

Build a Watsonx AI Chatbot with Db2

Jesse Gorzinski
Sanjula Ganepola
IBM

IBM i

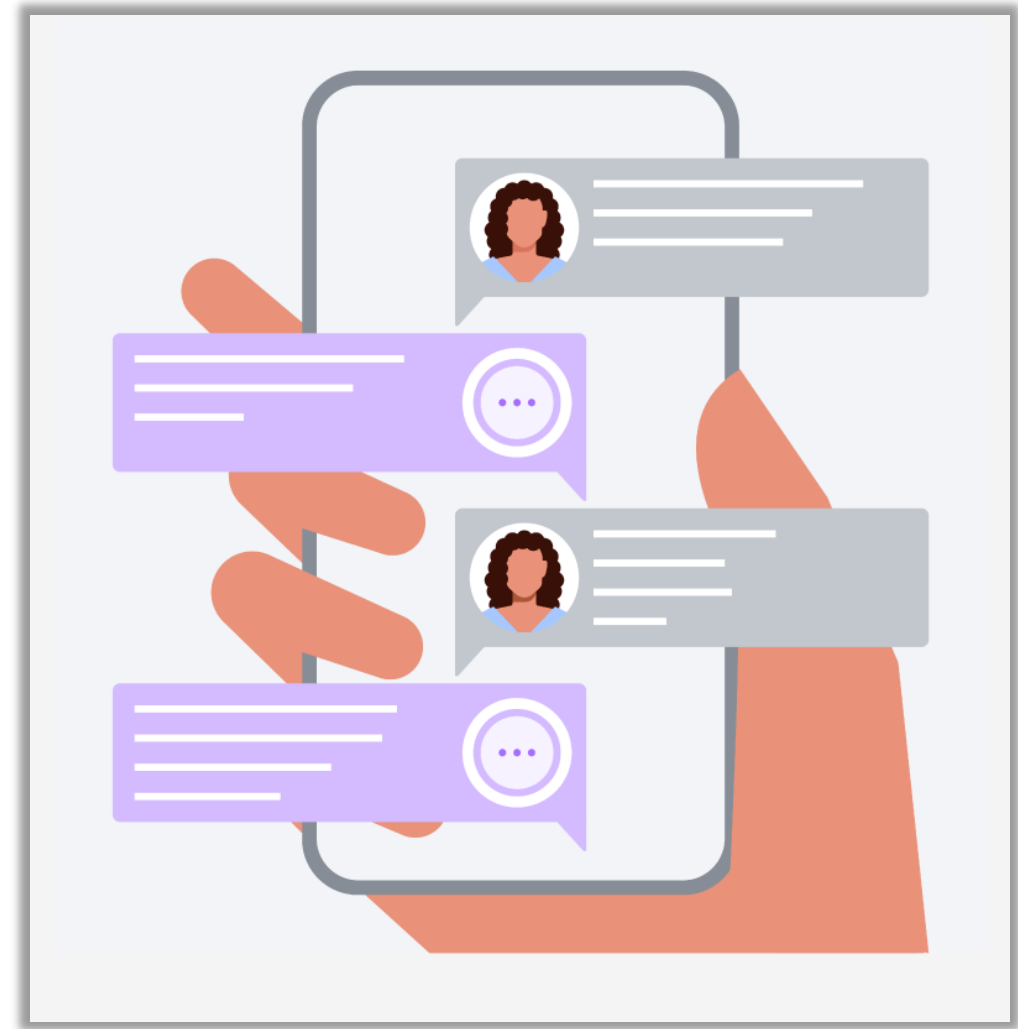
Agenda

- Overview of Watsonx Assistant
- Planning Your Assistant
- Terminology You Should Know
- Let's Build a Chatbot
- Integrate with Db2
- Next Steps

Have you ever built a chatbot?

Choose 1:

1. Yes, with Watsonx Assistant
2. Yes, with another tool
3. No, but we want to
4. No, we don't want to
5. No, never thought about it








Overview of Watsonx Assistant

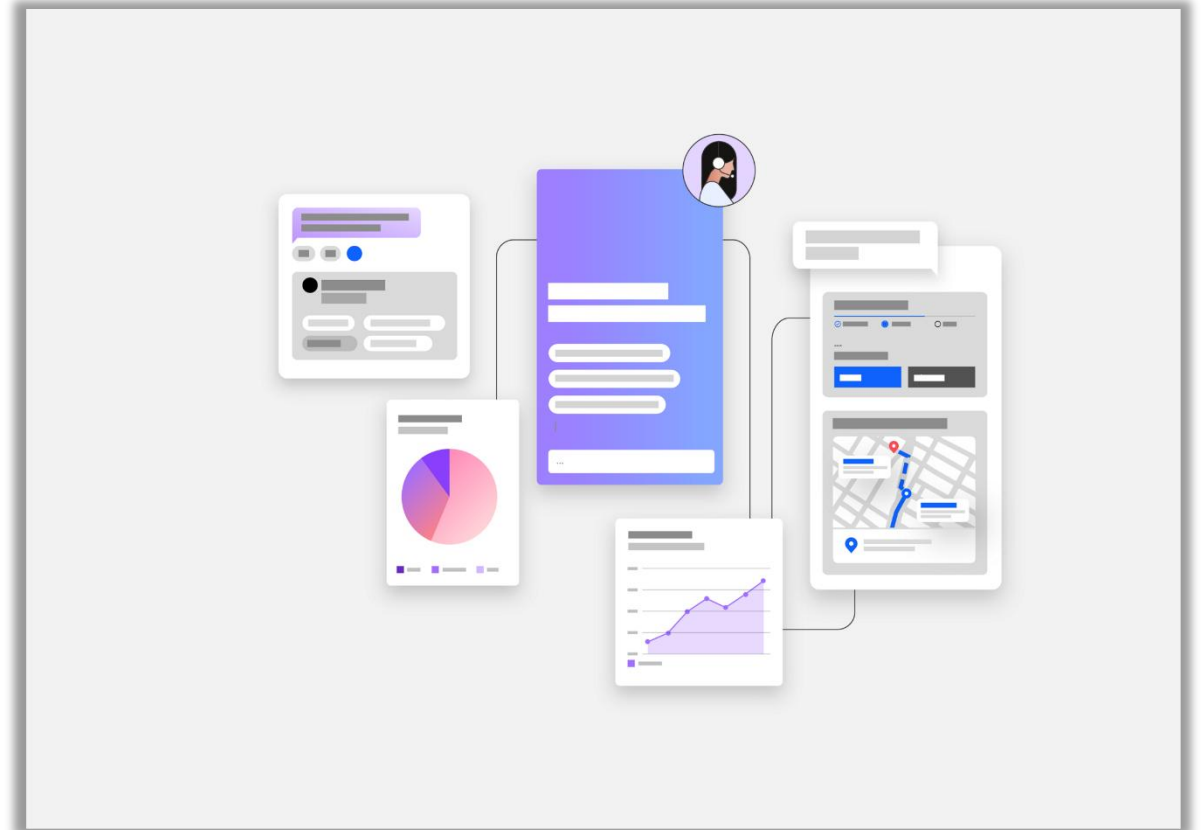
What is Watsonx Assistant?

“Conversational AI solution that empowers anyone to effortlessly build generative AI Assistants that deliver frictionless self-service experiences to customers”

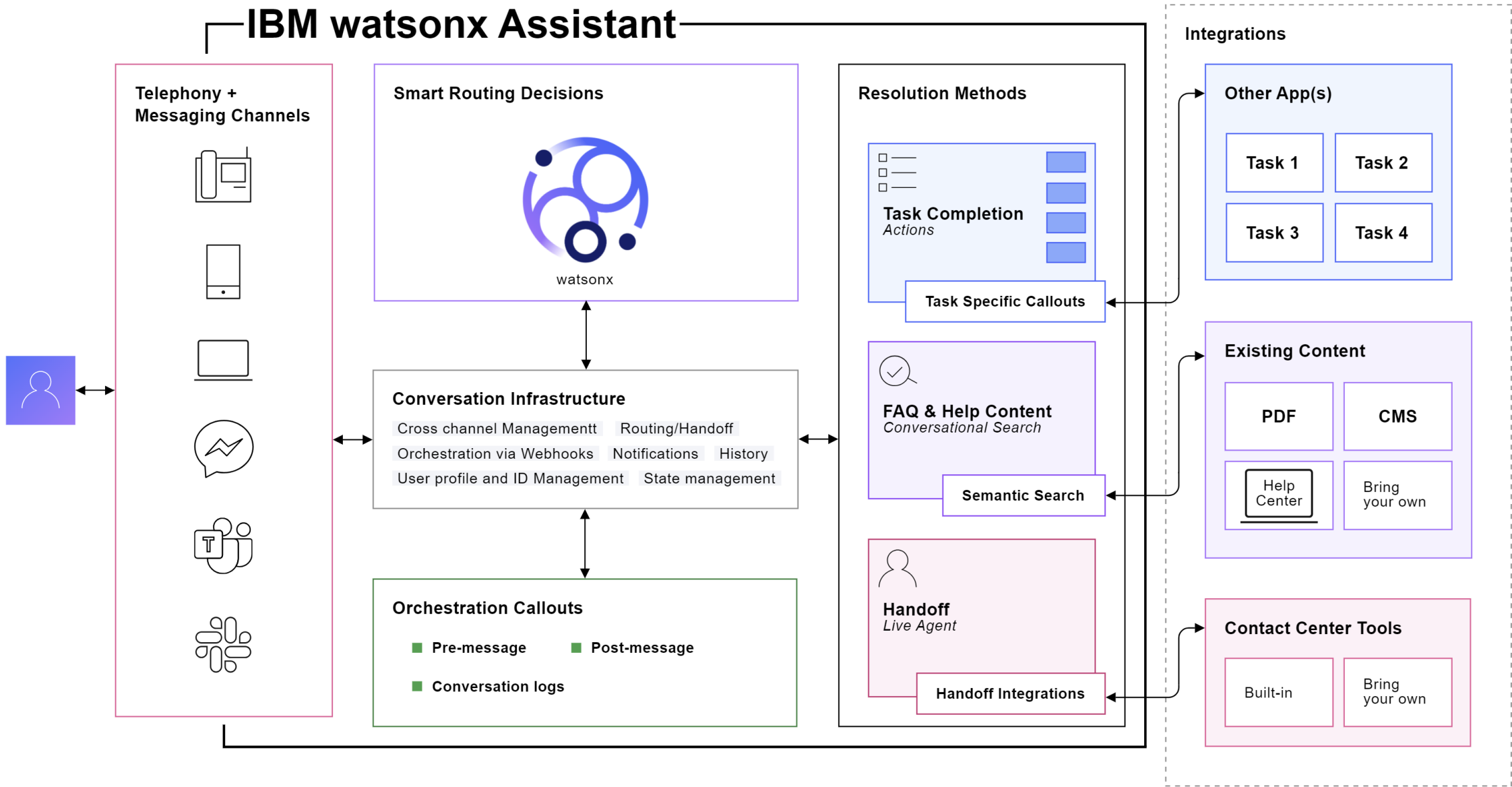


Use Watsonx Assistant to build your own branded live chatbot

-  Bring the assistant to your customers, where they are
-  Create AI-driven conversational flows
-  Embed existing help content
-  Connect to your customer service teams
-  Track customer engagement and satisfaction



How Does It Work?



Planning Your Assistant

Before building an assistant...

1. Select an initial channel

- Where will customers use your assistant?

2. Select starting domain

- What topic domain can your assistant unify and automate?

3. Outline frequent topics

- What are the most frequent topics which are mostly informational in nature?

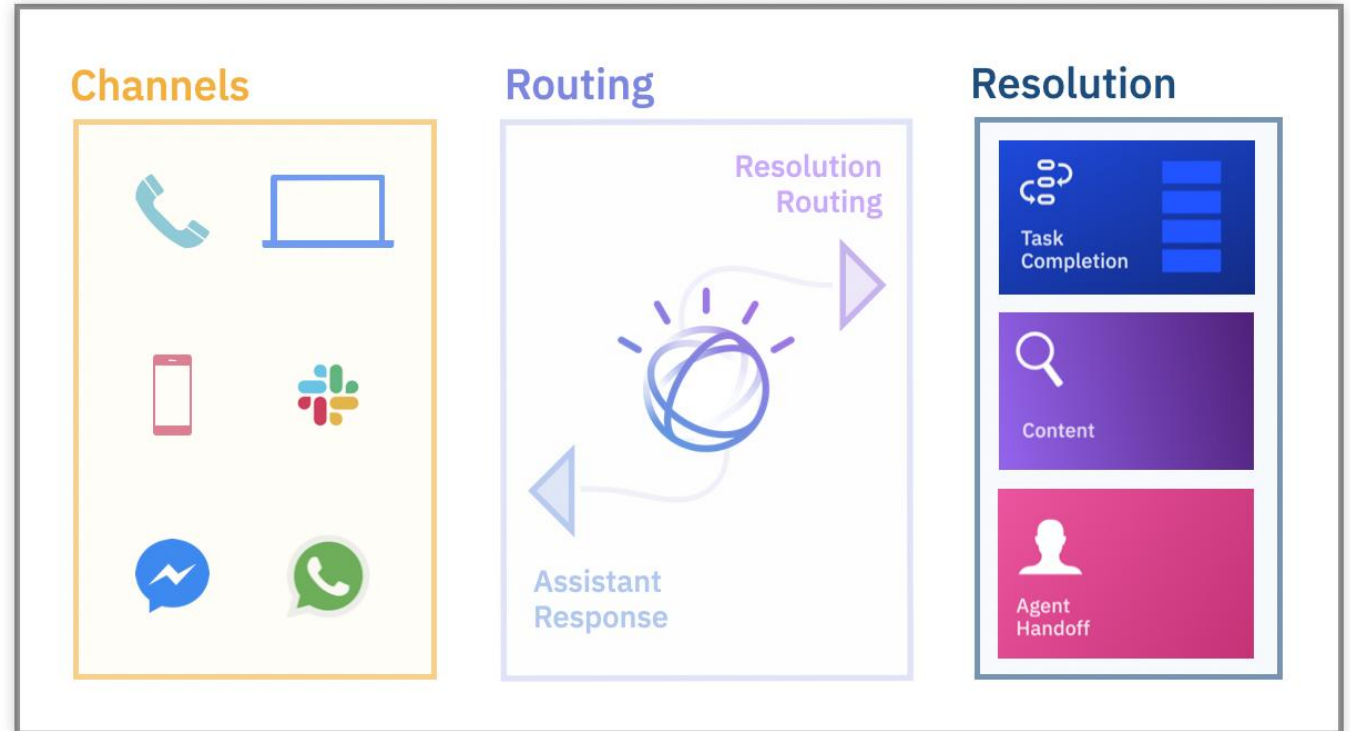
4. Collect content sources

- What help content is available to customers today?

5. Plan handoff strategy

- How should customers be routed to human agents?

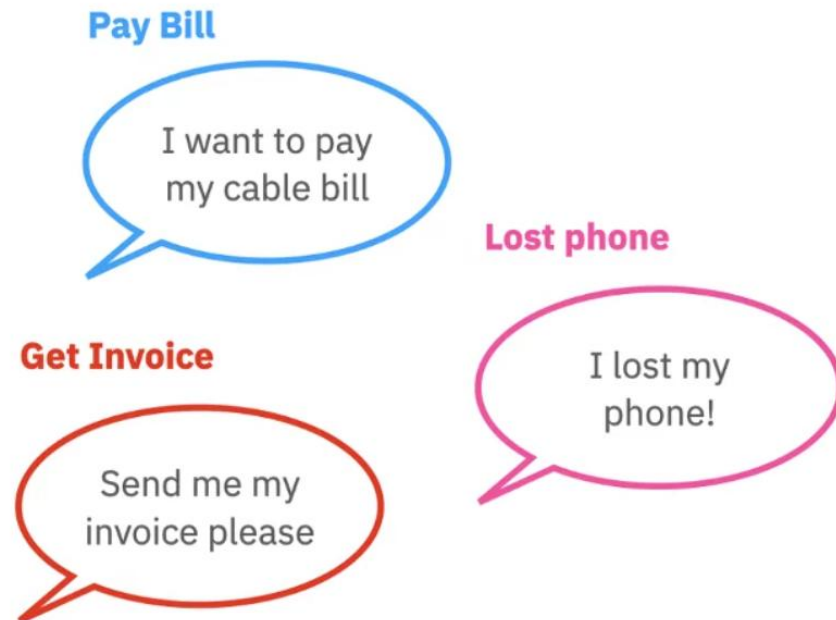
Your first assistant



Terminology You Should Know

What are Actions and Phrases?

- An action is a problem or a task that your customer wants to resolve
- Phrases are what customers type or say to start an action
- Add multiple phrases to better train your assistant to recognize what customers want



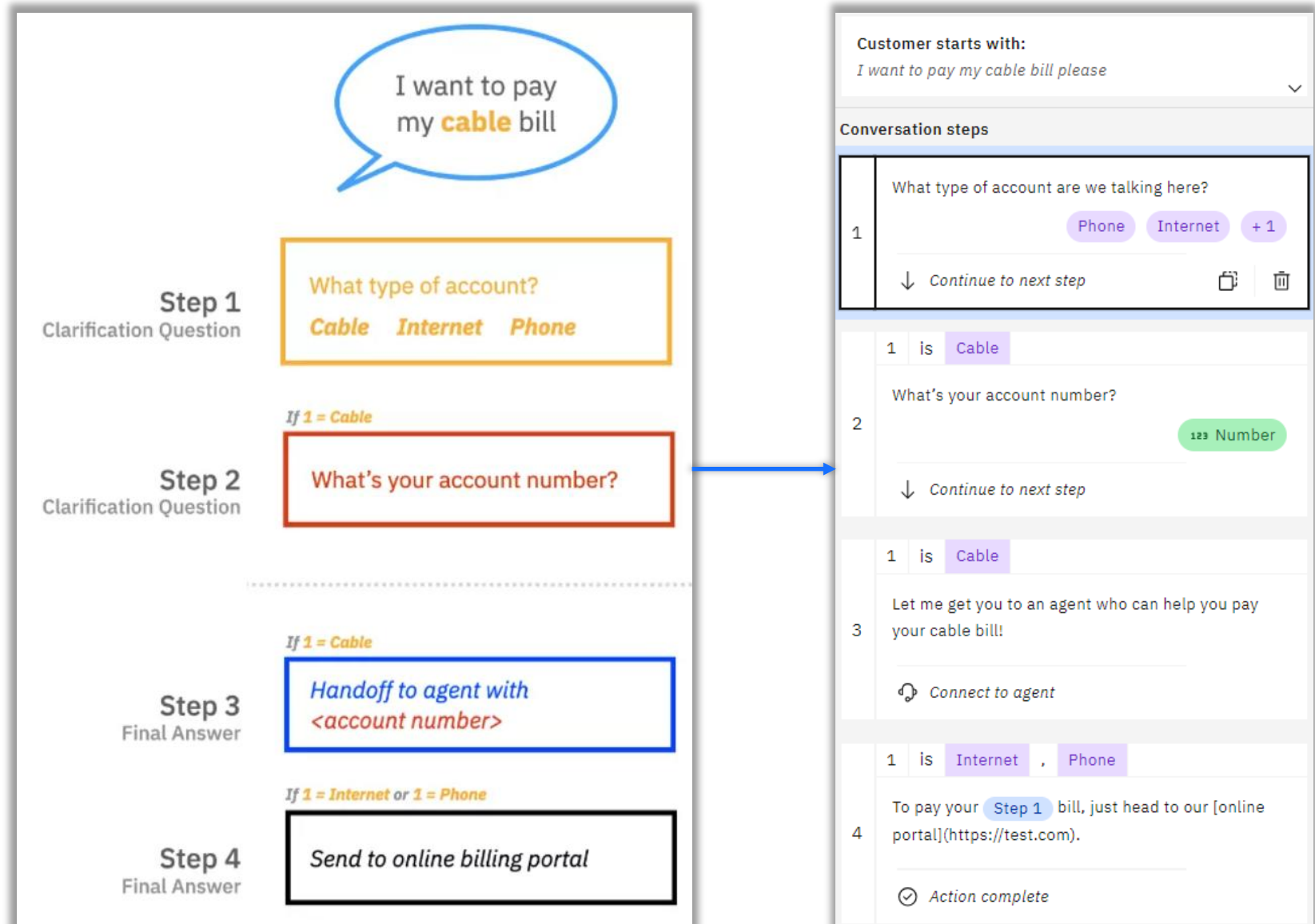
A screenshot of a "New action" dialog box. It has a title bar with a close button (X). The main text asks, "What does your customer say to start this interaction?". Below this is a text input field containing the example: "Example: I want to pay my credit card bill.". At the bottom, there are two buttons: "Cancel" (dark grey) and "Save" (light grey).

A screenshot of the "Add example phrases" interface. It has a title bar with icons for edit, up, and down. The main text explains: "Enter phrases that a customer types or says to start the conversation about a specific topic. These phrases determine the task, problem, or question your customer has. The more phrases you enter, the better your assistant can recognize what the customer wants." Below this is a section titled "Enter phrases your customer might use to start this action" with a "Total: 9" indicator. It contains a list of phrases, each with a delete icon (X) on the right:

- Enter a phrase (input field)
- What is my account balance right now
- See balance
- How much is my next bill
- How much is my account balance?
- How much do I owe
- How much do I have to pay?
- Check bill amount
- Check balance
- Check account balance

What are Steps and Conditions?

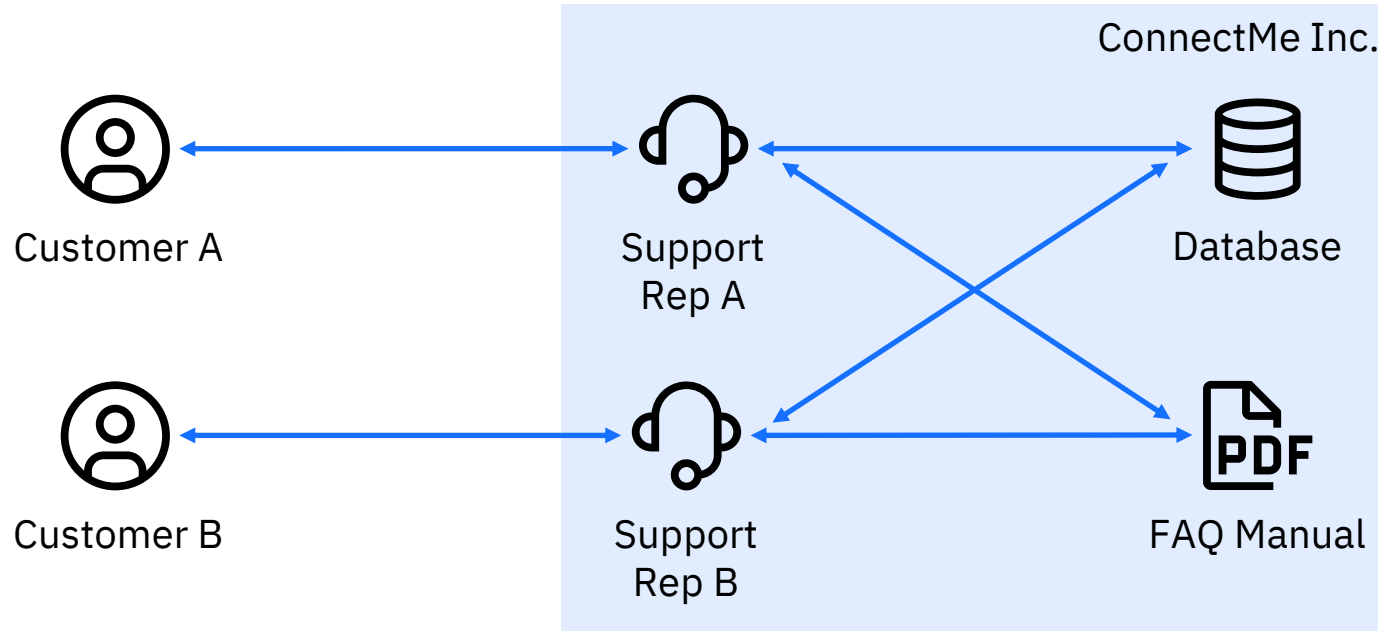
- A step is a back-and-forth interaction between your assistant and your customer
- Conditions are requirements which must be met for the step to be triggered



Let's Build a Chatbot

Demo Example – ConnectMe Inc.

ConnectMe Inc. is a telecommunications company with cable, internet, and phone services



Goals

- Streamline bill payment services
- Update account data in real-time
- Provide 24/7 support

Create Watsonx Assistant Instance

IBM Cloud

Search resources and products...

Catalog Manage Sanjula Ganepola's Account

Catalog /

watsonx Assistant

IBM watsonx Assistant lets you build conversational interfaces into any application, device, or channel.

Create About

Type

Service

Provider

IBM

Last updated

10/18/2024

Category

AI / Machine Learning

Compliance

EU Supported

HIPAA Enabled

IAM-enabled

Location

Sydney

Frankfurt

London

Tokyo

Washington DC

Dallas

Related links

API docs

Docs

Terms

Select a location

Washington DC (us-east)

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or location: [United States](#)

Plan	Features and capabilities	Pricing
Trial	<div>30 Day trial period (no credit card required)</div> <div>Up to 5,000 MAUs</div> <div>Up to 50,000 messages per month</div> <div>--- Features ---</div> <div>- Everything available in Plus</div> <div>- A low risk way to test paid features</div> <div>- Export or analyze chat logs anywhere you want by connecting to webhooks and API endpoints</div> <div>--- Limits ---</div> <div>Actions:</div> <div>- 10 assistants per instance</div> <div>- Up to 100 actions</div> <div>- 5 custom extensions per assistant</div> <div>- 7 days of usage analytics</div> <div>- Session inactivity timeout 5 minutes</div> <div>- Services are deleted after 30 days of inactivity</div> <div>Dialog:</div> <div>- Same as Lite</div>	Free
Lite	<div>Everything you need to get started, free for as long as you need it</div> <div>Up to 1,000 unique monthly active users (MAUs) chatting with your assistant</div> <div>Up to 10,000 messages per month</div> <div>--- Features ---</div> <div>- World-class conversational AI</div>	Free

Summary

watsonx Assistant

Free

Location: Washington DC

Plan: Trial

Service name: watsonx Assistant-o7

Resource group: Default

☐ I have read and agree to the following license agreements:

[Terms](#)

Create

Add to estimate

💡

30 DAY TRIAL!

↓

Create Watsonx Assistant

Video Demo: <https://www.youtube.com/watch?v=JTcrllB82QE>

Create a new assistant

Assistant name

ConnectMe

Your assistant name will be kept internally and not visible to your customers

Description (optional)

Chatbot assistant focused on helping ConnectMe customers.

57/128

Assistant language

English (US)

This is the language your assistant will speak.

Cancel

Create assistant

Web chat

Draft

Close

Save and exit

Style

Launcher

Home screen

Live agent

Suggestions

Security

Embed

Resources

Customize your chat UI

Update the style to match your brand and your website. A developer can also add more advanced styling changes with code. [Learn more](#)

Assistant's name as known by customers

ConnectMe

Intended purpose

☒ Standard: For virtual agents and customer support experiences.

☐ Carbon for AI: For use in internal IBM products.

Choose a theme

Light

Dark

Primary color

#FFFFFF

Secondary color

#3D3D3D

Chat header

Accent color

#0354E9

Significant and interactive objects

Add an avatar image

ConnectMe

Hi! I'm a virtual assistant.
How can I help you today?

Example: Find nearby location


Example: Check account balance

Example: See how I can help


Restart conversation

Create Action From Scratch and Set Initial Phrase

How would you like to build your action?



Start from scratch
Build with actions using your own use case.



Quick start with templates
Use one of our pre-built templates and use cases.

New action

What does your customer say to start this interaction?

I want to pay my bill please

Cancel

Save

Add More Example Phrases

Add example phrases:

Enter phrases that a customer types or says to start the conversation about a specific topic. These phrases determine the task, problem, or question your customer has.

The more phrases you enter, the better your assistant can recognize what the customer wants.

Enter phrases your customer might use to start this action

Total: 5

Enter a phrase

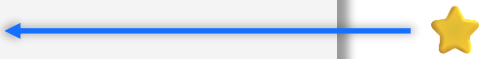
gotta pay you

my internet bill needs to be paid

I need to pay a past due bill

pay bill

I want to pay my bill please



Create First Step

I want to pay my bill please

Customer starts with:
I want to pay my bill please

Conversation steps

1

What type of account are we talking about here?

↓ Continue to next step

EditorVisualization

Step 1

Is takenwithout conditions

Set variable values

Assistant says

B I @ f_x ↺ 🖼 🎵 📺 📷 ⏸

What type of account are we talking about here?

Define customer response

And then

↓ Continue to next step

Add Custom Response Options

Define customer response

System

Options

Confirmation

Free text

Regex

Number

Date

Time

Saved

Options

Enable customers to select from a set of choices. If more than 4, options show as a list.

As buttons

Checking Savings 401 (k) Roth IRA

As a list

Pay Bill

Edit response

Enter a list of options or pull from a variable to generate a dynamic list of options. [Learn more](#)

Dynamic

Add synonyms +

Option 1

Cable

Option 2

Internet

Option 3

Phone

Option 4

Example: Savings account

Cancel

Apply

Create Another Clarification Step

I want to pay my bill please

EditorVisualization

Customer starts with:
I want to pay my bill please

Conversation steps

1

What type of account are we talking about here?

CableInternet+ 1

Continue to next step

2

What's your account number?

Continue to next step

New step +

Step 2

Is takenwithout conditions

Set variable valuesfx

Assistant says

What's your account number?

Define customer response

System

Options

Confirmation

Free text

Regex

123 Number

Date

Time

Saved

Number

Enable customers to enter a numeric amount.

As numerals

2,000

As alphanumeric text

two thousand

Add Condition

Step 2

Is taken

with conditions

Set variable values

Conditions

1 condition

If

All

of this is true:

1. What type of account are we tal...

is

Cable

and

Add condition

New condition group

Create an Agent Handoff Step

I want to pay my bill please

Editor

Visualization

Customer starts with:
I want to pay my bill please

Conversation steps

1

What type of account are we talking about here?

Cable Internet +1

Continue to next step

2

1 is Cable

What's your account number?

123 Number

Continue to next step

3

1 is Cable

Let me get you to an agent who can help you pay your cable bill!

Continue to next step

New step +

Step 3

Is taken with conditions

Set variable values fx

Conditions 1 condition

If All of this is true:

1. What type of account are we talk... is Cable

and Add condition +

New condition group +

Assistant says

B I @ fx ↺ ↻ 📎 🎧 📄 📁 📧 📧 📧

Let me get you to an agent who can help you pay your cable bill!

Define customer response

And then

Continue to next step

Connect to Agent

And then

↓ Continue to next step

Continue to next step

Re-ask previous step(s)

Go to a subaction

Use an extension

Search for the answer

Connect to agent

End the action

Connect to agent

Transfer the customer to someone on your support team.

ⓘ Requires a [a service desk integration](#)

Settings

×

Connect to agent

This will transfer customers based on the integrations you've set up with your assistant. [Learn more](#)

Response if agents are online

Let's send you to an available agent.

Response if agents are offline

There are no agents available at this time. When one becomes available, we'll connect you.

Message to agent (Optional)

This user wants to pay their cable bill. Account number is 2. ****What's your account...**

Route to specific queue (Optional)

Select integration ▼

Cancel

Apply

Create Final Response Step

Step 4

Is taken

with conditions

Set variable values

f_x

Conditions

2 conditions

If

All

of this is true:

1. What type of account are we tal...

is

Internet

X

and

1. What type of account are we tal...

is

Phone

X

and

Add condition

New condition group

Assistant says

B

I

f_x

To pay your

1. What type of account are we talking al

bill, you can head to our

link

Edit Response in JSON View

Assistant says

JSON view

abc

```
1 {  
2   "generic": [  
3     {  
4       "response_type": "text",  
5       "values": [  
6         {  
7           "text_expression": {  
8             "concat": [  
9               {  
10              "scalar": "To pay your "  
11            },  
12            {  
13              "variable": "step_919"  
14            },  
15            {  
16              "scalar": " bill, you can head to our [link]  
(https://connectme/onlineportal.com)."  
17          }  
        ]  
      }  
    ]  
  }  
}
```

End the Action

And then

✓ End the action

Continue to next step

Re-ask previous step(s)

Go to a subaction

Use an extension

Search for the answer

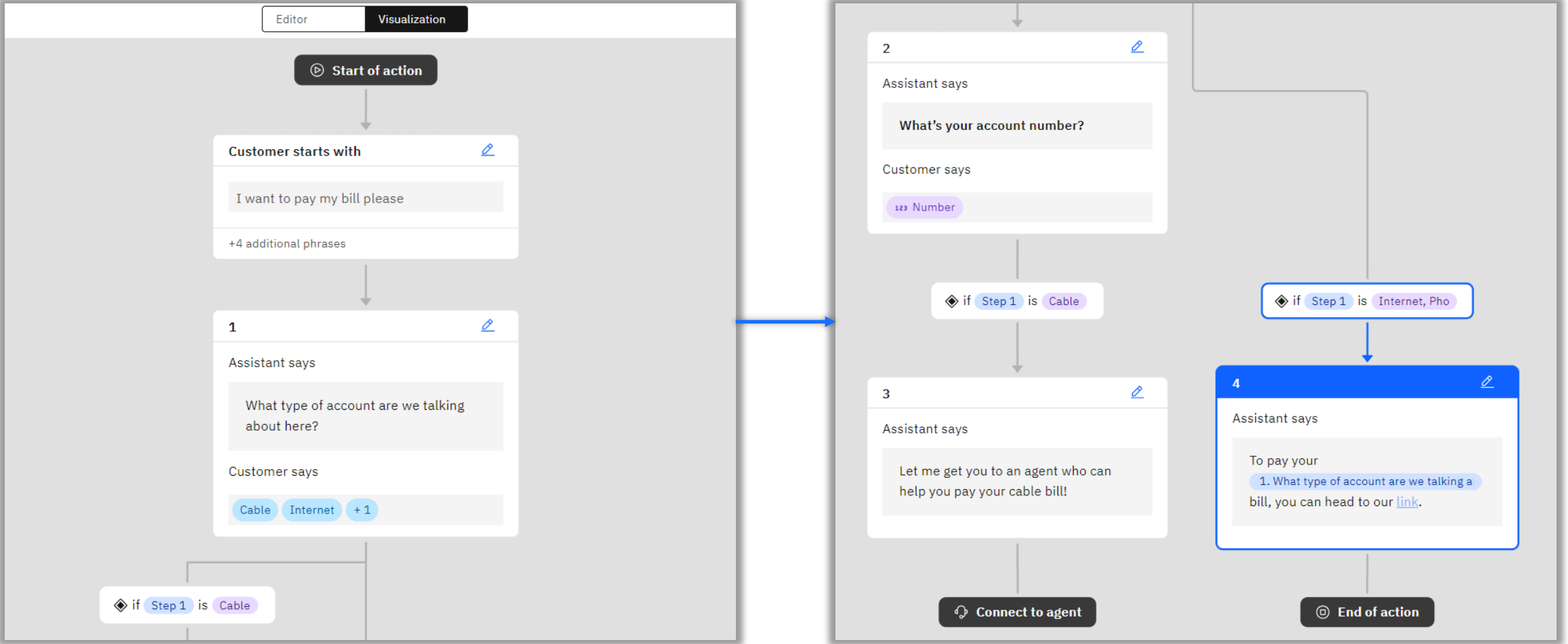
Connect to agent

End the action


End the action

Make this a final step that completes the action.

Use Visualization Tool



Preview Your Action


 3:47 PM

Greet customer [default]

Welcome, how can I assist you?

You 3:47 PM

I want to pay bill

 3:47 PM

I want to pay my bill please recognized

What type of account are we talking about here?


Cable

Internet

Phone

You 3:47 PM


Cable

 3:47 PM


What's your account number?

You 3:47 PM

1111

 3:47 PM


Let me get you to an agent who can help you pay your cable bill!

 Connect to an agent

Live agent support

Let's send you to an available agent.

Connect to agent



Integrate with Db2

Have you ever used Mapepire?

1. Yes, mapepire-java
2. Yes, mapepire-python
3. Yes, mapepire-js
4. Yes, more than one of the above SDKs
5. No, but heard of it
6. No, never heard of it

What is Mapepire?

Welcome to Mapepire

A cloud-friendly IBM i database access layer, built with simplicity and performance in-mind.

Find out more →

Pick your client language ⓘ

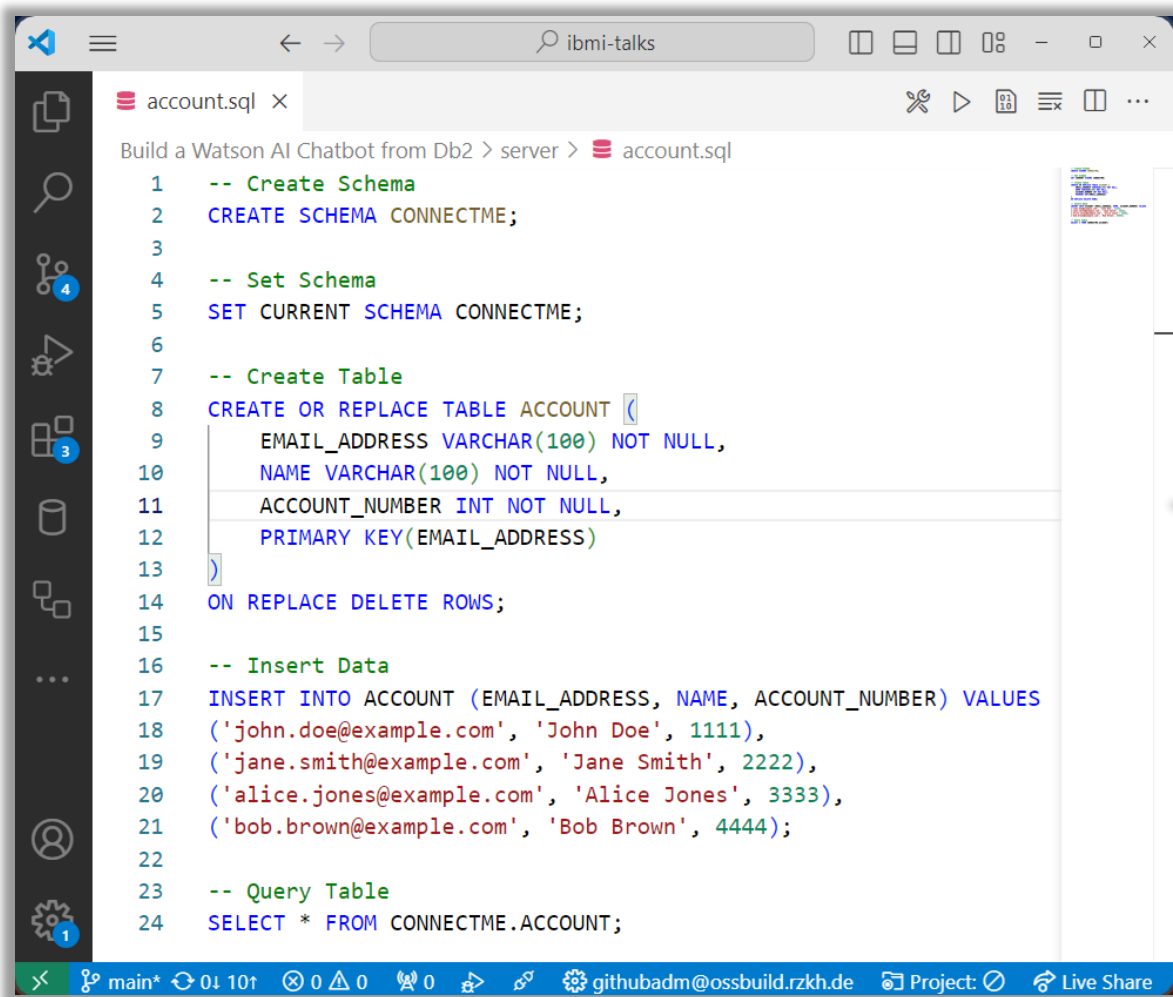


*Super easy to use way to access
Db2 for i from any application*



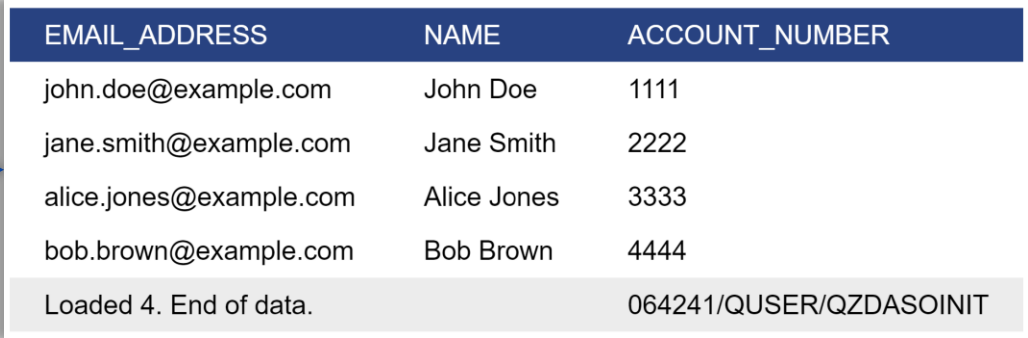
Create Table

Video Demo: <https://www.youtube.com/watch?v=RmDchLsPc18>



The screenshot shows a code editor window with a file named 'account.sql'. The code contains SQL statements to create a schema, set the current schema, create a table named 'ACCOUNT' with columns 'EMAIL_ADDRESS', 'NAME', and 'ACCOUNT_NUMBER', and insert four rows of data. The table has a primary key on 'EMAIL_ADDRESS'. The code is as follows:


```
1  -- Create Schema
2  CREATE SCHEMA CONNECTME;
3
4  -- Set Schema
5  SET CURRENT SCHEMA CONNECTME;
6
7  -- Create Table
8  CREATE OR REPLACE TABLE ACCOUNT (
9      EMAIL_ADDRESS VARCHAR(100) NOT NULL,
10     NAME VARCHAR(100) NOT NULL,
11     ACCOUNT_NUMBER INT NOT NULL,
12     PRIMARY KEY(EMAIL_ADDRESS)
13 )
14 ON REPLACE DELETE ROWS;
15
16 -- Insert Data
17 INSERT INTO ACCOUNT (EMAIL_ADDRESS, NAME, ACCOUNT_NUMBER) VALUES
18 ('john.doe@example.com', 'John Doe', 1111),
19 ('jane.smith@example.com', 'Jane Smith', 2222),
20 ('alice.jones@example.com', 'Alice Jones', 3333),
21 ('bob.brown@example.com', 'Bob Brown', 4444);
22
23 -- Query Table
24 SELECT * FROM CONNECTME.ACCOUNT;
```



The table displays the data inserted into the 'ACCOUNT' table. It has three columns: 'EMAIL_ADDRESS', 'NAME', and 'ACCOUNT_NUMBER'. The data is as follows:

EMAIL_ADDRESS	NAME	ACCOUNT_NUMBER
john.doe@example.com	John Doe	1111
jane.smith@example.com	Jane Smith	2222
alice.jones@example.com	Alice Jones	3333
bob.brown@example.com	Bob Brown	4444
Loaded 4. End of data.		064241/QUSER/QZDASOINIT

Create REST Server

 Server Source Code: [server.js](#)

```
JS server.js X
home > GITHUBADM > connectme > server > JS server.js > createSQLJob

24 app.get('/accounts', async (req, res) => {
25   // Extract and verify required query parameters
26   const { EMAIL_ADDRESS } = req.query;
27
28   let job;
29   try {
30     // Create SQL job
31     job = await createSQLJob();
32
33     // Execute query
34     const query = `SELECT * FROM CONNECTME.ACCOUNT WHERE EMAIL_ADDRESS = ?`;
35     const parameters = [EMAIL_ADDRESS];
36     const result = await job.execute(query, { parameters: parameters });
37
38     // Check result
39     if (result.success && result.data.length > 0) {
40       res.json({
41         EMAIL_ADDRESS: result.data[0].EMAIL_ADDRESS,
42         NAME: result.data[0].NAME,
43         ACCOUNT_NUMBER: result.data[0].ACCOUNT_NUMBER,
44         MESSAGE: 'Successfully retrieved account.'
45       });
46     } else {
47       res.status(500).json({
48         MESSAGE: `Failed to retrieve account with email address ${EMAIL_ADDRESS}.`
49       });
50     }
51   } catch (error) {
52     res.status(500).json({
53       MESSAGE: 'Internal server error.'
54     });
55   } finally {
56     await job.close();
57   }
58 };
```

SELECT * FROM CONNECTME.ACCOUNT
WHERE EMAIL_ADDRESS = ?

```
JS server.js X
Build a Watson AI Chatbot from Db2 > server > JS server.js > ...

71 app.put('/accounts', async (req, res) => {
72   // Extract and verify required body parameters
73   const { EMAIL_ADDRESS, NAME, ACCOUNT_NUMBER } = req.body;
74
75   let job;
76   try {
77     // Create SQL job
78     job = await createSQLJob();
79
80     // Execute query
81     const query = `UPDATE CONNECTME.ACCOUNT SET NAME = ?, ACCOUNT_NUMBER = ? WHERE EMAIL_ADDRESS = ?`;
82     const parameters = [NAME, ACCOUNT_NUMBER, EMAIL_ADDRESS];
83     const result = await job.execute(query, { parameters: parameters });
84
85     // Check result
86     if (result.success && result.update_count === 1) {
87       res.json({
88         MESSAGE: 'Successfully updated account'
89       });
90     } else {
91       res.status(500).json({
92         MESSAGE: `Failed to update account with email address ${EMAIL_ADDRESS}.`
93       });
94     }
95   } catch (error) {
96     res.status(500).json({
97       MESSAGE: 'Internal server error.'
98     });
99   } finally {
100     await job.close();
101   }
102 };
```

UPDATE CONNECTME.ACCOUNT SET NAME = ?,
ACCOUNT_NUMBER = ? WHERE EMAIL_ADDRESS = ?

Create Service Commander Definition to Start Server

Service Commander

A utility for managing services and applications running on IBM i

```
! connect-me.yaml ✕  
  
home > MAPEPIRE > connectme > server > ! connect-me.yaml  
1  name: ConnectMe Server  
2  dir: .  
3  start_cmd: npm run start  
4  check_alive: 3001
```

```
## Start server  
sc start connect-me.yaml
```

```
## Stop server  
sc stop connect-me.yaml
```

```
## Check server status  
sc check connect-me.yaml
```

Create OpenAPI Specification (in JSON format)

swagger.json

Build a Watson AI Chatbot from Db2 > server > swagger > {} swagger.json > {} paths > {} /accounts > {} put

```
1 {
2   "openapi": "3.0.3",
3   "info": {
4     "title": "ConnectMe Server API",
5     "description": "API for managing ConnectMe accounts.",
6     "version": "1.0.0"
7   },
8   "servers": [
9     {
10      "url": "http://{host}:3001",
11      "description": "IBM i",
12      "variables": {
13        "host": {
14          "default": "localhost",
15          "description": "Host name of IBM i"
16        }
17      }
18    }
19  ],
20  "tags": [
21    {
22      "name": "Account Services",
23      "description": "Account Services provide APIs to manage accounts."
24    }
25  ],
26  "paths": {
27    "/accounts": {
28      "get": {
29        "summary": "Get an account.",
30        "tags": [
31          "Account Services"
32        ],
33        "parameters": [
34          {
35            "name": "EMAIL_ADDRESS",
36            "description": "The email address associated with the account.",
37            "required": true,
38            "in": "query",
39            "schema": {
40              "type": "string"
41            }
42          }
43        ]
44      }
45    }
46  }
47 }
```

Swagger

swagger.yaml

Explore

ConnectMe Server API

swagger.yaml

API for managing ConnectMe accounts.

Servers

http://{host}:3001 - IBM i

Computed URL: http://localhost:3001

Server variables

host localhost

Account Services Account Services provide APIs to manage accounts.

GET /accounts Get an account.

PUT /accounts Update an account.

Test OpenAPI Specification

Curl

```
curl -X 'GET' \
  'http://ossbuild.rzkh.de:3001/accounts?EMAIL_ADDRESS=john.doe%40example.com' \
  -H 'accept: application/json'
```

Request URL

http://ossbuild.rzkh.de:3001/accounts?EMAIL_ADDRESS=john.doe%40example.com

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "EMAIL_ADDRESS": "john.doe@example.com", "NAME": "John Doe", "ACCOUNT_NUMBER": 1111, "MESSAGE": "Successfully retrieved account." }</pre></div><div><div></div><div>Download</div></div></div>

Curl

```
curl -X 'PUT' \
  'http://ossbuild.rzkh.de:3001/accounts' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "EMAIL_ADDRESS": "john.doe@example.com",
    "NAME": "Jonathan Doe",
    "ACCOUNT_NUMBER": 1111
  }'
```

Request URL

http://ossbuild.rzkh.de:3001/accounts

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "MESSAGE": "Successfully updated account" }</pre></div><div><div></div><div>Download</div></div></div>

Custom extension

☒ Get started ☒ Basic information ☐ Import OpenAPI ☐ Review extension

Basic information

Having a clear name and detailed description will help provide context and clarity to what your extension does.

Extension name

ConnectMe Server

Extension description

52/128

This extension integrates with the ConnectMe server.

Import OpenAPI Specification

Custom extension

Get started

Basic information

Import OpenAPI

Review extension

Import OpenAPI

Import an OpenAPI document in a .json format, describing the authentication and methods for your extension.

Drag and drop file here or click to upload

swagger.json

Review servers

Provided is a list of the servers and server variables found within the OpenAPI document.

URL	Description	Variables
http://{host}:3001	IBM i	host

Review operations

This table shows the operations defined in the OpenAPI document.

Operation	Method	Resource
Get an account.	GET	/accounts
Request parameters		Response properties
EMAIL_ADDRESS string Required		EMAIL_ADDRESS string
		NAME string
		ACCOUNT_NUMBER number
		MESSAGE string
Update an account.	PUT	/accounts
Request parameters		Response properties
EMAIL_ADDRESS string Required		MESSAGE string
NAME string Required		
ACCOUNT_NUMBER number Required		

Add Extension to Watsonx Assistant

Custom extension

Draft

✓ Get started

● Authentication

○ Review operations

Authentication

Authentication types are determined in the OpenAPI document and provide security for the extension.

Servers

http://{host}:3001

▼

i

Generated URL: http://ossbuild.rzkh.de:3001

Server variables

Enter values for your server variables to create a valid generated URL for requests.

host

ossbuild.rzkh.de

40

© Copyright IBM Corporation 2024

Create New Action with Initial Step

I want to view my account information

Customer starts with:
I want to view my account information

Conversation steps

1

I would be happy to help you view your account information. What is your email address?

Free text

Continue to next step

EditorVisualization

Step 1

Is takenwithout conditions

Set variable values

Assistant says

B I @ f_x


I would be happy to help you view your account information. What is your email address?

User enters free text


Edit responseEdit validation


Create Step with Extension

Step 2













Is taken

without conditions 


Set variable values 


Assistant says




</>

For example: What type of transfer would you like to make?



 Define customer response



Use an extension

✕

Extension setup

Choose an extension and operation. Then select the information to be shared with the external application to respond to your users' needs. [Learn more](#)

Extension ⓘ

ConnectMe Server

▼

Choose an extension that has been added to your assistant.

Operation ⓘ

Get an account.

▼

Choose from a list of operations included in your extension.

Privacy ⓘ

☐ Protect data returned from this extension

Parameters ⓘ

Set

Tr EMAIL_ADDRESS

To

1. I would be happy to help you view your account

▼

Cancel

Apply

Create Success and Failure Steps

Extension Success

Step 3

Is taken

with conditions

Set variable values

Conditions

1 condition

If All of this is true:

2 Ran successfully

==

true

X

and Add condition

New condition group

Assistant says

B I @ f_x

Your current account information:

Name: 2 body.NAME

Email Address: 2 body.EMAIL_ADDRESS

Account Number: 2 body.ACCOUNT_NUMBER

Define customer response

And then

End the action

Extension Failure

Step 4

Is taken

with conditions

Set variable values

Conditions

1 condition

If All of this is true:

2 Ran successfully

==

false

X

and Add condition

New condition group

Assistant says

B I @ f_x

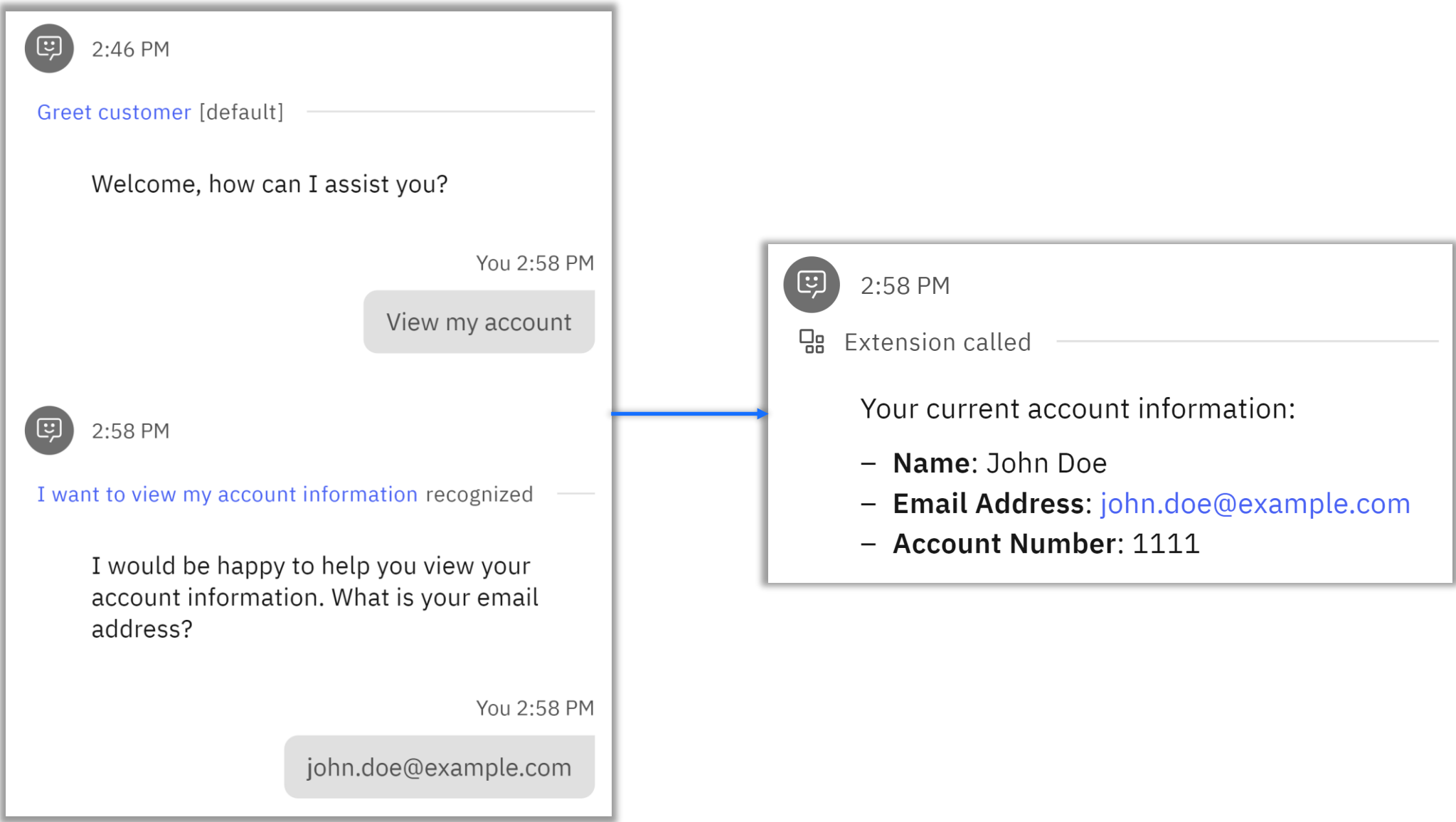
2 body.MESSAGE

Define customer response

And then

End the action

Preview Your Action



Duplicate Action and Update Phrase + Initial Step

Duplicate action

Everything in your action is duplicated except example phrases.

New action name

I want to update the name on my account

Cancel

Duplicate

I want to update the name on my account

EditorVisualization

Customer starts with:
I want to update the name on my account

Conversation steps

1

I would be happy to help you update your name. What is your email address?

Free text

Continue to next step

2

This step has no content

Use an extension

2 is true

Your current account information: - Name:

body.NAME - Email Address:...

Action complete

2 is false

body.MESSAGE

Action complete

Step 1

Is takenwithout conditions

Set variable valuesfx

Assistant says

B I π fx

I would be happy to help you update your name. What is your email address?

User enters free text

Edit responseEdit validation

And then

Continue to next step

Create Clarification Step

Step 3

Is taken

with conditions

Set variable values

Conditions

1 condition

If

All

of this is true:

2

Ran successfully

==

true

X

and

Add condition

New condition group

Assistant says

B I @ f_x Copy Paste Undo Redo Bold Italic

What is your updated name?

User enters free text

Edit response Edit validation

Create Step with Extension

Step 4

Is taken

with conditions

Set variable values

Conditions

1 condition

If All of this is true:

2 Ran successfully

==

true

X

and Add condition +

New condition group +

Assistant says

B I @ fX

🔗 📎 🎵 📧 📺 📻

</>

For example: Please select from the following options:

//

Define customer response

▼

Use an extension

Extension setup

Choose an extension and operation. Then select the information to be shared with the external application to respond to your users' needs. [Learn more](#)

Extension

ConnectMe Server

Choose an extension that has been added to your assistant.

Operation

Update an account.

Choose from a list of operations included in your extension.

Privacy

☐ Protect data returned from this extension

Parameters

Set

TNAME

To

3. What is your updated name?

Set

TEMAIL_ADDRESS

To

body.EMAIL_ADDRESS

Set

TACCOUNT_NUMBER

To

body.ACCOUNT_NUMBER

Cancel

Apply

Create Final Step

Step 5

Is taken

with conditions

Set variable values

Conditions

1 condition

If

All

of this is true:

2

Ran successfully

==

true

X

and

Add condition

New condition group

Assistant says

B I @ f_x

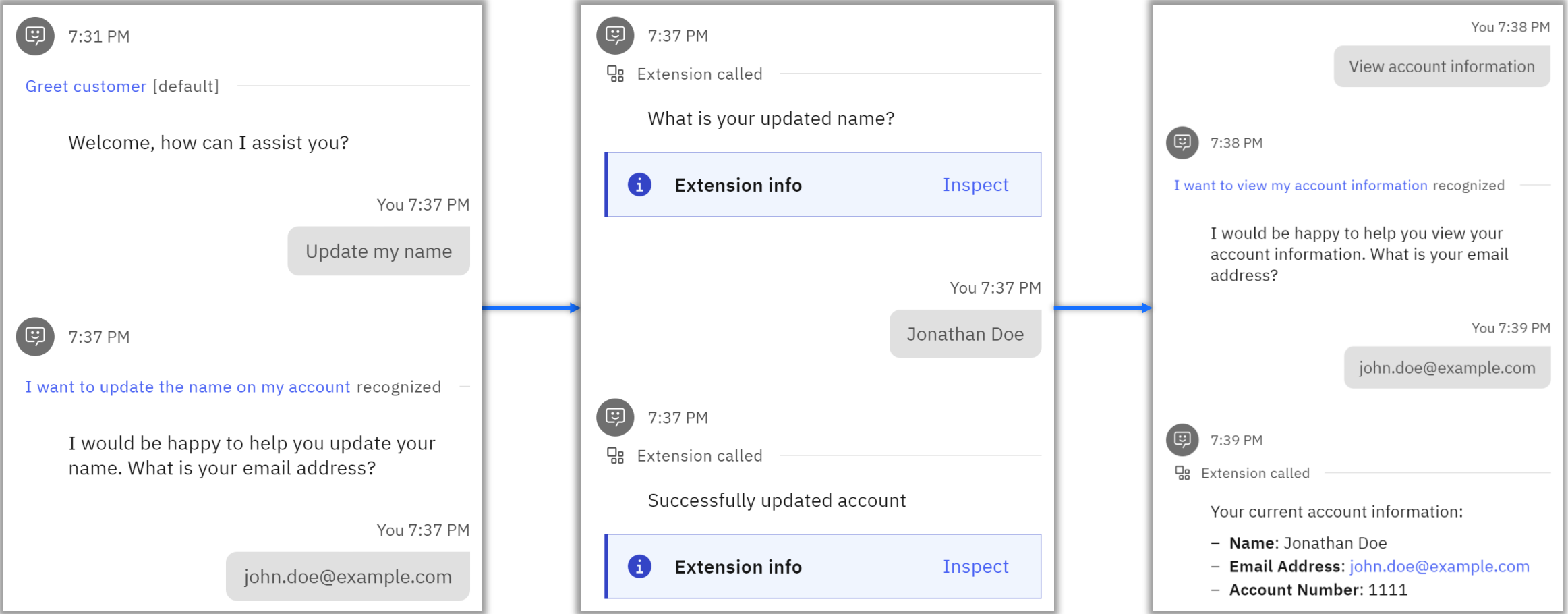
body.MESSAGE

Define customer response

And then

End the action

Preview Your Action



Next Steps

Add More Actions with Templates

Book a meeting

This template allows your customers to book a meeting, consultation, or appointment both quickly and easily. Your assistant will collect basic contact information, as well as set the date, time, and duration of the interaction.

Select this template ✓

What your customer says...

- Schedule a meeting
- Book a consultation
- Make an appointment
- I want to talk in person with someone about my case
- Set up an appt

What your assistant collects...

- Customer first name
- Customer last name
- Customer email address
- Meeting duration
- Meeting date
- Meeting time

Pre-built features

- Step conditions
- Options synonyms
- Customized response validation
- Action variables
- Regex response type
- Yes/No response type

Watson Assistant

Book a meeting

To set up a meeting, I'll need a few details. First, what's your name?

John Doe

Where would you like me to email the calendar invite?


john.doe@ibm.com


How long would you like to meet for?


15 minutes 30 minutes 1 hour


15 minutes


Adjust Default Actions














Actions

All items ^

Created by you

Set by assistant

Variables ^

Created by you

Set by assistant

Set by integration

Saved responses

Name	Last edited	Examples Count	Status
Greet customer	19 hours ago		✓
Trigger word detected	3 days ago		✓
No matches	an hour ago		✓
Fallback	3 days ago		✓

Adjust Global Settings

Clarifying Questions

Show options when multiple actions seem to match what the customer wants

Bank Bot

open an account

Did you mean:

- Open a new savings account
- Open a new checking account
- I want to apply for a mortgage loan
- None of the above

Change Conversation Topics

Switch to other actions when customers ask about other topics

Bank Bot

Can I make an appointment?

What time are you available?

Oh - what is the nearest branch location

Closest branch to you is 34 Front Street, Toronto, Ontario.

Do you want to continue making an appointment?

Yes No

Intelligent Information Gathering

The assistant retrieves and uses prior inputs to streamline the conversation

Bank Bot (feature ON)

Welcome, how can I assist you?

Can I report a defect about a problem?
On the home page, I click on My Account and get an error. My name is Cade and my email is cade@example.com.

Ok, I got it. Just to confirm:

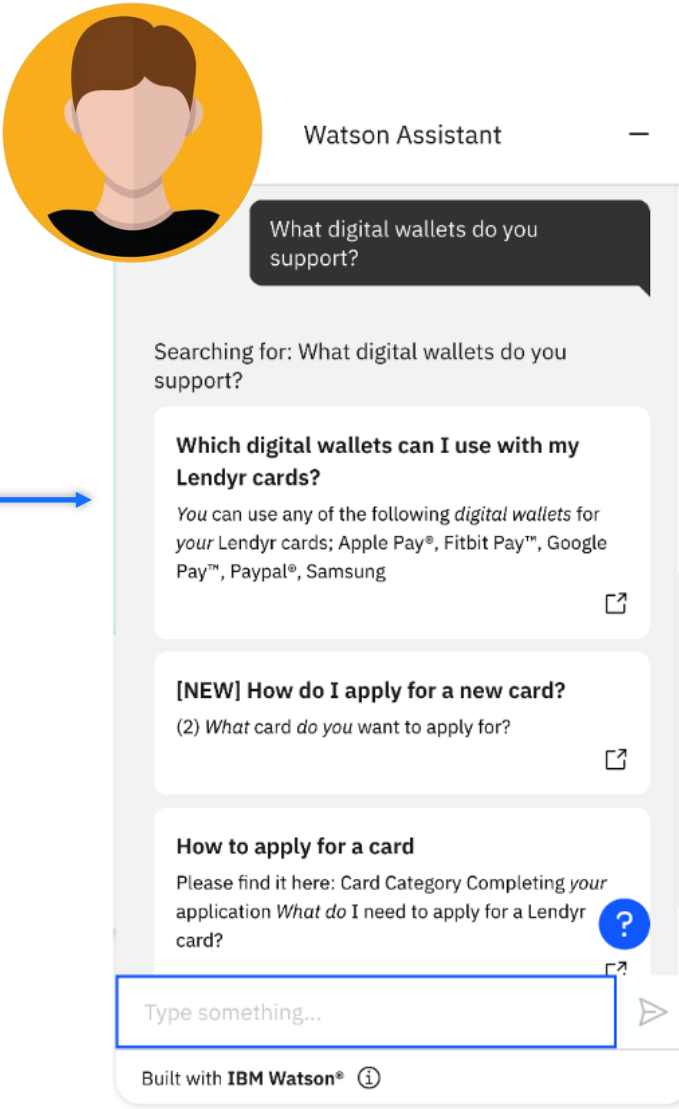
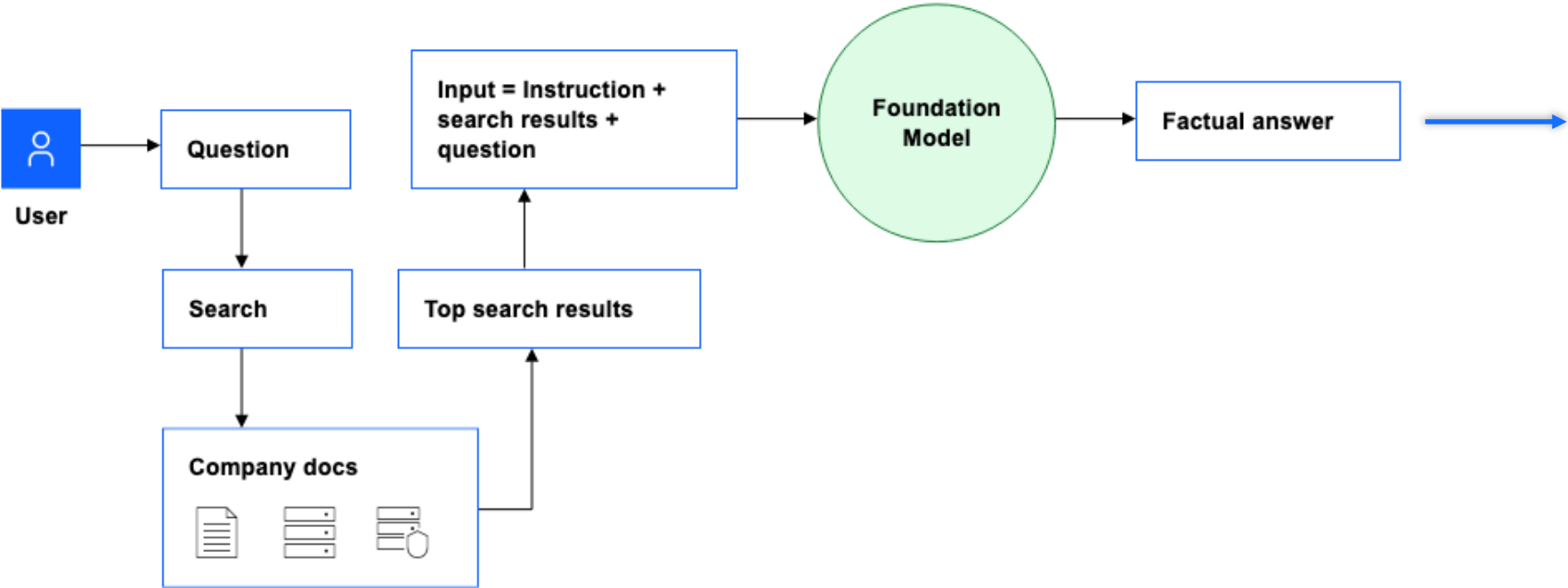
Defect is: **My Account produces an error**
Reported by: **Cade**
Email address is: **cade@example.com**
Steps to reproduce: **On the home page, click on My Account**

Do you want me to create a report with these details?

Search Integration with Watson Discovery

Search Integration

Extend the scope of what your assistant can answer by searching documents and websites.



01

Building a chatbot
with Watsonx
Assistant is easy!

02

Planning is critical
and building can be
an incremental
process.

03

Integrating your
chatbot with Db2 is
as simple as
creating a REST
server with
Maepire.

Any Questions?

IBM i