

Memoria chatbot con Watson

Antecedentes

Mi compañero de asignatura Sergio Semedi y yo, entendimos erróneamente que tanto la presentación del último día como la práctica del chatbot eran por parejas.

Durante el desarrollo de nuestra primera versión (conjunta) del mismo, un recomendador de restaurantes, se nos ocurrió la posibilidad de utilizar una fuente de conocimiento externa a Watson en la que poder hacer consultas utilizando los parámetros (entidades, intenciones, variables etc) identificados por el bot. De otro modo, tendríamos que gobernar el hilo de la conversación desde el bot, o tener un número de ramas que no consideramos escalable.

Se da además la circunstancia de que estoy haciendo prácticas en IBM por lo que no me ha sido difícil informarme de como integrar un bot en una infraestructura, la solución que se me propuso, consiste en implementar un orquestador, que básicamente es un cliente del bot, que a través de una API que ofrece IBM Watson para cada bot, identificándolo con el *workspaceId*, *clientId* y *password*, gestiona la comunicación entre el usuario y el bot, recibiendo de este además toda la información asociada al procesamiento del lenguaje (parámetros del párrafo anterior) reconocida por el mismo.

En nuestro caso leeremos dichas credenciales de un fichero en formato *yaml* cifrado con *pgp*.

De esta forma, ese mismo orquestador puede, cuando ya haya identificado las intenciones y preferencias del usuario (en el caso de los restaurantes precio, tipo de comida, localización etc del restaurante a recomendar) utilizarlas para consultar en cualquier fuente de conocimiento, nosotros utilizamos una base de datos en *Mongo*, para dar la información final, (por ejemplo mostrar al usuario un restaurante que se ajusta a las características que este ha manifestado).

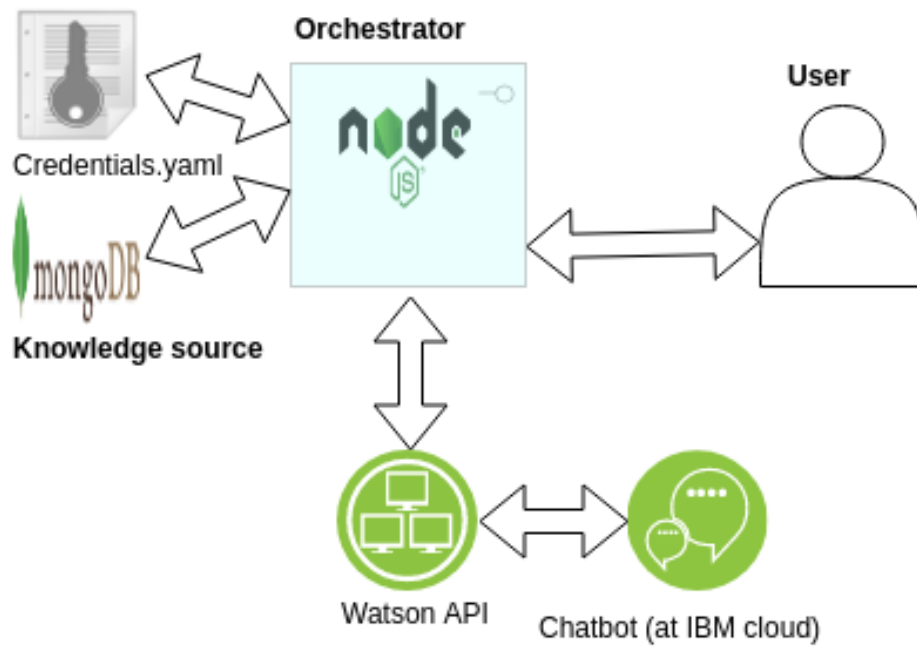


Figure 1: topologia

Esta aproximación es independiente al chatbot y fuente de conocimiento que se utilicen, y aunque finalmente vamos a entregar una práctica cada uno, como la investigamos juntos y la vamos a utilizar en ambas prácticas, consideramos más honesto entregar esta parte de la memoria común.

Datapath

Como se menciona anteriormente la comunicación entre el usuario y el chatbot se realiza a través de el orquestador desarrollado en javascript. La API a la que nos referimos en todo este documento es de tipo REST y la información se intercambia en formato *JSON*.

1. Al comenzar la ejecución del programa se leen las credenciales (identificadores de *workspace* y usuario y contraseña) del fichero *yaml* y se envían a la API de Watson para iniciar la conversación con nuestro bot.
2. Si, en el diálogo diseñado para el bot, este inicia la conversación con el usuario, el orquestador recibe el texto de ese diálogo, se lo muestra al usuario y permanece a la espera de que este último introduzca alguna respuesta. También se recibe un contexto inicialmente vacío que se incluirá en cada respuesta y se explica en el punto 4.
3. Cada respuesta que el usuario introduce en la interfaz del orquestador, este se lo retransmite al bot a través de la API, enviando además del texto del mensaje el contexto que mencionamos en el punto anterior y vamos a replicar al bot en cada petición que hagamos.
4. Cada vez que el bot responde al usuario, el orquestador recibe de este un objeto con el texto de la respuesta, y un campo con la información de la entidades, intenciones y variables que este ha detectado en el diálogo a este momento. A parte de replicarle estas variables al bot (para que sepa en que rama y nodo estamos) las procesaremos en la lógica del orquestador para saber qué tenemos que consultar en la base de conocimiento para mostrar al usuario y cuando hacer dicha consulta.
5. Los puntos 3 y 4 se repiten hasta que la aplicación ha recibido suficiente información para mostrarle al usuario el resultado de alguna consulta en su fuente de conocimiento.
6. Una vez realizada la consulta se ofrece su resultado al usuario y se procede a finalizar la conversación.

1 y 2

Identificación de workspace y usuario y autenticación, y retransmisión del texto del primer mensaje al usuario.

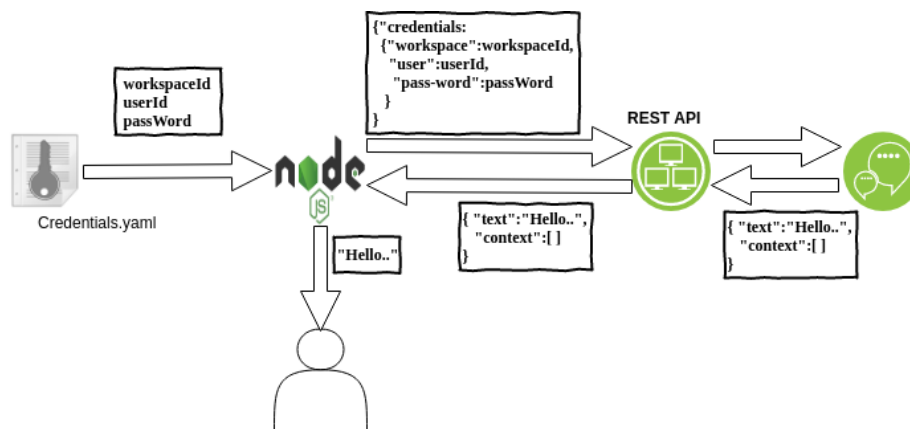


Figure 2: datapathLogin

3 y 4

Flujo de una conversación a través del orquestador.

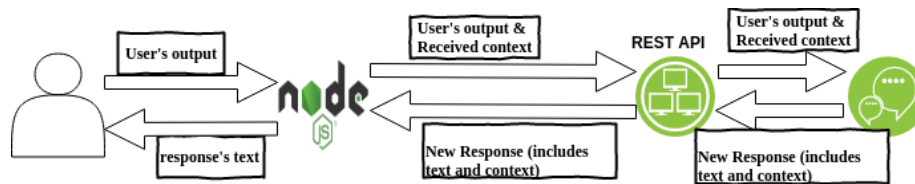


Figure 3: datapathConversation

6

Identificación de los objetivos del usuario y consulta en base de conocimiento.

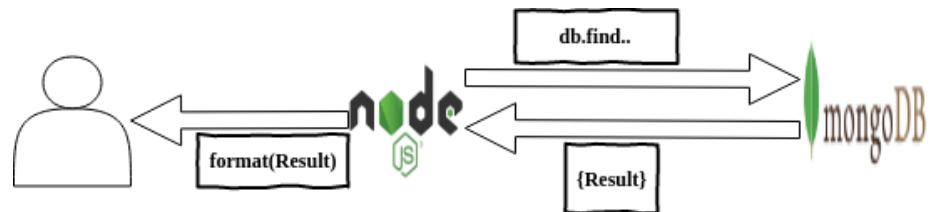


Figure 4: datapathQuery