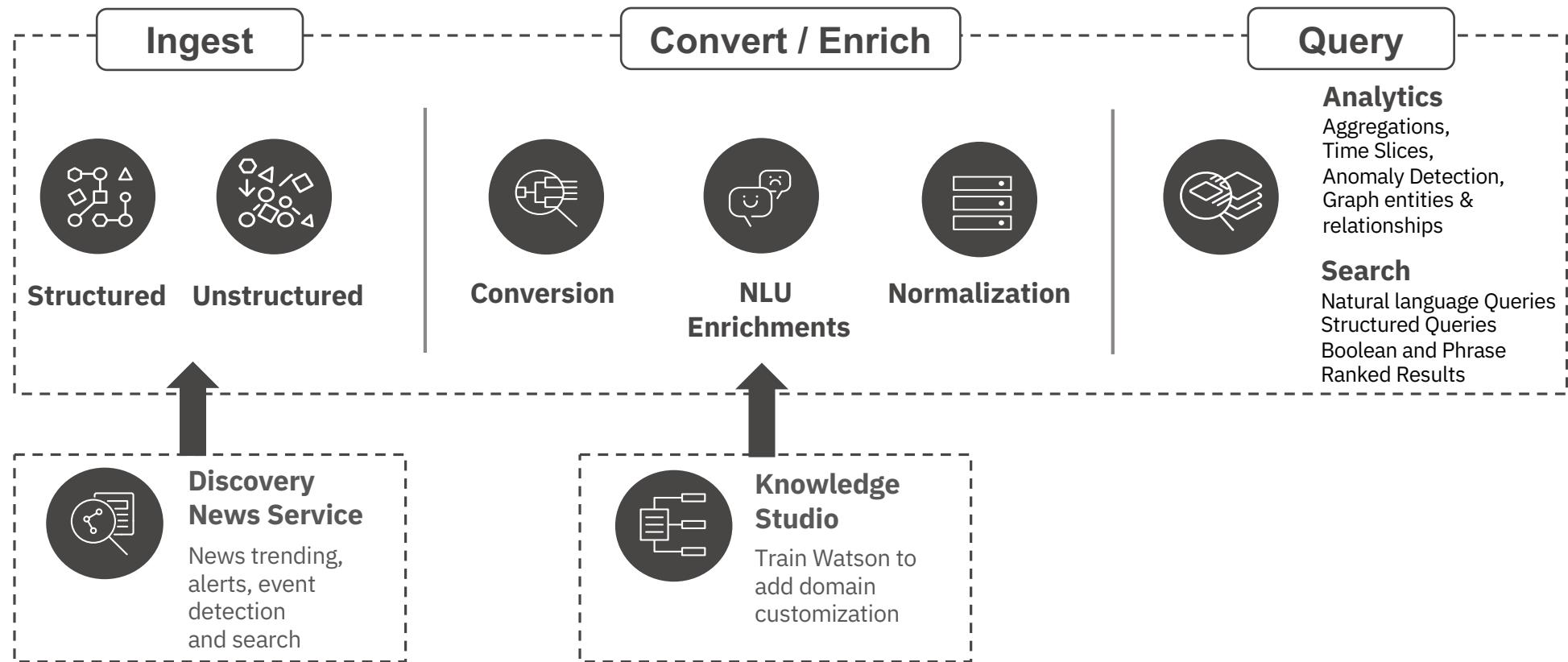
## Documents / Data



## Answers / Trends / Insights

- “What are the top terms mentioned by customers in support interactions”
- “How do I setup a new connection for a customer with a locked accounts”
- “How many times was my campaign mentioned over the last 30 days”
- “What proof is needed to terminate a housing contract?”

# AI Powered Search and Analytics



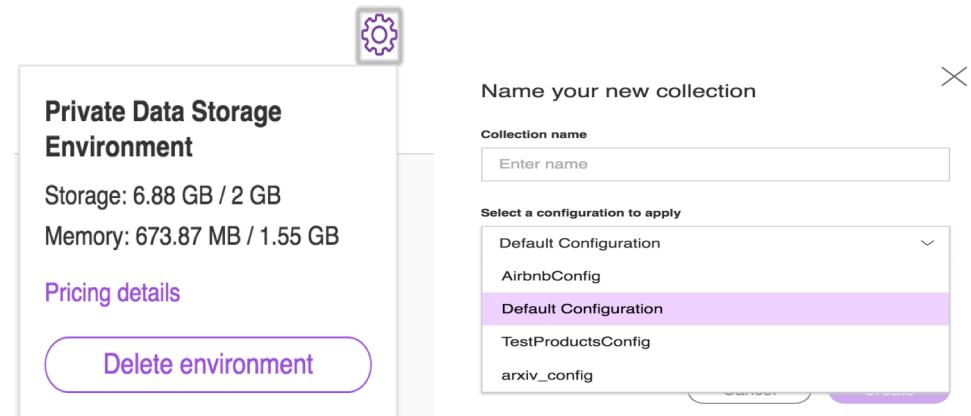
# Environment & Collection

## Environment

- Defines storage allocation for content.

## Collection

- Grouping of content within the environment.
- Assigns a configuration and language to use for documents.



The screenshot shows the API documentation for the '/v1/environments/{environment\_id}/collections' endpoint. It includes two tabs: 'GET /v1/environments/{environment\_id}/collections' and 'POST /v1/environments/{environment\_id}/collections'. The 'GET' tab shows a 'Response Class (Status 201)' with the message 'Collection successfully created.' Below it are 'Model', 'Example', and 'Value' sections. The 'Value' section contains a JSON object representing a collection document:

```
{  
  "collection_id": "800e58e4-198d-45eb-be87-74e1d6df4e96",  
  "name": "test-collection",  
  "configuration_id": "3c4fff84-1500-455c-b125-eaa2d319f6d3",  
  "language": "de",  
  "status": "active",  
  "description": "A test collection to show as an example",  
  "created": "2017-07-14T12:55:40.652Z",  
  "updated": "2017-07-14T12:55:40.652Z",  
  "document_counts": {  
    "available": 0  
  }  
}
```

# Configuration

Defines the document conversion process, data normalization and NLU enrichments applied.

- Conversion rules based on font styles and sizes.

The screenshot displays the configuration interface for an IBM Watson API endpoint. It is divided into three main sections: **Convert**, **Enrich**, and **Normalize**.

**Convert:** This section is for PDF conversion. It includes tabs for PDF, WORD, HTML, and JSON. A sub-section titled "Specify font properties" allows users to define rules for H1 headings based on font size ranges, styles (regular, italic, bold, bold italic, or any style), and font names (e.g., Times New Roman). There is also a "Font size range" input field with "min" and "max" fields.

**Normalize:** This section is for JSON normalization. It includes tabs for PDF, WORD, HTML, and JSON. A sub-section titled "Clean up your JSON" allows users to set up rules for document cleanup. Another section titled "Move, merge, copy or remove fields" provides options for managing JSON field structures, with a dropdown menu showing "remove", "remove", "copy", "move", and "merge" options, and an input field for "Enter field name".

**Enrich:** This section is for adding NLU enrichments. It features a title input field with suggestions for "entity", "keyword", and "concept". Below are eight enrichment components:

- Entity Extraction:** Extracts people, companies, organizations, cities, geographic features, and more from the field. Includes a "Learn more" link and an "Added!" button.
- Keyword Extraction:** Determines important keywords in the field, ranks them, and optionally detects the sentiment. Includes a "Learn more" link and an "Added!" button.
- Taxonomy Classification:** Classifies the field into a hierarchy of categories that's five levels deep. Includes a "Learn more" link and an "Add" button.
- Concept Tagging:** Identifies general concepts that aren't necessarily directly referenced in the field. Includes a "Learn more" link and an "Added!" button.
- Relation Extraction:** Parses sentences into subject, action, and object form and returns additional semantic information. Includes a "Learn more" link and an "Add" button.
- Sentiment Analysis:** Identifies the overall positive or negative sentiment within the field. Includes a "Learn more" link and an "Add" button.
- Emotion Analysis:** Analyzes the emotions (anger, disgust, fear, joy, and sadness) in the field. Includes a "Learn more" link and an "Add" button.

# Ingesting Content

Supported document types:

- PDF, Word, HTML, JSON

Three mechanisms for upload

- [Discovery tooling](#)
- [API](#)
- [Data Crawler](#)

Add data to this collection

Drag and drop your documents here  
or browse from computer

Maximum file size: 50MB  
File format: PDF, WORD, HTML, and JSON

Note: Each file name should be unique. If two files have the same name, the original will be overwritten when a newer version is uploaded.

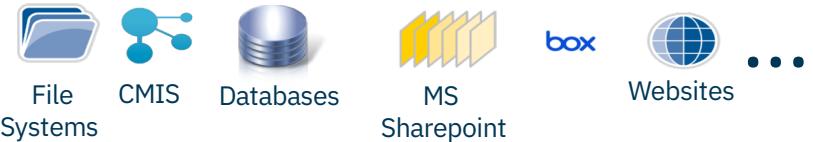
Documents available	24310
Documents processing	0
Documents failed	459

[View errors and warnings](#)

Documents

Show/Hide | List Operations | Expand Operations

POST	/v1/environments/{environment_id}/collections/{collection_id}/documents	Add a document
DELETE	/v1/environments/{environment_id}/collections/{collection_id}/documents/{document_id}	Delete a document
GET	/v1/environments/{environment_id}/collections/{collection_id}/documents/{document_id}	Get document details
POST	/v1/environments/{environment_id}/collections/{collection_id}/documents/{document_id}	Update a document



# Queries

Queries support structured/targeted searching or full text natural language search.

- **Natural language supports semantic search or machine learning model to rank the most relevant docs**

## Aggregation

- Timeslice – segmented counts over time (Anomaly)
- Top\_hits – return best ranked hits
- Term – return counts of field/term specified
- Min/max/avg – returns the <operator> for a structured field

## Return documents / passages



enriched\_text.keywords.text:IBM,enriched\_text.keywords.text:!Watson  
→ Documents that contain the keyword IBM, but not the keyword Watson.

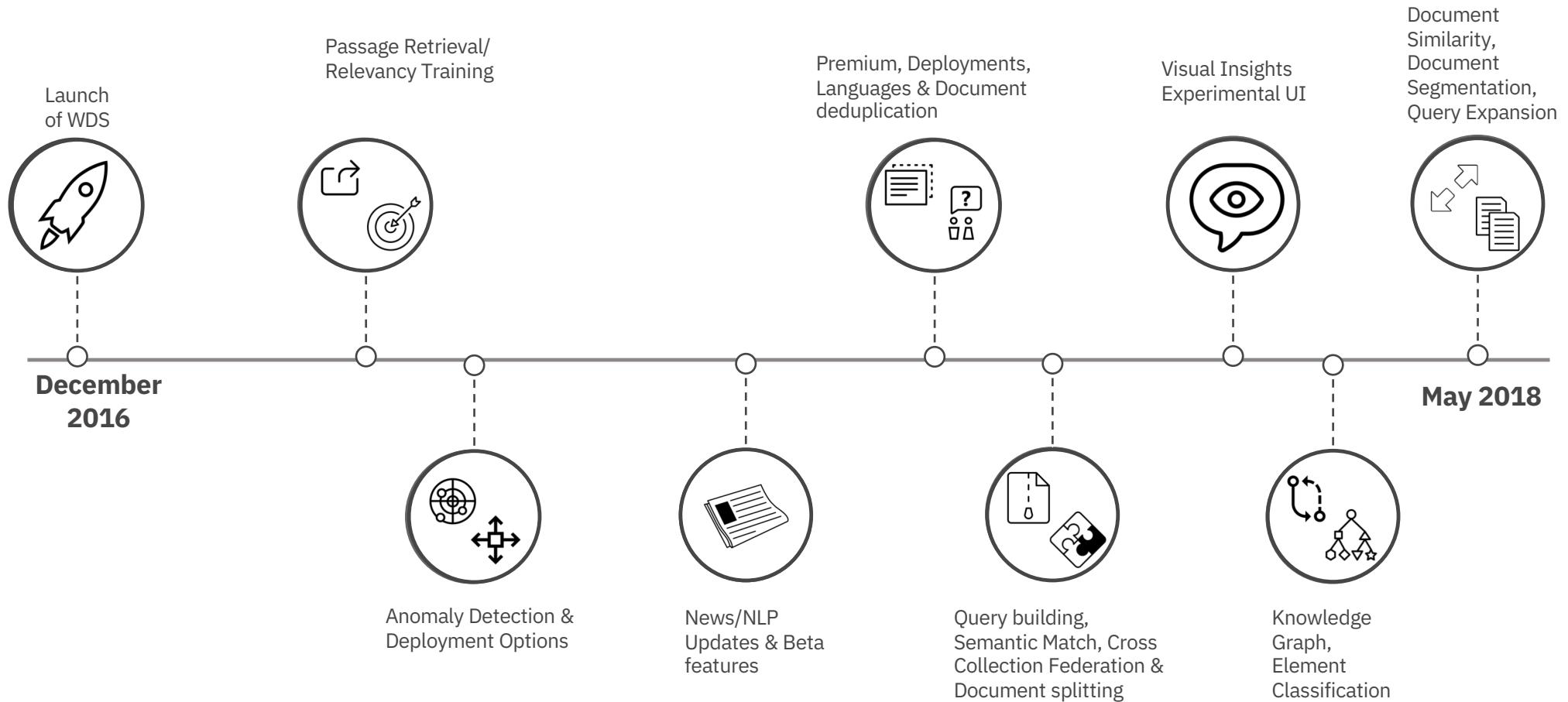


term(enriched\_text.concepts.text,count:10) → top 10 concepts in corpus

timeslice(field:review\_date,interval:1day,anomaly:true)

```
✓ {  
  "matching_results": 214,  
  "passages": [],  
  "results": [  
    {  
      "id": "b9a59485cb21282dc476c9be57b96dc7",  
      "result_metadata": {},  
      "body": "The heater acts as a starting aid by warming the engine coolant. This allows the climate control system to respond quickly. The equipment includes a heater element (installed in the engine block) and a wire harness. You can connect the system to a grounded 12volt AC electrical source.  
The engine block heater plug is located in a housing in the left fender panel. Open the hinged circular door and make sure the receptacle terminals are clean and dry prior to use. Clean them with a dry cloth if necessary.  
The heater uses 0.4 to 1.0 kilowatt-hours of energy per hour of use. The system does not have a thermostat. It will achieve maximum temperature after approximately three hours of operation. If you use the heater longer than three hours, this will not improve system performance and will use unnecessary electricity.",  
      "enriched_body": {},  
      "enriched_title": {},  
      "contentHtml": "<p>The heater acts as a starting aid by warming the engine coolant. This allows the climate control system to respond quickly. The equipment includes a heater element (installed in the engine block) and a wire harness. You can connect the system to a"
```

# Functionality Overview



# Resources

- Watson NLU hands on exercise
  - <https://github.com/djccarew/watson-toneanalyzer-nlu-lab>
- Watson Discovery hands on exercise
  - <https://github.com/indrann/watson-discovery-analyze-data-breaches>
- IBM Code Patterns with Watson Discovery and/or NLU
  - [Mine data breaches for insights](#)
  - [Create a news alerting app](#)
  - [Extend Watson text classification](#)

# Demo

Watson Assistant

IBM  
CODE

# What is Watson Assistant ?

Watson Assistant is a cloud based or on premise platform that allows developers and non-technical users to collaborate on building conversational AI solutions.

It comes with a graphical UI, and powerful NLP to allow the rapid creation of anything from simple chatbots to complex enterprise grade solutions for customer service and more.

# Creating a chatbot with Watson Assistant

1. Create an instance of Watson Assistant
2. Launch the Watson Assistant tool
  - Create Intents, Entities and Dialogs
  - Test
3. Connect to application using API
  - API call sends in user input + current state, returns bot output and updated state

# Intents

An *intent* is the goal or purpose of the user's input. Adding examples to intents helps your virtual assistant understand different ways in which people would say them.

Provide at least five examples for each intent.

← | #book\_reservation

**Intent name**  
#book\_reservation

**Description**  
Add a description to this intent

**Add user examples**  
| Add user examples to this intent

**Add example**

User examples (5) ▾

- book a reservation
- make a reservation
- reserve a table
- schedule a reservation
- secure a reservation

# Content Catalog

Get started faster by adding existing intents from the content catalog. These intents are trained on common questions that users may ask.

Intents Entities Dialog **Content Catalog**

Get started faster by adding existing intents from the content catalog. These intents are trained on common questions that users may ask.

Category	Description	Intents	
Banking	Basic transactions for a banking use case.	13	<a href="#">+ Add to workspace</a>
Bot Control	Functions that allow navigation within a conversation.	9	<a href="#">+ Add to workspace</a>
Customer Care	Understand and assist customers with information about themselves and your business.	18	<a href="#">+ Add to workspace</a>
eCommerce	Payment, billing, and basic management tasks for orders.	14	<a href="#">+ Add to workspace</a>
General	General conversation topics most users ask.	10	<a href="#">+ Add to workspace</a>
Insurance	Issues related to insurance policies and claims.	12	<a href="#">+ Add to workspace</a>
Telco	Questions and issues related to a user's telephony service, device, and plan.	21	<a href="#">+ Add to workspace</a>
Utilities	Help a user with utility emergencies and their utility service.	10	<a href="#">+ Add to workspace</a>

# Entities

An *entity* is a portion of the user's input that you can use to provide a different response to a particular intent. Adding values and synonyms to entities helps your virtual assistant learn and understand important details that your users mention.

The screenshot shows the IBM Watson Assistant interface for managing entities. At the top, there is a back arrow and the entity name '@cuisine'. Below this, the 'Entity name' is set to '@cuisine'. A 'Value name' field contains 'Enter value', with a 'Synonyms' dropdown menu open, showing 'Add synonym...' and a '+' icon. There are two buttons: 'Add value' and 'Show recommendations'. Below these, there are tabs for 'Dictionary' and 'Annotation BETA', with 'Dictionary' selected. A table lists five entity values: 'american', 'chinese', 'french', 'indian', and 'mexican', each with a checkbox, a 'Type' column (all listed as 'Synonyms'), and a 'Details' column showing their respective synonyms and examples like 'Kerbey Lane Cafe', 'Hao Hao', 'La Cafe', 'Bombay Grill', and 'Serrano's'.

Value	Type	Details
american	Synonyms	Kerbey Lane Cafe
chinese	Synonyms	Hao Hao
french	Synonyms	La Cafe
indian	Synonyms	Bombay Grill
mexican	Synonyms	Serrano's

# System Entities

System entities are common entities created by IBM that could be used across any use case. They are ready to use as soon as you add them.

Intents    **Entities**    Dialog    Content Catalog

My entities    **System entities**

> <a href="#">@sys-currency</a>	Extracts currency values from user examples including the amount and the unit. (20 cents)	<input type="checkbox"/> Off
> <a href="#">@sys-date</a>	Extracts date mentions (Friday)	<input checked="" type="checkbox"/> On
> <a href="#">@sys-location <small>BETA</small></a>	The @sys-location system entity extracts place names (country, state/province, city, town, etc.) from the user's input. (Boston)	<input type="checkbox"/> Off
> <a href="#">@sys-number</a>	Extracts numbers mentioned from user examples as digits or written as numbers. (21)	<input checked="" type="checkbox"/> On
> <a href="#">@sys-percentage</a>	Extracts amounts from user examples including the number and the % sign. (15%)	<input type="checkbox"/> Off
> <a href="#">@sys-person <small>BETA</small></a>	The @sys-person system entity extracts names from the user's input. (Anna)	<input checked="" type="checkbox"/> On
> <a href="#">@sys-time</a>	Extracts time mentions (at 10)	<input checked="" type="checkbox"/> On

# Dialog

A dialog uses intents, entities, and context from your application to define a response to each user's input.

Creating a dialog defines how your virtual assistant will respond to what your users are saying.

The screenshot shows a user interface for managing a dialog. At the top, there are tabs: Intents, Entities, Dialog (which is highlighted in blue), and Content Catalog. Below the tabs are three buttons: Add node (blue), Add child node (light gray), and Add folder (dark gray). The main area displays a list of dialog nodes under the heading "HungerBot".

- Welcome**  
welcome  
1 Response / 0 Context set / Does not return
- Book reservation**  
#book\_reservation  
1 Response / 5 Context set / 5 Slots / Does not return
- Anything else**  
anything\_else  
1 Response / 0 Context set / Does not return

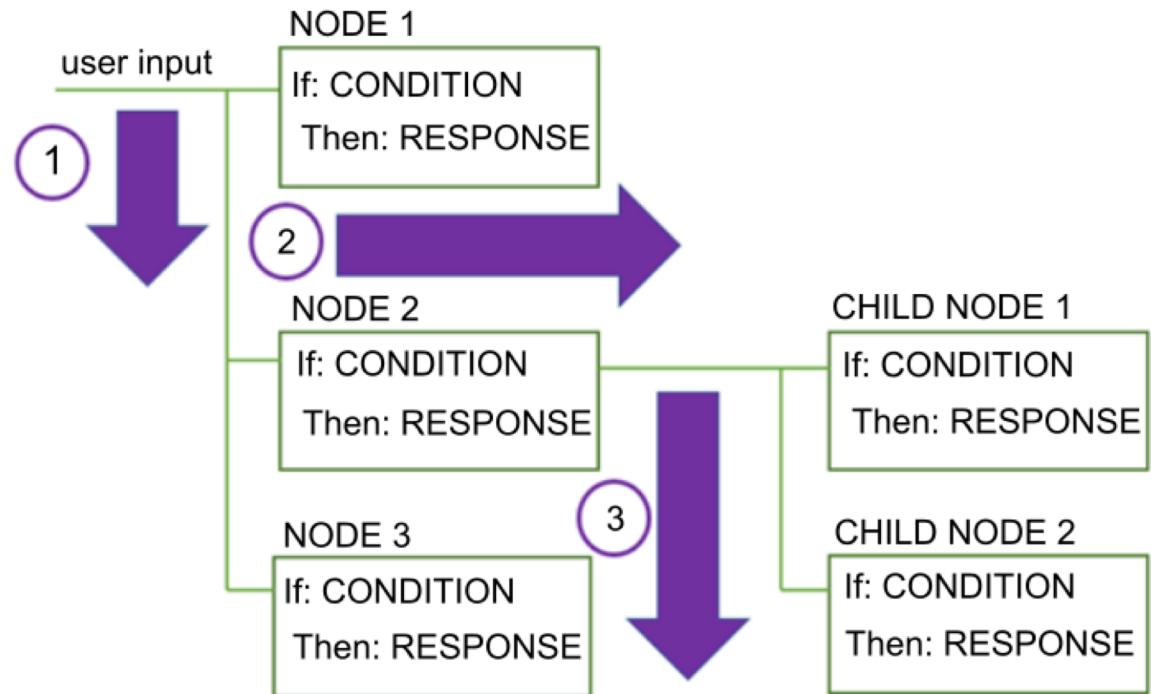
Each node entry includes a vertical ellipsis icon on the right side.

# Dialog

Dialog is processed by the service from the first node in the tree to the last.

If the service finds a condition that is met, it triggers that node.

It then moves along the triggered node to check the user input against any child node conditions.



# Slots

Add slots to a dialog node to gather multiple pieces of information from a user within that node.

Slots collect information at the users' pace. Details the user provides upfront are saved, and the service asks only for the details they do not.

## Book Reservation

Then check for:

	Check for	Save it as	If not present, ask
1	@cuisine	\$cuisine	What type of cuisine would you like?
2	@sys-date	\$date	What day would you like to eat?
3	@sys-time	\$time	What time would you like to eat?
4	@sys-number	\$number	How many people will be eating?

 Add slot

# Resources

- HungerBot hands on lab
  - <https://github.com/djccarew/watson-assistant-lab>
- Pattern based chatbot examples
  - <https://github.com/watson-developer-cloud/community/tree/master/watson-assistant>
- IBM Code Patterns with chatbots
  - [Create a cognitive banking chatbot](#)
  - [Create a cognitive retail chatbot with a Slack frontend](#)

# Demo

Watson Visual  
Recognition

IBM  
CODE



[developer.ibm.com/code](https://developer.ibm.com/code)

## What is Watson Visual Recognition ?

A cloud based service that uses deep learning algorithms to analyze images for scenes, objects, faces, and other content.

- The response includes keywords that provide information about the content

### **Major functionality**

- Image classify with keywords based on image recognized
  - Provides a default classifier that recognizes a wide range of images
  - Train your own classifiers with your images based on your own business scenarios
    - Custom models can exported as CoreML models
- Detect human faces giving age range and gender

### **Tooling**

- A Watson Studio based tool for classifying images and training your own classifiers

# Resources

- Company Logo hands on lab
  - <https://github.com/djccarew/watson-vr-logo-lab>
- Wedding detection hands on lab
  - <https://github.com/djccarew/watson-vr-custom-classifier-lab>
- IBM Code Patterns using Watson Visual Recognition
  - [Deploy a Core ML model with Watson Visual Recognition](#)
  - [Developing an image classifier using Watson Visual Recognition on Watson Studio \(with Android client\)](#)
  - [Enrich multimedia files using Watson services](#)

# Demo

## Wrap up and Q&A

