

#PartyCloud

IBM

Codice, Dati e Tecnologie Esponenziali
per accelerare il cambiamento.

Hands-on

CiBot: Un consulente virtuale che aiuta a scegliere una dieta sana ed equilibrata

Francesca Gigante: francesca_gigante@it.ibm.com

Fernanda Perego: fperego@it.ibm.com

Luca Crippa: luca_crippa@it.ibm.com



Indice

Introduzione	3
Obiettivi	3
Guida all'uso del presente documento.....	4
Prerequisiti	4
Creazione account IBM Cloud.....	5
STEP 1: Creazione del servizio Watson Assistant in IBM Cloud.....	6
STEP 2: Creazione / import di un nuovo workspace.....	9
Import del workspace	10
STEP 3: Creazione degli intenti	13
STEP 4: Creazione delle entità.....	15
STEP 5: Creazione del dialogo	18
Inizializzazione del dialogo.....	18
Costruzione della conversazione	18
STEP 6: Test della conversazione con POSTMAN	31
STEP 7: Installazione di una applicazione web Node.js.....	37
Test della applicazione in locale.....	38
Installazione della applicazione su IBM Cloud	38
Modifica della applicazione	39
Riferimenti	40
Acknowledgements and Disclaimers	41



Introduzione

I [servizi Watson](#) presenti sulla piattaforma IBM Cloud consentono di istruire ed integrare funzionalità di AI con qualsiasi applicazione.

I servizi Watson sono disponibili sotto forma di API REST per interagire con ogni tipo di software. Per i più comuni linguaggi di programmazione è disponibile il relativo SDK per una integrazione più semplice e veloce.

Il servizio Watson Assistant (precedentemente noto con il nome di Conversation), combina molteplici tecnologie cognitive per permettere la creazione di un *agente conversazionale* (comunemente conosciuto con il termine di *bot* o *chatbot*) definendo intenti ed entità e creando un flusso di dialogo che simula una conversazione.

Il sistema può essere successivamente rifinito con tecnologie supplementari per avvicinarlo sempre di più al comportamento umano e fornendo una più elevata variabilità di risposte legate anche al contesto sempre diverso della conversazione.

Il servizio Watson Assistant permette di creare una infinita varietà di bot, dai più semplici e focalizzati ai più sofisticati ed erogarli su qualsiasi canale (web, mobile app, telegram, messenger, slack, persino su robot fisici, ecc.).

Alcuni classici esempi di Watson Assistant sono:

- aggiungere un chatbot ad un sito web che risponda alle domande dei clienti
- costruire una app di messaggistica che interagisce istantaneamente con gli utenti
- ecc.

Obiettivi

Per questo semplice laboratorio ci siamo posti come obiettivo di creare un servizio di Watson Assistant ed utilizzare lo strumento per l'istruzione della chatbot.

Abbiamo quindi pensato a 'CiBot', un consulente che risponde alle più comuni domande di chi cerca consigli utili per la propria alimentazione.

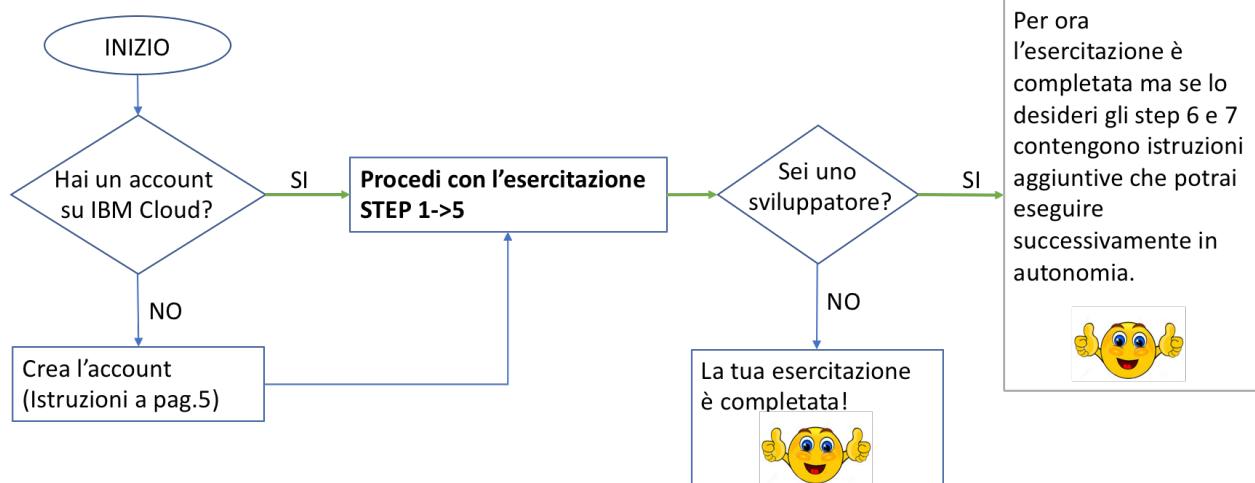
Nel corso del tempo, infatti, sempre più persone, anche quando non hanno evidenti problemi di mantenimento del peso, cercano di scegliere una dieta sana ed equilibrata. L'obiettivo principale che ci porremo è quello di evitare le comuni patologie legate all'alimentazione e garantirsi la minimizzazione delle penalizzazioni delle prestazioni fisiche legate all'invecchiamento.

Durante gli step 1 e 2 procederemo con la creazione del servizio e l'import della chatbot già parzialmente preparata; successivamente procederemo con l'istruzione di nuovi componenti

Il documento copre anche step opzionali (6 e 7) per utilizzare le API del Watson Assistant in una chiamata REST o applicazione Node.js



Guida all'uso del presente documento



Prerequisiti

Prima di partire con l'esercitazione è necessario completare i seguenti passi iniziali:

- Creare un account IBM Cloud – procedura descritta alla pagina seguente
- Scaricare in locale il repository github del lab contenente il file json che useremo durante l'esercitazione
<https://github.com/francesca-gigante/cibot-hands-on>
- (Opzionale per l'esecuzione del test in locale allo step 7) Installare sulla propria stazione di lavoro il runtime Node.js (<https://nodejs.org>) ed il pacchetto Express
npm install -g express-generator
- (Opzionale per l'installazione della applicazione su IBM Cloud allo step 7)
Installare sulla propria stazione di lavoro lo strumento di controllo del Cloud IBM da riga di comando.
Il tool si può scaricare accedendo a questo link
https://console.bluemix.net/docs/cli/reference/bluemix_cli/download_cli.htm
Una volta installato si può testare verificando che risponda a questo comando:
ibmcloud help

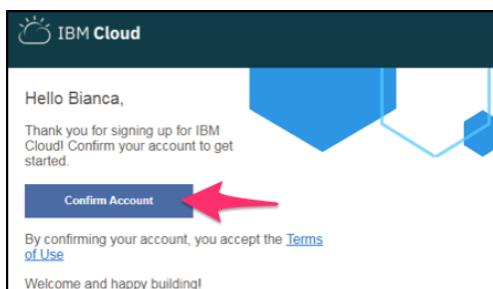


Creazione account IBM Cloud

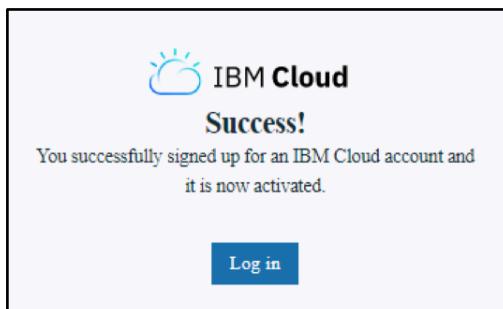
- Accedi al seguente link <https://console.bluemix.net/registration/> e procedi con la registrazione inserendo i tuoi dati; poi clicca sul pulsante ‘Crea Account’

The screenshot shows the IBM Cloud registration page. The left side has promotional text: "Sign up for an IBMid and create your IBM Cloud account", "Build on IBM Cloud for free with no time restrictions", "Guaranteed free development with Lite plans", "Start on your projects right away", "Get on us to try paid services", and "Ready to get started? Sign up today!". The right side is a form with fields: Email* (with value "biancamybooks"), First Name* (with value "Bianca"), Last Name* (with value "Romano"), Company (empty), Country or Region* (with value "Italy"), and Password* (with value "*****"). Below the form are two checkboxes: "Please keep me informed by email of products, services and offerings from IBM companies worldwide." and "By clicking Create Account, I accept the [IBM Cloud privacy policy](#) and [IBM Cloud terms](#)". At the bottom is a blue "Create Account" button with a pink arrow pointing to it.

- Apri la tua casella di posta e completa la registrazione aprendo la mail ricevuta dal Bluemix Team e cliccando il pulsante “Conferma Account”



- Complimenti! Il tuo account è stato creato!





STEP 1: Creazione del servizio Watson Assistant in IBM Cloud

- Apri il web browser al seguente indirizzo <http://console.ng.bluemix.net> ed accedi usando le credenziali inserite al momento della creazione dell'account effettuata al passo precedente
- L'account si apre visualizzando una dashboard (inizialmente vuota) a cui dovrà aggiungere il servizio. Scorri la pagina fino alla sezione "Tutti i servizi" e clicca sul pulsante "**Crea Risorsa**". In alternativa puoi selezionare il Catalogo.

The screenshot shows the IBM Cloud Dashboard. At the top, there are navigation links: Catalog, Docs, Support, and Manage. Below the header, the word "Dashboard" is displayed. On the left side, there are filters for RESOURCE GROUP (All Resources), REGION (United Kingdom), and CLOUD FOUNDRY ORG (iailarom@yahoo.it). A blue rectangular box highlights the "View Filters" button at the bottom of the sidebar. In the center, a large blue button labeled "Create resource" is shown, with a red arrow pointing directly at it.

- Dal menu di sinistra, in fondo, selezionare la voce **AI** per visualizzare solo i servizi cognitivi:

The screenshot shows the AI service catalog. On the left, a sidebar lists categories: All Categories, Compute, Containers, Networking, Storage, AI (selected and highlighted with a blue box), Analytics, Databases, Developer Tools, Integration, Internet of Things, Security and Identity, Starter Kits, Web and Mobile, and Web and Application. The main area displays various AI services in a grid. One service, "Watson Assistant (formerly Conversation)", is highlighted with a red arrow. The service details are as follows:
Watson Assistant (formerly Conversation)
Lite • IBM
Add a natural language interface to your application to automate interactions with your end users. Common applications...

- Clicca sul servizio **Watson Assistant**

The screenshot shows a detailed view of the Watson Assistant service card. It includes:
Watson Assistant (formerly Conversation)
Lite • IBM
Add a natural language interface to your application to automate interactions with your end users. Common applications...



- Si apre una schermata come la seguente:

The screenshot shows the IBM Cloud service catalog page for 'Watson Assistant (formerly Conversation)'. At the top, there's a brief description of the service: 'Add a natural language interface to your application to automate interactions with your end users. Common applications include virtual agents and chat bots that can integrate and communicate on any channel or device. Train Watson Assistant service through an easy-to-use web application, designed so you can quickly build natural conversation flows between your apps and users, and deploy scalable, cost effective solutions.' Below this, there are fields for 'Service name' (set to 'XXXXXXX'), 'Choose a region/location to deploy in' (set to 'United Kingdom'), 'Choose an organization' (set to 'francesca_gigante@it.ibm.com'), and 'Choose a space' (set to 'dev_uk'). There are also tabs for 'Lite' and 'IBM' plans, and links for 'View Docs' and 'Terms'. Below these, there are fields for 'AUTHOR' (set to 'IBM'), 'PUBLISHED' (set to '05/22/2018'), and 'TYPE' (set to 'Service'). At the bottom, there are links for 'Need Help?' and 'Contact IBM Cloud Sales', a 'Cost Calculator' link, and a prominent blue 'Create' button.

- Dai uno sguardo ai dettagli del servizio. In alto a sinistra è presente la descrizione del servizio e il puntamento ai documenti di dettaglio. Scorrendo la finestra in basso puoi rivedere i piani di pagamento. Il piano chiamato ‘Lite’ permette di testare e sviluppare le soluzioni con le API Watson senza che venga applicato alcun costo al servizio. Buon divertimento!
- Nella parte centrale della schermata puoi inserire le informazioni relative al tuo servizio. Riungi i campi come segue:
 - **Nome del servizio:** my_assistant (o altro nome a piacere)
 - Lascia i valori di default proposti negli altri campi (regione, organizzazione, spazio)
 - **Piano:** LiteIl piano Lite è gratuito con le seguenti limitazioni:
 - 10,000 API Calls al mese
 - 5 Workspaces
 - 100 Intenti
 - 25 Entità

Ti posso garantire che per sviluppare sono oltremodo abbondanti! 😊

- Clicca il pulsante ‘**Crea**’ in fondo a destra.
- IBM Cloud ha ora creato una nuova istanza del servizio associata al tuo account.
- Viene quindi mostrata una schermata ove è possibile accedere:
 - 1) allo strumento di creazione delle chatbot
 - 2) alle credenziali del servizio per l’uso tramite applicazione
 - 3) alla documentazione
 - 4) alle informazioni del piano di tariffazione con possibilità di cambiarlo
 - 5) alle informazioni sul consumo mensile (numero di chiamate)



The screenshot shows the IBM Cloud Watson service dashboard. On the left sidebar, there are four items: Manage (selected), Service credentials (with a red '2' badge), Plan (with a red '4' badge), and Connections. The main area displays the service details for 'my_conversation': Location: United Kingdom, Org: kps@de.ibm.com, Space: dev. It shows '1.06% Used | 9894 Api calls available'. A red arrow points to the 'Show' button next to the 'Configure credentials' link under the 'Credentials' section. The credentials JSON is displayed as follows:

```
{  
  "url": "https://gateway.watsonplatform.net/assistant/api",  
  "username": "*****",  
  "password": "*****"  
}
```

- Nella sezione ‘Credenziali’ puoi cliccare il pulsante ‘Show’ per vedere il tuo username e password che serviranno più in là negli step 7 e 8 per creare il collegamento al servizio dalla tua applicazione. Puoi copiare queste credenziali in un file di testo oppure più semplicemente rientrare in questa sezione quando ti verranno richieste le credenziali per configurare la tua applicazione.
- Ricordati che se accederai nuovamente ad IBM Cloud il servizio sarà già attivo e presente nella dashboard. Per accedere alla dashboard basta cliccare sulla voce Menu (Hamburger button) in alto a sinistra

The screenshot shows the IBM Cloud dashboard. The top navigation bar includes Catalog, Docs, Support, and Manage. Below the navigation, there's a search bar labeled 'Filter by resource name...' and a 'Create resource' button. The main area is titled 'Dashboard' and shows filter options: RESOURCE GROUP (All Resources), CLOUD FOUNDRY ORG (All Organizations), CLOUD FOUNDRY SPACE (All Spaces), LOCATION (All Locations), and CATEGORY (Watson). A red arrow points to the 'Watson' category filter.

- Successivamente clicca sul nome del servizio per accedere al tool

The screenshot shows the IBM Cloud dashboard with the 'Watson' category selected. The main area is titled 'Cloud Foundry Services' and lists two instances: 'Conversation-FG' and 'Conversation-FG'. A red arrow points to the first 'Conversation-FG' entry. The table columns are: Name, Region, CF Org, CF Space, Plan, and Service Offering.

Name	Region	CF Org	CF Space	Plan	Service Offering
Conversation-FG	United Kingdom	francesca_gigante@	dev_uk	Lite	Watson Assistant (fi)
Conversation-FG					



STEP 2: Creazione / import di un nuovo workspace

Il servizio di Watson Assistant appena creato ti permette di ‘istruire’ più chatbot/conversazioni. Ciascuna di esse sarà contenuta in un contenitore chiamato **workspace** che rappresenta tutto l’ambiente di lavoro relativo alla singola chatbot.

Ricorda che le modifiche che farai di volta in volta sulla tua chatbot sovrascriveranno il tuo workspace senza tenere traccia delle fasi intermedie di lavoro. Quindi, quando ti sembra di essere arrivato a buon punto, e vuoi provare ulteriori modifiche/cancellazioni senza sapere se vorrai mantenerle o meno, è consigliabile scaricare il workspace attuale in un file json da utilizzare come backup o come modalità di partenza per un nuovo progetto.

Allo stesso modo se qualcuno realizza una chatbot da un account e la vuole passare ad un altro account, basta scaricare il json e ricaricarlo sull’account di destinazione.

Niente di più facile!

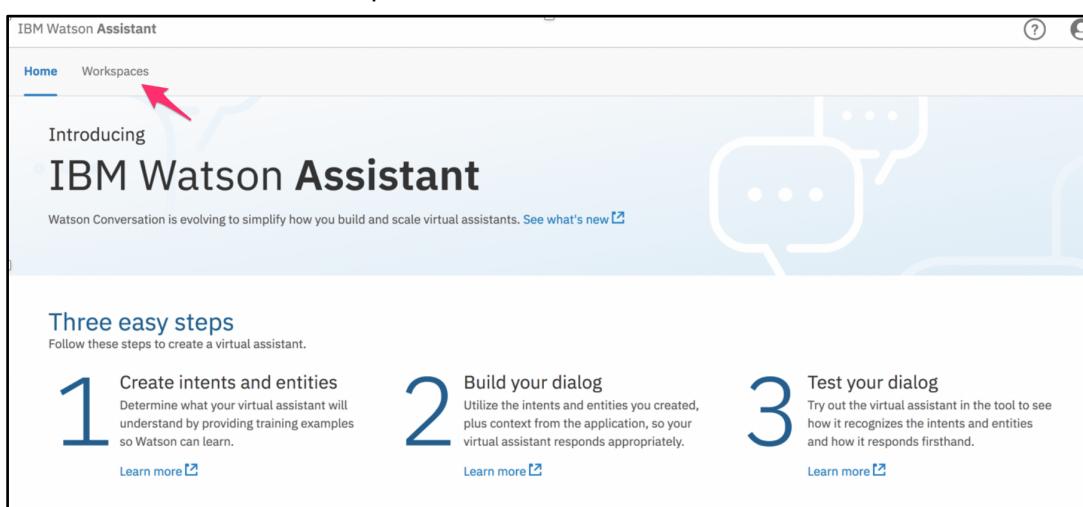
Ma come si fa?

Per iniziare bisogna entrare nello strumento di ‘istruzione’ delle conversazioni.

- Clicca sul pulsante ‘Launch tool’ (o ‘avvia strumento’ a seconda che il tuo browser sia in inglese o in italiano)



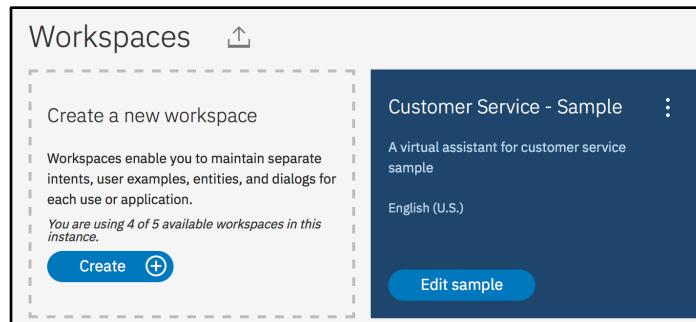
- Si apre questa schermata ove è possibile seguire un tutorial che passo passo ti guida alla creazione del tuo primo assistente virtuale.



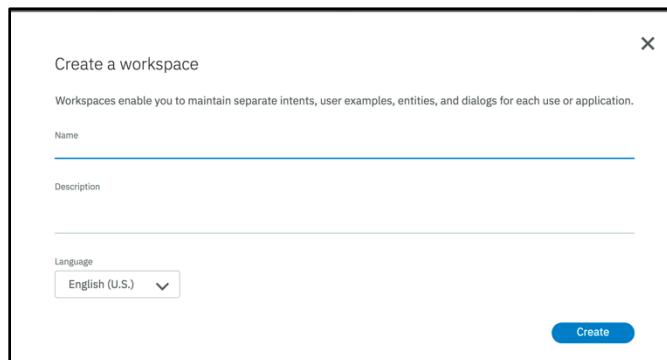
- Cliccando su ‘Workspaces’ in alto a destra, si entra nel vero e proprio strumento. Da qui si può partire in due modi diversi:
 - Creare un nuovo workspace cliccando il pulsante ‘Create’
 - Oppure importarne uno già esistente a partire da un file json cliccando sul pulsante ‘Import’ 



E' inoltre già presente un esempio da cui si può partire per iniziare a familiarizzare con lo strumento.



- Come già anticipato, questo laboratorio si svolgerà importando un assistente già creato precedentemente ed aggiungendo alcuni elementi della sua istruzione per migliorarlo.
- Nel caso invece avessimo voluto creare un nuovo workspace bisognava cliccare sul pulsante ‘**Crea**’ ed inserire le seguenti informazioni:
 - **Nome** : Nome che volete dare alla chatbot
 - **Descrizione**: descrizione a piacere
 - **Lingua**: Italiano (o altra lingua tra quelle disponibili)



- cliccando nuovamente sul pulsante ‘**Crea**’ il workspace viene creato ‘vuoto’ e si accede al tool di ‘istruzione’.

Import del workspace

Come già introdotto negli obiettivi del corso, per questo laboratorio abbiamo pensato ad un assistente che aiuta nella scelta del menu ideale.

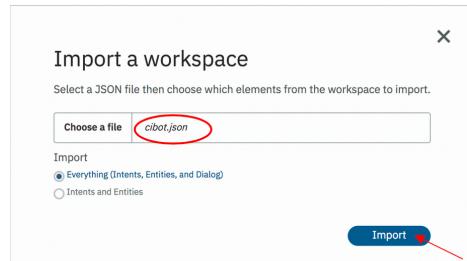
Vedremo quali sono tutti gli strumenti che consentono di raffinarlo e specializzarlo facilmente e in completa autonomia.

E' importantissimo sottolineare come sia fondamentale, per quello che diremo da qui in avanti, che l'esempio rappresenta solo una piccola parte delle funzioni che sono state rese disponibili attraverso lo strumento. Il consiglio è quello di approfondire sempre la documentazione di dettaglio accessibile in contesto.

- A partire dalla schermata principale del tool clicca su ‘Import’



- Seleziona il file '**cibot_iniziale.json**' scaricato in locale dal repository github (<https://github.com/francesca-gigante/cibot-hands-on>):

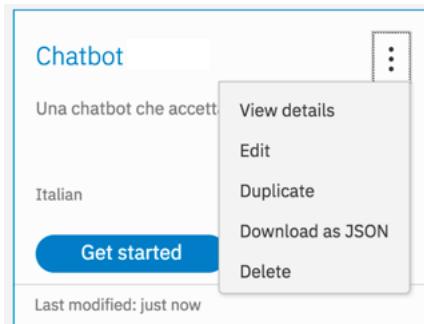


- Troverai il nuovo workspace chiamato Cibot, come da immagine sotto riportata:

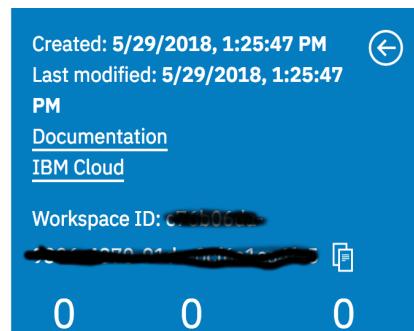


- Da qui, cliccando su è possibile:

- accedere alle informazioni di dettaglio del workspace selezionato (come da immagine successiva -> ‘View Details’),
 - modificare nome o descrizione (Edit)
 - duplicarlo
 - cancellarlo
 - esportare il Json corrispondente.



- Una informazione di dettaglio fondamentale per ogni workspace è il ‘workspace id’ che si trova nella schermata di dettaglio e costituisce, assieme alle credenziali al servizio, uno dei tre parametri con cui vengono richiamate le API



- Clicca ora nel riquadro della chatbot ‘CiBot’ per entrare al suo interno ed iniziare a modificarla.
- Ti troverai in una schermata come questa:

- E’ interessante notare i 4 menù principali a sinistra:
 - sezione di ‘istruzione’ della chatbot
 - sezione di gestione delle credenziali della chatbot e dove è possibile accedere ad una serie di esempi di modalità di installazione
 - sezione di ‘monitoraggio’ e ‘miglioramento’ della chatbot
 - pulsante di ritorno alla lista dei workspaces (E’ possibile tornare alla lista dei workspaces anche cliccando sul link ‘Workspaces’ in alto a sinistra)



STEP 3: Creazione degli intenti

Prima di usare una nuova conversazione è necessario istruirla con intenti, entità e nodi di dialogo che realizzano un caso d'uso prefissato

Un intento è il proposito o obiettivo che l'utente esprime attraverso i suoi input/domande.

Esempio: "Quante proteine al giorno dovrei assumere?" -> Intento: *fabbisogno proteico*

E' necessario innanzitutto definire quali sono gli intenti che potranno aiutare il controllo del flusso del dialogo. Il Watson userà le sue capacità di comprensione del linguaggio naturale per riconoscere l'intento legato alla richiesta / affermazione dell'utente e di conseguenza potrà scegliere quale nodo del dialogo deve essere attivato per gestire la conversazione.

In questa sezione vedremo passo passo come si crea un intento.

- Se non sei già nella sezione degli intenti clicca sul link '**Intenti**' in alto a sinistra

The screenshot shows a navigation bar with the following items: Workspaces / CIBOT / Build. Below the bar, there are four tabs: Intents (which is highlighted in blue), Entities, Dialog, and Content Catalog. The 'Intents' tab has a blue underline.

- Il nostro assistente è stato già inizializzato con una serie di intenti che potrai visualizzare e modificare come meglio credi in autonomia dopo che avrai completato il laboratorio.
- Per continuare l'esercizio clicca sul bottone **Add intent** ed inserisci i seguenti valori:
 - Nome intento:** ciao
 - Descrizione intento:** saluti
- Clicca sul bottone **Create intent**
- A questo punto l'intento 'ciao' è stato creato. Ora bisogna indicare al Watson *alcuni* esempi di espressioni verbali che vengono utilizzate per i saluti. Con gli esempi forniti gli algoritmi di comprensione linguistica presenti all'interno del servizio si addestreranno per determinare quando un input generato da un utente ricade in questa categoria.
- Aggiungere i seguenti esempi: 'buondì', 'buonasera', 'buongiorno', 'ciao', 'salve'. Puoi anche aggiungerne altri che ti sembrano appropriati allo scopo. Vedrai alla fine tutta la lista come in figura seguente:



The screenshot shows the CiBot interface for creating a new intent. At the top, there's a form with fields for 'Intent name' (#ciao), 'Description' (saluti), and a section for 'Add user examples'. Below this, a button says 'Add example'. Underneath, a list titled 'User examples (5)' shows examples like 'buonasera', 'buondì', 'buongiorno', 'ciao', and 'salve', each preceded by a small checkbox.

- Dopo aver completato l'operazione cliccare il pulsante per tornare alla lista degli intenti
- Dovresti vedere il nuovo intento nella lista:

The screenshot shows the CiBot workspace interface with the 'Intents' tab selected. At the top, there are tabs for 'Workspaces' (Pizzeria / Build), 'Intents' (selected), 'Entities', 'Dialog', and 'Content Catalog'. A 'Try it' button is at the top right. Below, there's a search bar and a table with columns for Intent, Description, Modified, and Examples. The table shows one intent: '#ciao' with description 'saluti', modified '2 minutes ago', and examples '5'.

- La lista indica tutti gli intenti con relativa descrizione e il numero di esempi che sono stati utilizzati per definire l'intento stesso.
- Creare ancora un intento che aiuta l'utente a chiedere alla chatbot di spiegare di cosa si occupa

Nome intento: cosa_fai

Descrizione intento: descrizione a piacere

Esempi:

- Cosa sai fare?
- Come mi puoi aiutare?
- Di cosa ti occupi?
- Chi sei?
- Ecc... (se ti venissero in mente altre espressioni analoghe)

- A questo punto si può ora procedere con lo step successivo per creare le entità.
- **Attenzione!** Per permetterti di imparare velocemente tutte le funzionalità di base non abbiamo introdotto il 'Content Catalog' (si attiva come ultimo tab dopo 'Intenti' 'Entità e 'Dialogo') il quale è una sorta di libreria di intenti già predisposti (per ambito ecommerce, bancario, telecomunicazioni, generico, ecc.) che possono essere facilmente integrati nella propria conversazione semplicemente cliccando il pulsante 'Aggiungi al workspace'. Quando avrai del tempo per approfondire, non dimenticarti di provare gli intenti già pronti all'uso!



STEP 4: Creazione delle entità

Una entità è l'oggetto della richiesta dell'utente e quindi rappresentano il contenuto informativo associato al singolo intento. Le entità serviranno a modificare la risposta/comportamento dell'agente conversazionale.

Per esempio, se il nostro assistente ci chiede informazioni sul nostro profilo, tipo “Sei uomo o donna? Sei vegetariano o vegano? Sei intollerante o allergico a qualche alimento?” è necessario creare delle entità che possano associarsi a tali informazioni (sesso, intolleranze, regime alimentare)

- Se non sei già nella sezione delle entità clicca sul link ‘Entità’ in alto:

The screenshot shows the IBM Watson Assistant interface. At the top, there's a navigation bar with icons for Home, Workspaces, Intents, Entities (which is highlighted with a blue underline), Dialog, and Content Catalog. Below the navigation bar, there are two tabs: 'My entities' (underlined in blue) and 'System entities'. On the right side of the interface, there's a large text input field and a 'Try it' button.

- Potrai visualizzare le entità già presenti ed eventualmente modificarle come meglio credi in autonomia dopo che avrai completato il laboratorio
- A questo punto puoi cliccare il bottone ‘Aggiungi entità’ ed inserire la prima entità ‘sesso’ con le seguenti informazioni:

Nome entità: @sesso

Valori:

donna -> sinonimo: femmina, ‘una femmina’, ‘una donna’,.....

uomo -> sinonimo: maschio, ‘un maschio’, ‘un uomo’,

The screenshot shows the entity creation form for '@sesso'. The 'Entity name' field contains '@sesso'. The 'Value name' field has 'Enter value'. Under 'Synonyms', there is a dropdown menu set to 'Synonyms' and a button 'Add synonym...'. Below this, there is a 'Add value' button. The 'Entity values (2)' section lists two entries: 'donna' and 'uomo', each associated with a 'Type' of 'femmina' and 'maschio' respectively, via 'Synonyms' links.

- Ora clicca sul pulsante per tornare alla lista delle entità. Dovresti vedere la tua nuova entità nella lista.
- Ripeti l'operazione inserendo la seconda entità ‘**regime_alimentare**’ come segue:



The screenshot shows the CiBot Entities configuration interface. At the top, there's a field for 'Entity name' containing '@regime_alimentare'. To the right is a toggle switch for 'Fuzzy Matching'. Below this, there's a section for 'Value name' with a 'Enter value' input, a 'Synonyms' dropdown, and a button to 'Add synonym...'. A blue 'Add value' button is also present. The main area displays three entries under 'Entity values (3) ▾':

Value	Type	Synonyms
onnivoro	mangio tutto ne vegetariano ne vegano	mangio di tutto nessuno dei due né vegetariano e né vega...
vegano		Add synonym...
vegetariano		Add synonym...

- Come hai avuto modo di notare è facilissimo distinguere gli intenti dalle entità perché gli intenti sono preceduti dal simbolo ‘#’ mentre le entità sono precedute dal simbolo ‘@’
- Inoltre, per ogni entità posso esprimere diversi sinonimi (come nel nostro caso anche non esplicitamente sinonimi nel dizionario della lingua di riferimento ma rappresentativi del parlato comune)
- Un altro aspetto importante da considerare è la possibilità di attivare il pulsante di ‘Fuzzy Matching’ (in alto a destra), che permette al sistema di riconoscere le entità anche quando vengono digitate con dei piccoli errori (es una lettera mancante o errata oppure due lettere invertite). Nel nostro caso potrebbe essere utile attivarlo per tutte le entità (è tanto utile perché permette di usare indifferentemente i singolari / plurali / maschili e femminili in quanto l’alterazione di una lettera non impatta il riconoscimento del termine)
- Come ultima entità ci serve una entità numerica che l’utente potrà indicare come numero (es. 2) o come cifra (es. due). Le entità numeriche, assieme ad altre, sono già state previste dal servizio come entità di sistema; vanno solo attivate.
- Clicca quindi il link ‘Entità di sistema’

The screenshot shows the CiBot Entities tab. At the top, there are tabs for 'Intents', 'Entities' (which is active and highlighted in blue), 'Dialog', and 'Content Catalog'. Below this, there are two sections: 'My entities' and 'System entities'. An arrow points to the 'System entities' section. At the bottom, there are buttons for 'Add entity', 'Import', 'Export', and 'Delete'.

- Ora attiva l’entità ‘sys-number’. Successivamente, se vorrai arricchire ulteriormente il bot, potrebbe essere per esempio utile attivare il sys-date per gestire i giorni della dieta (es: “cosa devo mangiare domani?”).



Intents Entities Dialog Content Catalog

My entities System entities

These 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. *System entities cannot be edited. [Learn more](#)

Name (5) ▾	Description	Status
> @sys-currency	Extracts currency values from user examples including the amount and the unit. (20 cents)	<input type="checkbox"/> Off
> @sys-date	Extracts date mentions (Venerdì)	<input type="checkbox"/> Off
> @sys-number	Extracts numbers mentioned from user examples as digits or written as numbers. (21)	<input checked="" type="checkbox"/> On
> @sys-percentage	Extracts amounts from user examples including the number and the % sign. (15%)	<input type="checkbox"/> Off
> @sys-time	Extracts time mentions (alle 10)	<input type="checkbox"/> Off

- Ora che sono state completate entrambe le attività di creazione degli intenti e delle entità possiamo passare finalmente alla creazione del dialogo.



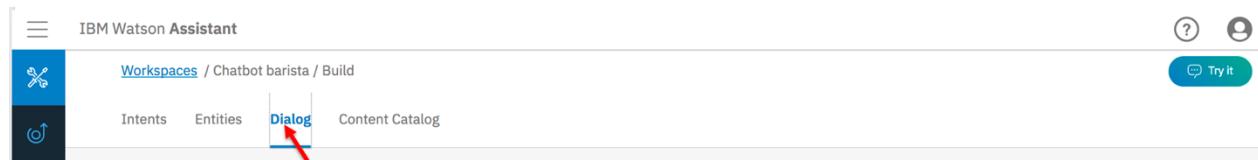
STEP 5: Creazione del dialogo

Il dialogo utilizza gli intenti e le entità che sono stati identificati, a cui si aggiunge il contesto della applicazione, per interagire con l'utente e fornire delle risposte.

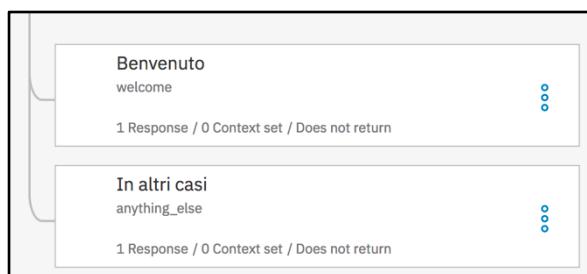
Ora che sei giunto in questa sezione, sarà necessario creare alcuni *dialoghi*. Un dialogo è un insieme di nodi *conversazionali* contenuti in un workspace. Insieme, ciascun insieme di nodi crea l'albero di dialogo generale. Ogni ramo è una parte diversa di la conversazione che può essere svolta con un utente.

Inizializzazione del dialogo

- Se non sei già nella sezione del dialogo clicca sul link ‘**Dialogo**’ in alto:



- Quando si crea una conversazione ex-novo in questa sezione si trova una schermata vuota in cui è presente un pulsante ‘**Crea**’ per creare una nuova conversazione. La nuova conversazione viene inizializzata con due nodi predefiniti chiamati ‘Benvenuto’ e ‘In altri casi’. Nel nostro caso, per velocizzare l'esercizio, questa operazione è stata già effettuata e sono già stati creati altri nodi di dialogo oltre quelli predefiniti.
- Descriviamo comunque i nodi iniziali:
Il nodo di benvenuto è il punto di partenza per la conversazione di questa applicazione. Cioè, se una API viene chiamata senza che sia stato definito un contesto legato alla conversazione, questo nodo verrà restituito in output.



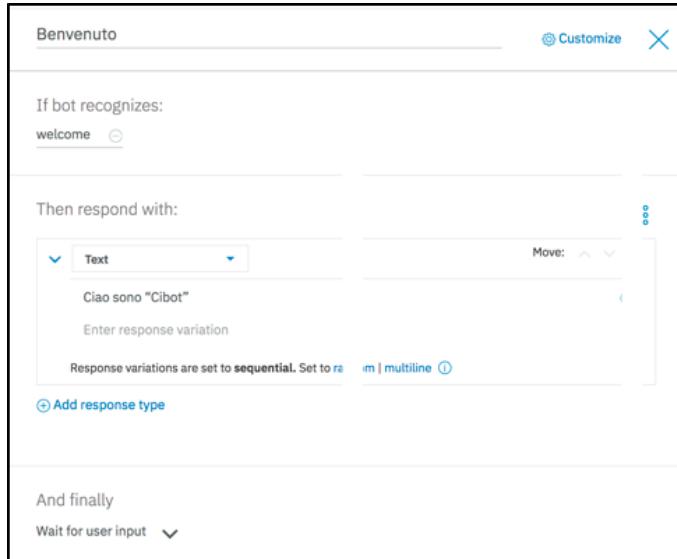
- Il nodo ‘In altri casi’ viene invece usato se l'utente non corrisponde a nessuno degli altri nodi definiti. Infatti, ad ogni interazione con gli utenti, il servizio cerca la risposta corretta controllando tutti i nodi dall'alto verso il basso... ricordati quindi di lasciare questo nodo sempre come ultimo!

Costruzione della conversazione

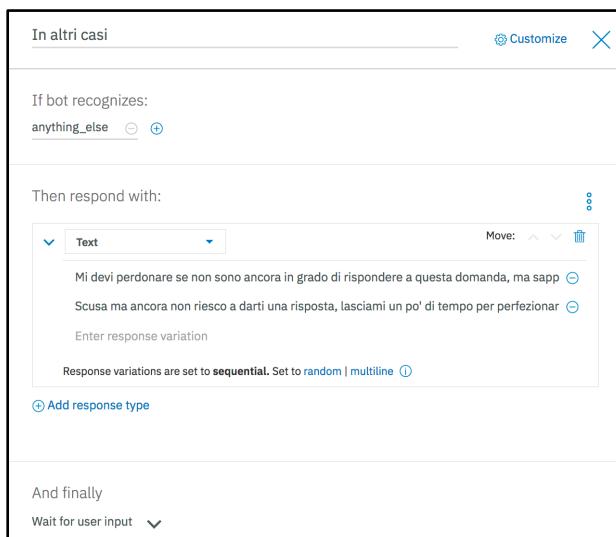
- Fai clic sul nodo di benvenuto; come puoi vedere ogni nodo è definito con un nome (in questo caso ‘Benvenuto’), con una condizione di attivazione (in questo caso ‘welcome’ che è una condizione predefinita sull’attivazione della chat) ed una serie di possibilità di definire le risposte. Ricordati di accedere alla documentazione per avere maggiori dettagli su tutte le funzionalità presenti sul nodo di dialogo (<https://console.bluemix.net/docs/services/conversation/dialog-overview.html>).



- Per ora limitati ad osservare la risposta di benvenuto “Ciao sono CiBot” e ad aggiungi alla condizione ‘welcome’ anche la condizione #ciao in alternativa (in questo modo il nodo si attiva quando parte la chatbot ma anche tutte le volte in cui l’utente saluta)



- Come puoi notare esiste in fondo una condizione di uscita ‘And finally’ che permette di definire il comportamento del nodo al termine della predisposizione della risposta. Per ora il nodo ritorna la risposta all’utente, successivamente vedremo come puoi modificarlo.
- Fai clic sul per chiudere la vista delle proprietà.
- Espandi il nodo ‘In altri casi’/‘Anything else’ e rivedi i suoi valori. Potrai modificarli successivamente come meglio preferisci, per ora sono state previste alcune risposte per il caso specifico come nella figura seguente:



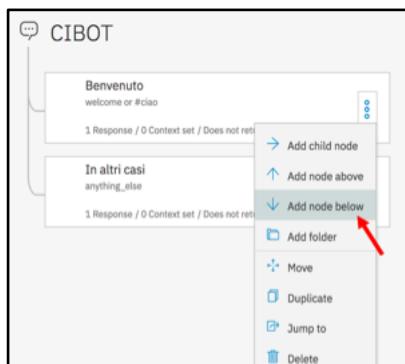
- Le risposte inserite in questo caso sono 2, puoi aggiungerne anche altre; le varie risposte verranno usate in maniera ‘sequenziale’, ma puoi anche impostarle in modalità ‘casuale’ (random). Questo principio è valido per tutte le risposte.



- Chiudi la vista.
- Ora continuerai a costruire il tuo bot utilizzando gli intenti, le entità e nodi di dialogo creati nei passaggi precedenti. Lo farai interamente con l'interfaccia web (non è richiesta alcuna programmazione o hacking di file txt/XML !)
- Dovresti creare un ramo di dialogo per ciascuno degli intenti identificati così come anche per definire bene l'inizio e la fine della conversazione. Determinare l'ordine più efficace con cui verificare le condizioni è un'abilità importante nella costruzione di una alberatura di dialogo. Qualora diventi troppo complesso trovare un ramo, ricordati che puoi raggruppare la conversazione in cartelle proprio come fai con i files del pc; inoltre controlla le condizioni per vedere se puoi semplificare il tuo dialogo riordinandole.

Nota: Spesso è meglio elaborare prima le condizioni più specifiche.

- Fai clic sul menu nel nodo di benvenuto, quindi fare clic su ‘Aggiungi nodo sotto’. In questo passaggio si sta creando un nuovo ramo nell'albero di dialogo che rappresenta una conversazione alternativa.



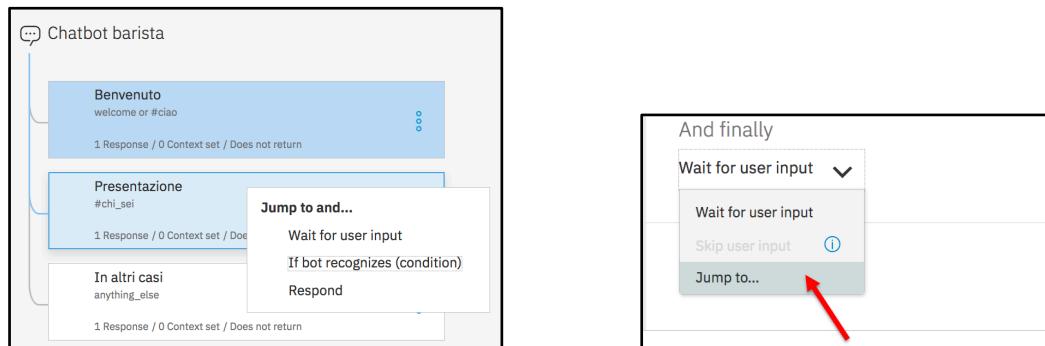
- In questo nuovo nodo, immettere i seguenti valori:
Nome nodo: Presentazione
Se il bot riconosce: #cosa_fai
Testo risposta: Sono il tuo nuovo amico nutrizionista. Ho bisogno di qualche informazione su di te per creare la tua dieta ideale. Sei uomo o donna? Quanti anni hai? Sei vegetariano o vegano? Sei intollerante o allergico a qualche alimento?



The screenshot shows the CiBot interface for creating a new node named 'Presentazione'. The configuration includes:

- If bot recognizes:** #cosa_fai
- Then respond with:** A 'Text' response: "Sono il tuo nuovo amico nutrizionista. Ho bisogno di qualche informazione su di te per cre..." followed by a placeholder "[Enter response variation]". A note indicates "Response variations are set to sequential. Set to random | multiline".
- And finally:** A 'Wait for user input' section.

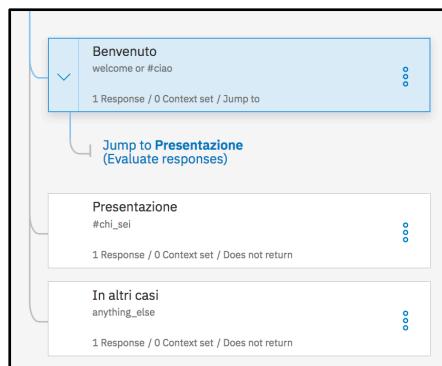
- Impostando la condizione su un intento (#cosa_fai), stai indicando che questo nodo verrà attivato da qualsiasi input che corrisponde all'intento specificato. Sarebbe però molto efficace poter inserire questa breve spiegazione delle *capacità* della chatbot anche dopo i saluti iniziali (nodo 'Benvenuto') oppure quando l'utente chiede qualcosa per cui non è stato istruito (nodo 'In altri casi'). Per fare questo possiamo evitare di riscrivere le stesse frasi negli altri nodi perché è possibile far *saltare* la chatbot da un nodo ad un altro per dare risposte multiple. Vediamo subito come.
 - Chiudi questa finestra di dialogo.
 - Riapri il nodo *Benvenuto* e vai nell'ultima sezione in fondo della finestra, dove c'e' scritto 'And finally... Wait for user inputs' (che significa che dopo che la chat ha dato la risposta ritorna all'utente per una nuova interazione).
 - Ora cambia 'Wait for user inputs' in 'Jump to...' e seleziona come nodo di destinazione il nodo 'Presentazione' appena creato



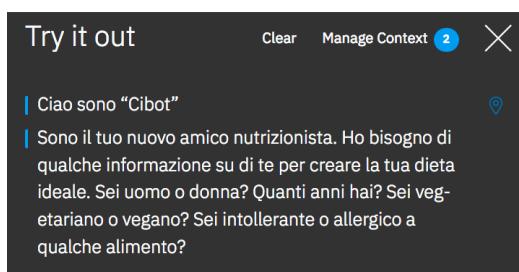
- Selezionando il nodo 'Presentazione' hai ancora a disposizione tre scelte, nel nostro caso useremo la terza 'Rispondi' senza aspettare l'utente o valutare altre condizioni.



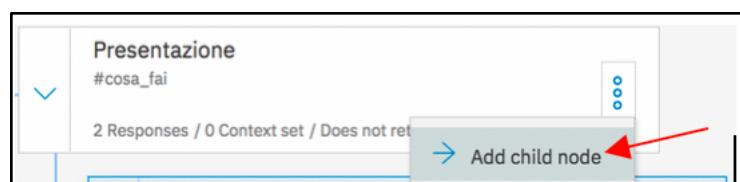
- La tua conversazione si presenta ora così:



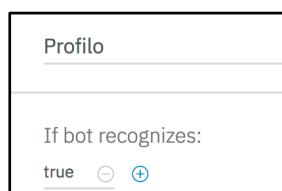
- Puoi subito provare questa nuova funzionalità semplicemente attivando l'interfaccia di test. Ti ricordi come si fa? Clicca su in alto a destra.



- Come puoi vedere ora il nostro assistente fornisce due risposte in sequenza: la prima è quella definita nel nodo *Benvenuto*, mentre la seconda è quella del nodo *Presentazione*.
- Se vuoi puoi provare in autonomia a fare la stessa cosa per il nodo *In altri casi* in modo che il nostro assistente dichiara di non aver compreso la domanda ma subito dopo ripete cosa sa fare.
- Ma ora passiamo al pezzo forte della conversazione. L'obiettivo è quello di creare un nodo in grado di comprendere tutte le informazioni richieste (sesso, età, regime alimentare e intolleranze) ed essere in grado di 'registrarle' nel profilo dell'utente per poterle poi usare successivamente per dare le risposte giuste quando l'utente pone domande specifiche che dipendono dal suo profilo.
- Crea un nuovo nodo 'figlio' sotto 'Presentazione'

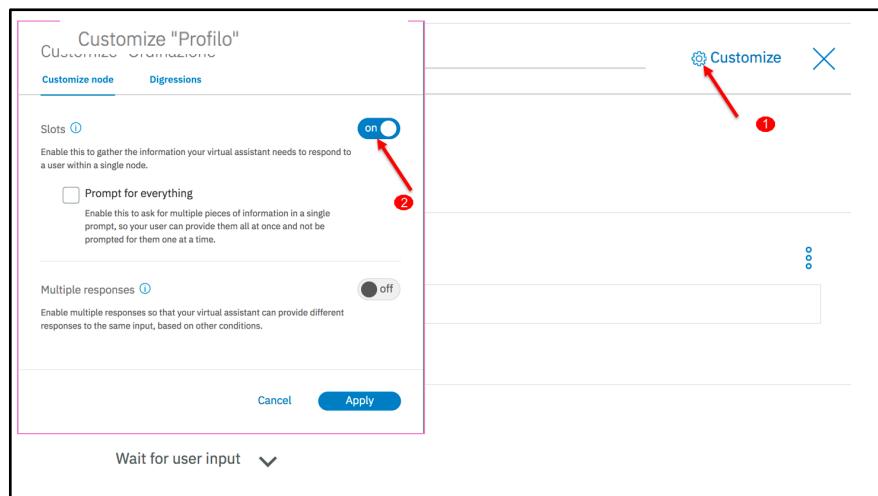


- Chiamalo Profilo, indica che si deve attivare **in ogni caso** (if bot recognize: true)





- Il nostro nutrizionista ora deve verificare che l'utente indichi correttamente tutte le informazioni del suo profilo. Per fare questo lavoro molto complesso ci viene aiuto una funzionalità preziosissima: gli *slots*. Come funzionano e come si attivano?
 - Gli slot si utilizzano per ottenere le informazioni necessarie prima di poter rispondere in modo accurato all'utente. Ad esempio, se gli utenti chiedono informazioni sulle proteine della carne, ma la risposta dipende dalla carne rossa o bianca e dal tipo di cottura, è possibile porre una domanda di follow-up sul tipo di carne e come viene cucinata prima di rispondere. È quindi possibile aggiungere condizioni di risposta che tengono conto delle informazioni sulla posizione fornite. Poiché in questo tutorial non vengono usate tutte le potenzialità degli slots ti invito a ad approfondire qui <https://console.bluemix.net/docs/services/conversation/dialog-slots.html#dialog-slots>
 - Gli slots si attivano personalizzando il nodo di dialogo (1) e attivando la funzione 'Slots' (2)



- Una volta attivati gli slots potrai configurare il nodo di dialogo in modo che gestisca da solo tutte le informazioni necessarie al completamento del profilo, così come indicato nella figura seguente



The screenshot shows the CiBot profile configuration interface. At the top, it says "Profilo" and has a "Customize" button and a close button. Below that, it says "If bot recognizes:" and shows a slot named "true". Then, it says "Then check for:" and lists four slots:

Check for	Save it as	If not present, ask	Type
1 @sesso	\$sesso	Prima di aiutarti ho bi	Required
2 @regime_alimentare	\$regime_alimentare	sei vegano, vegetaria	Required
3 @intolleranze	\$intolleranze	quali intolleranze hai'	Required
4 @sys-number	\$eta	quanti anni hai?	Required

Below this, there's a "+ Add slot" button. Then, it says "Then respond with:" and shows a "Text" response type. The text is: "Bene, da quello che posso aver capito dovresti essere \$sesso di \$eta anni, avere un regim". There's a "Move" button with arrows and a trash can icon. Below the text input, it says "Enter response variation" and "Response variations are set to sequential. Set to random | multiline". There's also a "+ Add response type" button. Finally, it says "And finally" and "Wait for user input".

- Per ogni informazione necessaria viene creato uno ‘slot’ contenente il riferimento all’entità corrispondente, il nome della variabile di contesto in cui registrare il valore indicato e la domanda da fare quando l’utente non indica l’informazione. In più si può definire una risposta che interviene quando nella frase dell’utente non è indicata nessuna delle informazioni e la risposta definitiva quando tutte le informazioni sono state date.
- Nel nostro caso abbiamo quindi:
 - Check for: @sesso
Save it as: \$sesso
If not present ask: “*Sei uomo o donna?*”
 - Check for: @regime_alimentare
Save it as: \$regime_alimentare
If not present ask: “*sei vegano, vegetariano o nessuno dei due?*”
 - Check for: @intolleranze
Save it as: \$intolleranze
If not present ask: “*quali intolleranze hai?*”



- Check for: `@sys_number`
Save it as: `$eta`
If not present ask: “*Quanti anni hai?*”
 - Then respond with: “*Bene, da quello che posso aver capito dovresti essere \$sesso di \$eta anni, avere un regime alimentare di tipo \$regime_alimentare e avere \$intolleranze. Corretto?*”
- A questo punto del tutorial sono state introdotte le variabili di contesto: `$sesso`, `$regime_alimentare`, `$intolleranze`, `$eta` ma cosa sono? Sono dei valori che vengono registrati nel contesto della conversazione e servono a conservare la ‘memoria’ della conversazione stessa. Per avere in ogni momento la situazione delle variabili di contesto puoi sempre cliccare sul pulsante: *Gestisci contesto* che si trova in alto a destra nella finestra di test della conversazione (il numeretto accanto indica in tempo reale quante sono le variabili di contesto).

Try it out Clear Manage Context 1 X

Quando vuoi ricominciare una conversazione, ricordati che il bottone ‘Clear’ ripulisce tutto, comprese le variabili di contesto, e riparte dal nodo di benvenuto.

- Lasciamo a te, poi in seguito, di predisporre il dialogo in modo da ripulire le variabili di contesto ad ogni nuovo ordine di bibita. Magari prova a far chiedere all’utente se desidera altro e in caso affermativo ripulisci tutte le variabili così che quando ripassi dall’ordine non vengono trovate variabili già impostate.
- Per accedere alle variabili di contesto puoi cliccare sul pulsante ‘Open context editor’ come indicato in figura



- Ma ritorniamo alla nostra conversazione.
- Ora dobbiamo creare due nodi figlio sotto il nodo Profilo, uno che risponde alla entità si e uno che risponde alla entità no, in modo da poter formulare una risposta diversa a seconda che le informazioni recepite dal nostro assistente siano corrette o meno. Prova a seguire le immagini riportate di seguito, dovresti essere in grado di creare gli stessi nodi anche tu.



Come si presentano i due nuovi nodi:



Nodo che risponde al no

If bot recognizes:
@si_no:no ⊖ ⊕

Then respond with:

Text
Mi dispiace se ho capito male, puoi riformulare meglio il tuo profilo? ⊖
Enter response variation
Response variations are set to **sequential**. Set to random | multiline ⓘ

⊕ Add response type

And finally
Jump to... ⊖ "Profilo" (User input) ⊖

Nodo che risponde al si

Name this node... _customize X

If bot recognizes:
@si_no:si ⊖ ⊕

Then respond with:

Text
Benissimo, fammi pure qualche domanda sulla dieta da seguire o sul fabbisogno calorico ⊖
Enter response variation
Response variations are set to **sequential**. Set to random | multiline ⓘ

⊕ Add response type

And finally
Wait for user input ⊖



- Ora possiamo sicuramente fare qualche prova.

Puoi provare ad indicare le informazioni tutte insieme, una alla volta o secondo delle combinazioni a caso. Qui di seguito sono riportati alcuni esempi. Prova anche a farne altri e poi fai qualche domanda sulle regole generali e sull'apporto proteico. La conversazione ha anche un nodo per il fabbisogno calorico, ma non è stato gestito....magari potrebbe essere un utile esercizio per affinare la tua tecnica 😊

Esempio 1: Regole alimentari per un vegetariano

Try it out Clear Manage Context 6 X

Ciao sono "Cibot"

Sono il tuo nuovo amico nutrizionista. Ho bisogno di qualche informazione su di te per creare la tua dieta ideale. Sei uomo o donna? Quanti anni hai? Sei vegetariano o vegano? Sei intollerante o allergico a qualche alimento?

sono un uomo di 35 anni vegano e non ho intolleranze

#chi_sono

@sys-number:1
@sesso:uomo
@sys-number:35
@regime_alimentare:vegano
@intolleranze:nessuna intolleranza

Bene, da quello che posso aver capito dovresti essere uomo di 35 anni, avere un regime alimentare di tipo vegano e avere nessuna intolleranza. Corretto?

si

Irrelevant

@si_no:si

Benissimo, fammi pure qualche domanda sulla dieta da seguire o sul fabbisogno calorico o proteico necessario per una dieta bilanciata.

cosa devo mangiare

#regole_generali

Assumere in modo variato cibi appartenenti a tutti i gruppi alimentari vegetali. Solo in caso di rilevata carenza, allora occorre porre in atto alcuni accorgimenti per aumentarne l'assimilazione, che sono: privilegiare l'assunzione di legumi, ricchi in proteine e ferro, abbinandoli nello stesso pasto a cibi ad elevato contenuto di vitamina C (come la frutta od il succo di limone) per esaltare l'assimilabilità del ferro. Evitare tutti i cibi che contengono calcio, tannini e fitatini, nello stesso pasto in cui si assumono cibi ricchi di ferro, perché riducono l'assorbimento del ferro vegetale.



Esempio 2: Regole alimentari per un non vegetariano

Try it out Clear Manage Context 6 X

Ciao sono "Cibot" ?

Sono il tuo nuovo amico nutrizionista. Ho bisogno di qualche informazione su di te per creare la tua dieta ideale. Sei uomo o donna? Quanti anni hai? Sei vegetariano o vegano? Sei intollerante o allergico a qualche alimento?

ciao sono donna, ho 22 anni, mangio di tutto e non sono intollerante

#ciao ▼

@sesso:donna
@sys-number:22
@regime_alimentare:onnivoro
@intolleranze:nessuna intolleranza

Bene, da quello che posso aver capito dovresti essere donna di 22 anni, avere un regime alimentare di tipo onnivoro e avere nessuna intolleranza. Corretto? ?

si

Irrelevant ▼

@si_no:si

Benissimo, fammi pure qualche domanda sulla dieta da seguire o sul fabbisogno calorico o proteico necessario per una dieta bilanciata ?

cosa devo mangiare?

#regole_generali ▼

Tieni conto delle seguenti regole generiche: non esageriamo con la carne (2-3 volte a settimana) , introduciamo il pesce (2-3 volte a settimana) e i legumi (2 volte a settimana) un po' più spesso e non limitiamoci sulle uova (1, ma anche 2 volte a settimana). I formaggi non sono da considerarsi delle aggiunte al pasto (del tipo "ho ancora fame, prendo un pezzo di formaggio" dopo aver mangiato un filetto di pesce, verdure e pane): piuttosto inseriamolo come vero secondo piatto 1-2 volte a settimana, idem i salumi e affettati.



Esempio 3 : fabbisogno proteico di un adulto

le proteine da assumere durante la giornata quante sono?

#fabbisogno_proteico

Il fabbisogno proteico dipende dal peso corporeo, dall'attività fisica e da altre variabili. Ad ogni livello di attività corrisponde un valore che va moltiplicato al peso per determinare quanti grammi di proteine giornaliere sono necessarie. Ecco di seguito alcuni esempi

- Se hai uno stile di vita sedentario o svolgi un'attività moderata, moltiplica il peso per 0,8. Quindi, se pesi 70 kg, avrai bisogno di 56 grammi di proteine.
- Se sei fisicamente attivo, moltiplica il peso per un coefficiente compreso tra 0,8 e 1,2.
- Se sei uno sportivo che deve costruire massa muscolare o sei un giovane atleta, hai bisogno di 1,2 – 1,8 grammi di proteine per ogni chilo di peso corporeo.

Ricorda inoltre che, in linea generale:

- ai bambini ne servono di più rispetto agli adulti
- la popolazione maschile ha un fabbisogno proteico maggiore di quella femminile
- le donne incinte o che stanno allattando devono assumere più proteine
- gli anziani potrebbero aver bisogno di un apporto maggiore per proteggersi dalla sarcopenia, arrivando a 1,2 g di proteine per chilogrammo di peso
- chi soffre di patologie epatiche o renali dovrebbe ridurre le proteine rispettando le indicazioni del medico.

Esempio 4: Fabbisogno proteico di un bambino.

quante sono le proteine necessarie ad un bambino?

#fabbisogno_proteico

@sys-number:1
@categorie:bambino

Il fabbisogno proteico dipende dal peso corporeo, dall'attività fisica e da altre variabili. In generale ad un bambino in età prescolare occorrono 1,2 grammi di proteine per ogni kg di peso mentre per i bambini in età scolare occorre circa 1 grammo per kg di peso

- Wow! Hai completato la parte più divertente del laboratorio!



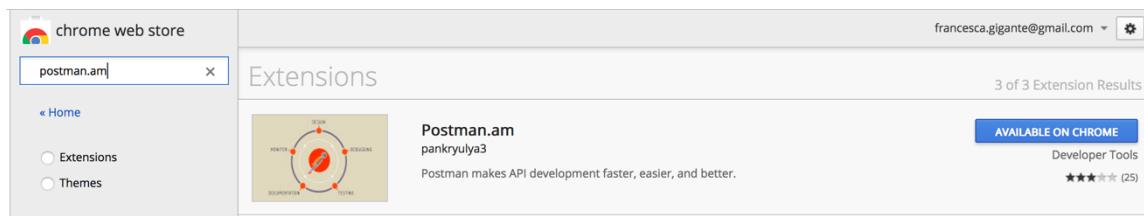
- Se avessimo avuto più tempo avremmo sicuramente inserito altri intenti... ma il tempo stringe e poi..... sicuramente ora sei in grado di continuare in autonomia non solo nel completare questo semplice esercizio ma anche inventare una chatbot tutta nuova!
- Mi raccomando, rivedi sempre la documentazione per la creazione di nodi di dialogo e per la definizione di condizioni più complesse di quanto è stato possibile realizzare con il tempo dell'hands-on.
- Ad esempio, è molto interessante poter manipolare i testi, le entità e le variabili di contesto. Prenditi tempo per rivedere qui cosa è possibile fare:
 - <https://console.bluemix.net/docs/services/conversation/dialog-methods.html#expression-language-methods>
 - <https://console.bluemix.net/docs/services/conversation/expression-language.html#expressions-for-accessing-objects>
- Un'altra funzione altrettanto importante è quella che permette l'accesso a funzionalità applicative direttamente dal Watson Assistant come descritto qui <https://console.bluemix.net/docs/services/conversation/dialog-actions.html#dialog-actions>
Pensa a quanto è utile questo approccio per rispondere alle domande che richiedono accesso a una base dati o a un servizio esterno, oppure se è necessario effettuare delle operazioni specifiche richieste dall'utente ed inviare conferma per messaggio.
- Ora, se vuoi procedere con le sezioni successive potrai vedere come si effettuano le chiamate API al servizio di Watson Assistant e come usarle all'interno del tuo programma.



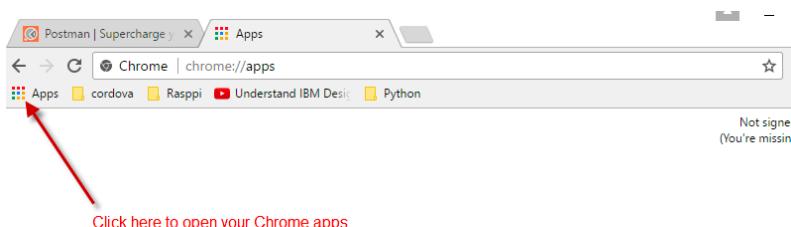
STEP 6: Test della conversazione con POSTMAN

I servizi presenti su IBM Cloud hanno tutti una interfaccia API REST (Representational State Transfer) e possono essere quindi testate con una interfaccia rest quale ad esempio il Postman. Puoi trovare le specifiche delle API ('references') nella documentazione di ciascun servizio.

- Postman (<https://www.getpostman.com/>) è disponibile sia come applicazione su MacOS, Windows, Linux oppure come plug-in per il browser Chrome. Se hai Chrome già installato puoi installare il plug-in da qui (<https://chrome.google.com/webstore>) cercando 'Postman'.



- Dopo aver installato il plug-in sarà possibile usarlo accedendo alla sezione 'apps' di Chrome. Clicca l'icona Postman per avviare l'applicazione.



- Nella pagina 'API reference' del servizio IBM Cloud Watson Assistant (<https://www.ibm.com/watson/developercloud/assistant/api>) è possibile trovare tutte le informazioni relative alle chiamate API disponibili service. Il comando di base è quello che permette di recuperare la lista dei propri workspaces.
- In Postman aggiungi la seguente richiesta **GET**:
Url: <https://gateway.watsonplatform.net/assistant/api/v1/workspaces?version=2018-07-10>
Authorization Type: Basic Auth – inserisci username e password del servizio (vedi step 1)
Headers: Content-Type: application/json



- Nota: Cliccando il pulsante ‘Salva come...’ puoi anche salvare la richiesta per usarla anche in futuro. (esempio: Request Name: Assistant_GetWorkspaces, Collection: Watson Services).

The screenshot shows the Postman interface with a request named 'Conv_GetWorkspaces'. The request method is 'GET' and the URL is 'https://gateway.watsonplatform.net/conversation/api/v1/workspaces?version=2017-05-26'. The 'Headers' tab is selected, showing two headers: 'Authorization' (Basic NTdkMmNiZDktZWQ0ZC00ZjQxLWFINTEtN...) and 'Content-Type' (application/json). The 'Save' button is visible at the top right, with a red box highlighting the 'Save As...' option.

- Il risultato è il seguente:

The screenshot shows the Postman interface displaying the response to the GET request. The response status is '200 OK', time is '1025 ms', and size is '1.6 KB'. The JSON response body is shown in the 'Pretty' tab, detailing a workspace named 'CIBOT' with its language set to 'it'. The workspace has a description: 'Un consulente virtuale che aiuta a scegliere una dieta sana ed equilibrata - Party Cloud - 20 settembre 2018 - Milano'. The workspace ID is redacted. The response also includes system settings, learning opt-out status, and pagination information.

- Ora possiamo recuperare le informazioni del workspace.

Url: https://gateway.watsonplatform.net/assistant/api/v1/workspaces/<il TUO workspace_id dal risultato precedente >?version=2018-07-10

Authorization Type: come nello step precedente

Headers: come nello step precedente

- Il risultato è il seguente:

The screenshot shows the Postman interface displaying the detailed response for a specific workspace. The workspace is named 'CIBOT' with its language set to 'it'. The workspace has a description: 'Un consulente virtuale che aiuta a scegliere una dieta sana ed equilibrata - Party Cloud - 20 settembre 2018 - Milano'. The workspace ID is redacted. The response includes system settings, learning opt-out status (false), and a status of 'Available'.



- Il prossimo passo consiste nell'inviare (richiesta **POST**) un messaggio iniziale per avviare la conversazione

Url: <https://gateway.watsonplatform.net/assistant/api/v1/workspaces/<il TUO workspaceid dal risultato precedente>/message?version=2018-07-10>

Authorization Type: come nello step precedente

Headers: come nello step precedente

Body: (raw) come indicato di seguito

```
{"input":  
    {"text": ""}  
}
```

- Il risultato è il seguente json:

```
{  
    "intents": [],  
    "entities": [],  
    "input": {  
        "text": ""  
    },  
    "output": {  
        "generic": [  
            {  
                "response_type": "text",  
                "text": "Ciao sono \"Cibot\""  
            },  
            {  
                "response_type": "text",  
                "text": "Sono il tuo nuovo amico nutrizionista. Ho bisogno  
di qualche informazione su di te per creare la tua dieta ideale. Sei uomo o  
donna? Quanti anni hai? Sei vegetariano o vegano? Sei intollerante o  
allergico a qualche alimento?"  
            }  
        ],  
        "text": [  
            "Ciao sono \"Cibot\"",  
            "Sono il tuo nuovo amico nutrizionista. Ho bisogno di qualche  
informazione su di te per creare la tua dieta ideale. Sei uomo o donna?  
Quanti anni hai? Sei vegetariano o vegano? Sei intollerante o allergico a  
qualche alimento?"  
        ],  
        "nodes_visited": [  
            "Benvenuto",  
            "node_12_1535551422052",  
            "node_2_1535975259621"  
        ],  
        "log_messages": []  
    },  
    "context": {  
        "conversation_id": "62b4c00d-7c8a-427e-b4b2-4e06691f8c8b",  
        "system": {  
            "dialog_stack": [  
                {  
                    "dialog_node": "node_2_1535975259621"  
                }  
            ],  
            "dialog_turn_counter": 1,  
            "dialog_request_counter": 1,  
            "_node_output_map": {  
                "Benvenuto": {  
                    "0": [  
                        "text": "Ciao sono \"Cibot\"",  
                        "text": "Sono il tuo nuovo amico nutrizionista. Ho bisogno di qualche  
informazione su di te per creare la tua dieta ideale. Sei uomo o donna?  
Quanti anni hai? Sei vegetariano o vegano? Sei intollerante o allergico a  
qualche alimento?"  
                    ]  
                }  
            }  
        }  
    }  
}
```



```
          0
        ],
      },
      "node_2_1535975259621": {
        "0": [
          0
        ]
      }
    },
    "profilo": null
  }
}
```

- Questo risultato mostra come sia stata fornita correttamente la risposta di benvenuto, ma anche come il Watson Assistant abbia instanziato un contesto, ritornato nel json; tale contesto **deve** essere usato nella chiamata successiva come una sorta di memoria (visto che le API rest non ne hanno)
- Ora prova ad inviare una richiesta POST per fornire alla chatbot le informazioni che ha richiesto

Url: come nello step precedente

Authorization Type: come nello step precedente

Headers: come nello step precedente

Body: (raw) come indicato in seguito → [Context](#) copia il context ricevuto in output dalla richiesta precedente.

```
{"input":
  {"text": """Sono donna ho circa 20 anni, sono vegetariana e non sono intollerante"""},
  "context": {copia il context ricevuto dalla richiesta precedente}
}
```

- Ed ecco qui che la chatbot ci fornisce la sua risposta!

```
{
  "intents": [
    {
      "intent": "chi_sono",
      "confidence": 0.5728160619735718
    }
  ],
  "entities": [
    {
      "entity": "sesso",
      "location": [
        5,
        10
      ],
      "value": "donna",
      "confidence": 1
    },
    {
      "entity": "sys-number",
      "location": [
        20,
        22
      ],
      "value": "20",
      "confidence": 1
    }
  ]
}
```



```
        "confidence": 1,
        "metadata": {
            "numeric_value": 20
        }
    },
    {
        "entity": "regime_alimentare",
        "location": [
            34,
            45
        ],
        "value": "vegetariano",
        "confidence": 0.91
    },
    {
        "entity": "intolleranze",
        "location": [
            48,
            69
        ],
        "value": "nessuna intolleranza",
        "confidence": 1
    }
],
"input": {
    "text": "Sono donna ho circa 20 anni, sono vegetariana e non sono intollerante"
},
"output": {
    "generic": [
        {
            "response_type": "text",
            "text": "Bene, da quello che posso aver capito dovresti essere donna di 20 anni, avere un regime alimentare di tipo vegetariano e avere nessuna intolleranza. Corretto?"
        }
    ],
    "text": [
        "Bene, da quello che posso aver capito dovresti essere donna di 20 anni, avere un regime alimentare di tipo vegetariano e avere nessuna intolleranza. Corretto?"
    ],
    "nodes_visited": [
        "node_2_1535549574036",
        "handler_4_1535549581676",
        "handler_10_1535549596362",
        "handler_7_1535549589062",
        "handler_5_1536332155954",
        "node_2_1535549574036"
    ],
    "log_messages": []
},
"context": {
    "conversation_id": "62b4c00d-7c8a-427e-b4b2-4e06691f8c8b",
    "system": {
        "dialog_stack": [
            {
                "dialog_node": "node_2_1535549574036"
            }
        ],
        "dialog_turn_counter": 2,
        "dialog_request_counter": 2,
        "_node_output_map": {

```



```
    "Benvenuto": {
        "0": [
            0
        ]
    },
    "node_2_1535975259621": {
        "0": [
            0
        ]
    },
    "node_2_1535549574036": {
        "0": [
            0
        ]
    }
},
"profilo": null,
"sesso": "donna",
"regime_alimentare": "vegetariano",
"intolleranze": "nessuna intolleranza",
"eta": 20
}
}
```

Le che hai imparato come funzionano le API del Watson Assistant, potrai usarle in qualsiasi applicazione scritta con qualsiasi linguaggio di programmazione.

Per alcuni linguaggi di programmazione IBM ha anche predisposto i relativi SDK (vedi link in ultima pagina) per velocizzare il processo di creazione della applicazione. Nell'esercizio seguente vedremo come scaricare ed installare una semplice applicazione web scritta in Node.js.



STEP 7: Installazione di una applicazione web Node.js

In questa sezione vedremo come è possibile installare e configurare una semplice applicazione web attraverso la quale utilizzare la chat progettata e realizzata nelle sezioni precedenti. L'applicazione consiste in una interfaccia dove è possibile interagire con la chat (nella parte sinistra) e nello stesso tempo di vedere il json generato dal servizio Watson Assistant durante le interazioni (nella parte destra).

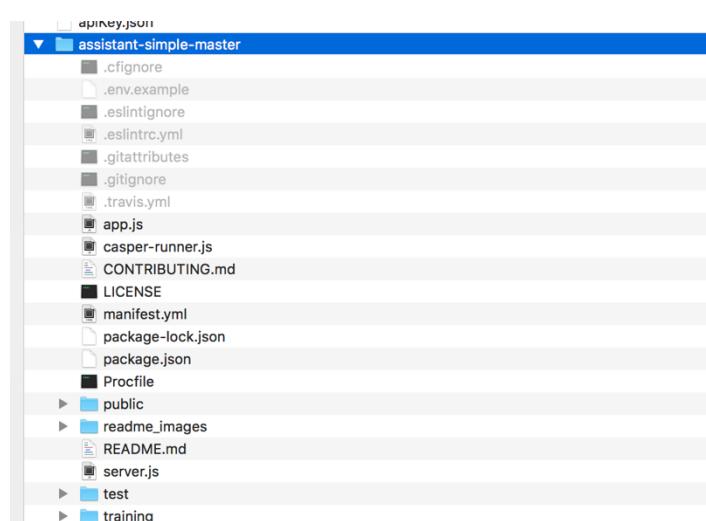
Un esempio è riportato nella figura seguente:

The screenshot shows a user interface for a Node.js application. On the left, there is a "User input" section where a user types "Hi". Below it, a message from the application says "Hello! I'm doing good. I'm here to help you. Just say the word." On the right, there is a "Watson understands" section displaying a JSON log of the interaction. The log shows the user's input and the Watson Assistant's response, including the conversation ID, dialog stack, and intent recognition.

```
1 {
2   "input": {
3     "text": "Hi"
4   },
5   "context": {
6     "conversation_id": "51c02fb2-34bf-4cbf-883c-60822f35f88d"
7     "system": {
8       "dialog_stack": [
9         "root"
10      ],
11      "dialog_turn_counter": 1,
12      "dialog_request_counter": 1
13    },
14    "default_counter": 0
15  }
16 }
```

```
1 {
2   "intents": [
3     {
4       "intent": "greetings",
5       "confidence": 1
6     }
7   ],
8   "entities": []
9 }
```

- Per iniziare è necessario scaricare il file `assistant-simple-master.zip` che contiene tutti i file che compongono l'applicazione. Il file si trova a questo link:
<https://github.com/watson-developer-cloud/assistant-simple/archive/master.zip>
- Spacchettare il file in una directory locale. Aprendo la directory si ottiene il seguente elenco di file e sotto-directory:



- Copia o rinomina il file `.env.example` in `.env`



- Apri il file .env con un qualsiasi editor di testo e sostituisci al posto di <workspace-id> <conversation-username> e <conversation-password> i valori relativi alla tua chat, che, come descritto allo step 2, sono riepilogate nel menu ⚙ del workspace.

```
1 # Environment variables
2 WORKSPACE_ID=<workspace-id>
3 ASSISTANT_USERNAME=<conversation-username>
4 ASSISTANT_PASSWORD=<conversation-password>
5
```

- Salva il file

Test della applicazione in locale

Per eseguire il test in locale è necessario aver installato il runtime node.js sulla propria stazione di lavoro come descritto alla sezione ‘Prerequisiti’ della Introduzione

- Apri una riga di comando e posizionati sulla directory della applicazione
- Esegui il comando
 npm install
- Esegui il comando
 npm start
- Apri il browser alla pagina <http://localhost:3000> per provare l’applicazione in locale.

Installazione della applicazione su IBM Cloud

L’installazione consiste nel creare un runtime Cloudfoundry Node.js su IBM Cloud utilizzando i file della applicazione.

Prima di procedere con la procedura di installazione è necessario apportare qualche piccola modifica al file **manifest.yml** che si trova nella directory principale della applicazione

- Apri il file manifest.yml:

```
1 ---
2 declared-services:
3   my-watson-assistant-service:
4     label: conversation
5     plan: free
6 applications:
7   - name: watson-assistant-simple
8     command: npm start
9     path: .
10    memory: 256M
11    instances: 1
12    services:
13      - my-watson-assistant-service
14    env:
15      NPM_CONFIG_PRODUCTION: false
```

- Nella sezione **applications**: cambia il nome **watson-assistant-simple** con un nome univoco a tua scelta che sarà parte dell’indirizzo web della applicazione



- Nella sezione services: cambia il nome **my-watson-assistant-service** con il nome del tuo servizio di Watson Assistant specificato allo step 1 (lo puoi trovare sia nella pagina principale del servizio come illustrato di seguito ma anche nella dashboard)

The screenshot shows the IBM Cloud service catalog interface. On the left, there's a sidebar with 'Manage' selected, followed by 'Service credentials', 'Plan', and 'Connections'. The main area displays a service named 'Assistant: Conversation-FG' with a red oval highlighting it. Below the service name, it says '0.15% Used | 9985 Api calls available'. To the right of the service name are 'Details', 'Location: United Kingdom', 'Org: francesca_gigante@it.ibm.com', and 'Space: dev_uk'. At the top right of the main area are 'Catalog', 'Docs', 'Support', and 'Manage' buttons.

- Esegui il comando:
- Imposta la riga di comando sulla directory in cui
- Esegui il comando di installazione della applicazione in cloud:
cf push
- Wow! La tua applicazione è ora su IBM Cloud!!** Puoi accedervi cliccando sull'indirizzo web indicato dall'output del comando, si aprirà il browser e potrai usare la tua chatbot, potrai dare il link ai tuoi amici / conoscenti e collaboratori per farla utilizzare ma non finisce qui..... man mano che la chatbot verrà usata il servizio di Watson Assistant raccoglie i log che potrai vedere alla voce di menu non solo per vedere quello che accade ma anche per fare tutti i miglioramenti al volo!
- Ricordati che accedendo alla documentazione del servizio disponibile qui <https://console.bluemix.net/docs/services/conversation> potrai avere tutte le informazioni aggiuntive che per motivi di tempo non sono state inserite in questo laboratorio..... e sono davvero tantissime!

Modifica della applicazione

Se vuoi modificare l'applicazione puoi esplorare i file sorgenti ed apportare le modifiche che preferisci.

Puoi provare le seguenti modifiche:

- Modifica i file .js per modificare la logica dell'applicazione.
- Modifica i file .html per cambiare l'aspetto grafico della pagina web dell'applicazione.
- Aggiorna l'istruzione della chatbot aggiungendo e/o modificando gli intenti e i nodi di dialogo attraverso il tool di gestione del workspace

E ORA NON CI RESTA CHE AUGURARTI BUON DIVERTIMENTO!



Riferimenti

- La documentazione di questo hands-on <https://github.com/francesca-gigante/cibot-hands-on>
- Watson Developer Resources <http://www.ibm.com/watson/developer>
Questa pagina permette l'accesso a tutte le informazioni relative ai servizi, incluse demo ed i link a Youtube, ai blog, ai webinar, e a tutte le risorse messe a disposizione dalle community di sviluppatori
- IBM Code Patterns: <https://developer.ibm.com/code/technologies/artificial-intelligence/>
- IBM Code Open Source Projects from IBM: <https://developer.ibm.com/code/open/projects/>
- IBM Watson SDK: <https://github.com/watson-developer-cloud>



Acknowledgements and Disclaimers

Copyright © 2017 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed “as is” without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice. Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM



products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular, purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right. IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services®, Global Technology Services®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli® Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.