

NET-A-PORTER

Customer Service Chatbot

System Manual

1	Introduction	2
2	IBM Watson Conversation	2
3	JSON to CSV Scripts	2
4	IBM Watson Retrieve & Rank	2
5	IBM Watson Document Conversion	3
6	IBM Watson Language Translator	3
7	Amazon Web Services (AWS) Lambda	3
8	Amazon Web Services (AWS) API Gateway	3
9	Front-End Application	4

1 Introduction

This manual is dedicated to explain how the chatbot service is developed and how to deploy the service. The chatbot system architecture consist of several subsystems, each of which is independent from the others so as to allow additional functionalities to be added without dealing with dependencies. The main subsystems are IBM Watson Conversation, IBM Retrieve & Rank, IBM Watson Language Translator, Amazon Web Services (AWS) and the Front-End application that demos the service.

2 IBM Watson Conversation

Watson Conversation handles the intents and entities of the user query in order to understand what information the user is trying to get.

Under services.js of the source code repository, a username, password, workspace ID and version date is required. The information can be obtained under the service credentials, that is available when a new Conversation workspace is created.

An export of the workspace (JSON file) can be found under /watson-exports/conversation folder. This workspace can easily be imported into a new workspace to populate the intents, entities, and dialogs that we have created, and no additional work is needed.

3 JSON to CSV Scripts

NET-A-PORTER is a fashion company, which means that changes to its product catalogue can be frequent. These changes would mainly concern the category, brands, colour and sizes. To avoid manual updates to the Conversation workspace, a Java programme is written to convert the JSON responses from NET-A-PORTER APIs to CSV files, which can then be imported to the workspace.

In generating the CSV files, variations in entity values are implemented so as to capture a wider set of user queries. For example, a “Dr. Jackson’s” entity value is accompanied with “Dr Jackson’s”, “Dr Jackson”, etc. In addition, misspellings are also implemented by scrapping results from an online typo generator. An example is “Gucci” is accompanied with “Guucci”, “Guccii”, etc.

4 IBM Watson Retrieve & Rank

Watson Retrieve & Rank (R&R) is the backbone of the customer care functionality. After the Conversation service identifies the user query is be an FAQ intent, the query is passed on to R&R to produce the JSON response.

Under services.js of the source code repository, a username and password is required. The information can be obtained under the service credentials, that is available when a new R&R instance is created. Also, under /lib/faq.js, a cluster ID and collection name is required to create a Solr Client. This information is available when a new collection is created in the R&R instance.

An export of the collection can be found under /watson-exports/retrieve-and-rank folder. The export consist of JSON files of the questions and documents used to populate the R&R collection. These JSON files can be imported into a new collection.

5 IBM Watson Document Conversion

Watson Document Conversion is a service that is used to consolidate .docx files that contains the knowledge base of the Retrieve & Rank (R&R) service. This service is accessible from the R&R platform under IBM BlueMix and does not require any additional configurations.

6 IBM Watson Language Translator

Watson Language Translator identifies the languages used in the user query. The chatbot service provides link to the language supported website, where available (German, French, Mandarin), but otherwise process the query if it is in English.

Under services.js of the source code repository, a username and password is required. The information can be obtained under the service credentials, that is available when a new instance of language translator is created.

7 Amazon Web Services (AWS) Lambda

Lambda is used to wrap the chatbot service into a single function. There is no need to manage any servers and payment is only due for the compute time used. Lambda automatically handles running of the code and scaling it with high availability.

A Lambda can be created on the AWS platform. Select Node.js 6.10 as the Runtime and upload the zip file of the source code as the Lambda function code. The ZIP file can be obtained by calling `npm run zip` on the terminal at the root of the source code repository.

8 Amazon Web Services (AWS) API Gateway

AWS API Gateway is used to host the Lambda. It involves creating a POST method which passes on the query from the client to the Lambda and returns the JSON response to the client.

The screenshot displays the AWS API Gateway console interface for configuring a POST method. The breadcrumb trail shows: Amazon API Gateway > APIs > conversationApi (qsrniqt7zk) > Resources > /api/message (gdr0pm) > POST. The main content area shows the 'Method Execution' configuration for the selected resource. It includes a 'Resources' sidebar with navigation options like Stages, Authorizers, Models, and Documentation. The central panel shows the 'Method Request' and 'Integration Request' boxes, and the 'Method Response' and 'Integration Response' boxes. The 'Method Request' box shows 'Auth: NONE' and 'ARN: arn:aws:execute-api:eu-west-2:416411756259:qsrniqt7zk/*/*POST'. The 'Integration Request' box shows 'Type: LAMBDA' and 'Region: eu-west-2'. The 'Method Response' box shows 'HTTP Status: 200' and 'Models: application/json => Empty'. The 'Integration Response' box shows 'HTTP status pattern: - - -' and 'Output passthrough: Yes'. The interface also includes a 'Client' label on the left and a 'Lambda processUserQuery' label on the right.

9 Front-End Application

To demo the chatbot service, a web chat application is developed, in HTML, CSS, JavaScript, AJAX. Currently, it is only hosted locally to test the service. To continue using the web app, the REST API end-point URL has to be changed in the *makeRequest()* method, under */front-end/assets/common/js/chatbot.js*.