

Lab 3: Creating a Chatbot (1h)

Objective for Exercise:

- How to create a Simple Student Advisor Chatbot.
- How to add various intents and entities.

Pre-requisite:

We assume that you are already familiar with Watson Assistant. If you're not, then you can refer back to the previous reading documentation for more information or can consider taking Antonio's [course on building chatbots without programming](#) first.

In the next lab, we'll integrate it with the Watson Discovery collection we just defined. The chatbot, as mentioned before, will be rather barebone... just enough of a shell for you to build upon and show you how to integrate it with other services.

Create a chatbot, its intents and entities

1. Launch your Watson Assistance instance and create an assistant called Student Advisor Chatbot.
2. Add a dialog skill to the assistant (call it Student Advisor or something similar).

The screenshot shows the IBM Watson Assistant Lite interface. At the top, there is a navigation bar with the text "IBM Watson Assistant Lite" and "Upgrade". Below the navigation bar, there are two main sections: "Skills" and "Intents". The "Skills" section is currently active, indicated by a blue vertical bar on the left. The "Skills" heading is bolded. Below the heading, there is a descriptive text: "Skills contain the training to respond to your customer queries. Add skills to your assistant and then deploy to your channels." A large blue button labeled "Create skill" is centered below this text. In the main content area, there is a card for a skill named "My first skill" (Beta). The card displays the following details:

- TYPE:** Actions – English (US)
- CREATED:** Jun 8, 2021 3:19 PM IST
- UPDATED:** Jun 9, 2021 9:39 AM IST
- LINKED ASSISTANTS (1):** My first assistant

IBM Watson Assistant Lite Upgrade Learning center

Create a skill

Skills can be combined to improve your assistant's capabilities. [Learn more](#)

Actions skill Beta

Have an assistant ready to chat in less time. Compose step-by-step flows for any range of simple or complex conversations. Made so that anybody can build.

Dialog skill

Dialog offers all the smarts, power, and flexibility you've come to trust. Select to keep building with the tools you know and love.

Search skill Plus

Create Q&A experiences in minutes. Sync with websites and data sources for always up-to-date answers. Handle even complex questions with inclusive, contextual responses.

[Next](#)



IBM Watson Assistant Lite Upgrade

Create dialog skill

Create a new skill, start building a skill using the customer care sample, or import an existing skill.

[Create skill](#) [Use sample skill](#) [Upload skill](#)

Name

Name your skill; for example, Account application or Personal banking.

Description (optional)

Language ⓘ

[Create skill](#)

- From within that skill, add the *General* collection of intents in the *Content Catalog* by clicking on *Add to skill*.

The screenshot shows the 'Student Advisor' interface with the 'Content Catalog' tab selected. A message at the top says: 'Get started faster by adding existing intents from the content catalog. These intents are trained on questions that customers commonly ask.' Below is a table of categories:

Category	Description (optional)	Intents	Action
Banking	Basic transactions for a banking use case.	13	Add to skill
Bot Control	Functions that allow navigation within a conversation.	9	Add to skill
Customer Care	Understand and assist customers with information about themselves and your business.	18	Add to skill
eCommerce	Payment, billing, and basic management tasks for orders.	14	Add to skill
General	General conversation topics most users ask.	10	Add to skill
Insurance	Issues related to insurance policies and claims.	12	Add to skill
Mortgage	Get access to an entire mortgage AI system--including search integration, speech models, and more--for free: ibm.biz/mortgage	20	Add to skill
Telco	Questions and issues related to a user's telephony service, device, and plan.	21	Add to skill
Utilities	Help a user with utility emergencies and their utility service.	10	Add to skill

This will add some generally useful intents, including chitchat to the chatbot. In this lab we'll only leverage a few of them, but feel free to flesh them out as you build a more robust chatbot in the final project within module 6.

- Now that we have some chitchat intents, let's add some domain-specific intents to our dialog skill. Start by creating an *#Enrollment_Cost intent*, since this is a very common question for students. For the examples used to train Watson, feel free to add the following phrases:
 - Are your courses free?
 - Can I audit your courses for free?
 - Do I have to pay for every course in a specialization?
 - How much do your courses cost?
 - How much is a subscription?
- Create a *#Professional_Certificate_Recommendation intent* using the following examples:
 - Can you recommend a professional certificate to me?
 - Do you offer a deep learning certificate?
 - Which computer science certificate?
 - Which data science certificate?
 - Which data science certificate do you recommend?

Since there are only a few professional certificates being offered at the moment, we can hardcode our responses in a node of our dialog. But we want to provide specific answers, so that when people ask about the Deep Learning certificate, we provide them with a link to the right certificate. So we'll need an entity to identify that information.

6. Go ahead and create a `@professionalcertificate` entity with the following values and synonyms.

The screenshot shows the Watson Assistant interface for managing entities. The entity name is `@professionalcertificate`. The 'Value name' field contains the placeholder `Enter value`. There are two 'Synonyms' sections: one for the current value and another for adding new ones. A 'Dictionary' tab is selected, showing a list of five entity values with their types and synonyms:

Entity values (5) ▾	Type	Synonyms
<input type="checkbox"/> customer engagement	Synonyms	customer care, customer support
<input type="checkbox"/> data science	Synonyms	ibm, data analytics, data analysis, data engineering
<input type="checkbox"/> entrepreneurship	Synonyms	innovation management, entrepreneur, business development, mba
<input type="checkbox"/> IT support	Synonyms	technical support
<input type="checkbox"/> project management	Synonyms	

7. Finally, create a `#Course_Recommendation` intent with the following examples:
 - Do you have any courses on genetics?
 - I want to learn Python
 - Recommend me a course on nutrition
 - What courses do you recommend?
 - What's the best course on Machine Learning?
 - Which data science course should I take?

We'll use this intent in the next module when dynamically retrieving responses from the Watson Discovery collection.

Add nodes to the Dialog

With intents and entities defined, it is now time to create the dialog.

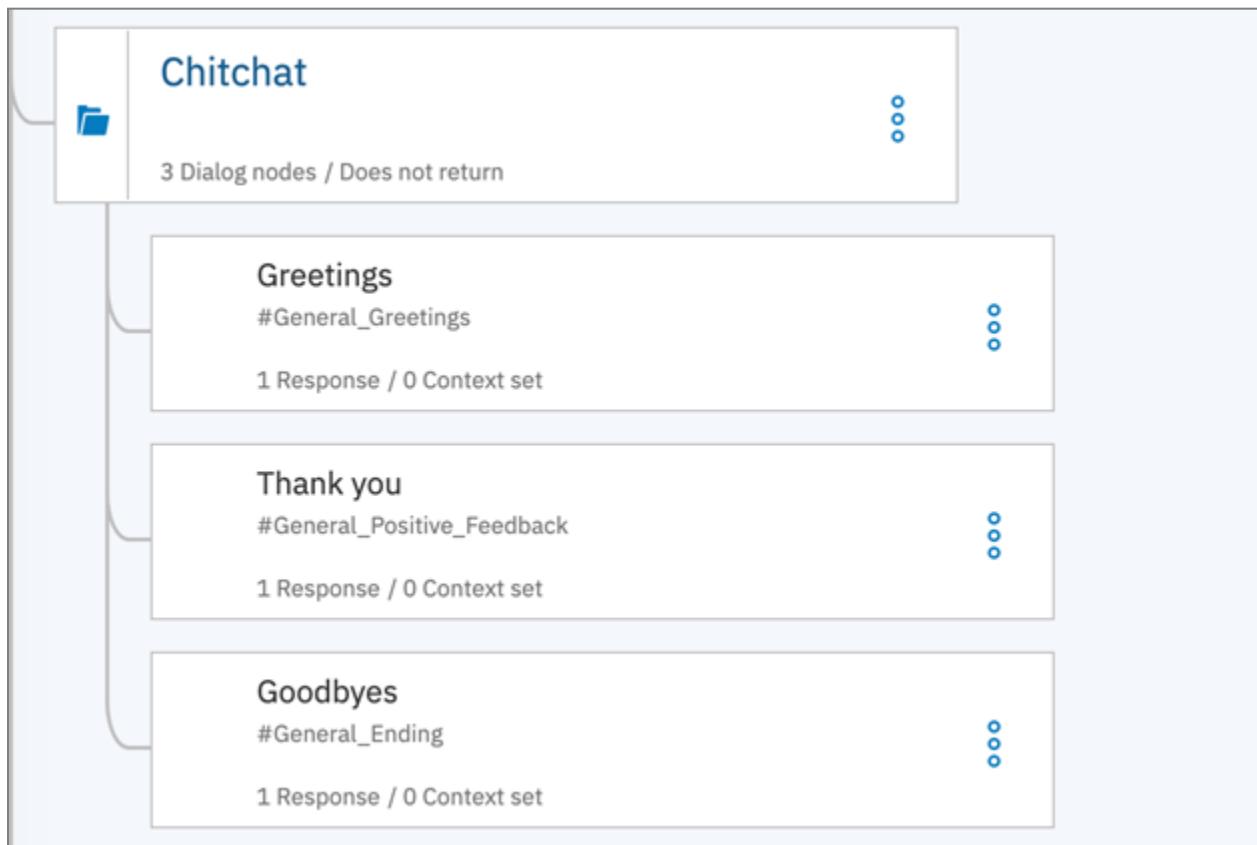
1. From within your skill, click on the *Dialog* tab and create the dialog.

2. Customize the *Welcome node* prompt with:

Hello. I'm a Student Advisor Chatbot. I can help you with your questions about our site and give you course recommendations.

Or something similar.

3. If you wish, customize the *Anything else* node responses.
4. Create a Chitchat folder, and add three nodes within it to handle greetings, thank yous, and goodbyes. You can leverage the General intents we added in the first part of this lab (i.e., #General_Greetings, #General_Positive_Feedback, and #General_Ending respectively). Define an appropriate response for each (e.g., *Hi there. How can I help you?* for Greetings.)



In module 6, for the project assignment, you'll want to flesh out this chitchat folder further to make the chatbot even more flexible and user friendly.

5. Above the Chitchat folder, we'll need to create three domain-specific nodes. Let's start with the first one. Add an *Enrollment Cost* node with the #Enrollment_Cost intent as its condition. For the response use following or something similar:

You can audit our courses for free, but in order to access graded assignments and a Course Certificate you'll need to pay. The cost varies from course to course. If you are aiming for a professional certificate, then you'll need to complete and pay for each course within that specialization.

The screenshot shows a chatbot configuration interface. At the top, there is a header "Enrollment Cost" with a "Customize" link and a blue "X" icon. Below this, a section titled "If assistant recognizes:" contains the condition "#Enrollment_Cost" followed by a minus sign (-) and a plus sign (+). A "Text" response type is selected, indicated by a dropdown menu. The response text is: "You can audit our courses for free, but in order to access graded assignments and a Course Certificate you'll need to pay. The cost varies from course to course. If you are aiming for a professional certificate, then you'll need to complete and pay for each course within that specialization." Below the response text, there is a placeholder "Enter response variation". A note at the bottom states: "Response variations are set to sequential. Set to random | multiline ⓘ". At the bottom left is a button "+ Add response type", and at the bottom right is a blue circular icon with a white speech bubble symbol.

6. Test the chatbot in the *Try it out* panel to ensure everything is working as expected so far.
7. Add a *Professional Certificate* node just below Enrollment Cost. For its condition, use `#Professional_Certificate_Recommendation`.
8. Enable multiple conditional responses from the *Customize link* within the node. Then add a condition and response for each entity value available, and the *true* fallback case, as shown in the table (and image) below.

Condition	Response

@professionalcertificate:(project management)	We recommend our Applied Project Management Certificate by University of California Irvine.
@professionalcertificate:(IT support)	We recommend Google's IT Support certificate.
@professionalcertificate:(customer engagement)	We recommend the IBM Customer Engagement Specialist professional certificate.
@professionalcertificate:(data science)	We recommend the IBM Data Science professional certificate.
@professionalcertificate:entrepreneurship	We recommend the professional certificate in Innovation Management and Entrepreneurship by HEC Paris.
true	You can see a list of our professional certificates on our site.

Professional Certificates

#Professional_Certificate_Recommendation - +

Then respond with:

If assistant recognizes	Respond with
1 @professionalcertificate:(project management)	We recommend our Applied Project Management Course
2 @professionalcertificate:(IT support)	We recommend Google's IT Support Course
3 @professionalcertificate:(customer experience)	We recommend the IBM Customer Experience Course
4 @professionalcertificate:(data science)	We recommend the IBM Data Science Course
5 @professionalcertificate:entrepreneurship	We recommend the professional certificate in entrepreneurship
6 true	You can see a list of our professional certificates

- Finally, add a Courses node with the condition `#Course_Recommendation`. Ignore the response section for now.

At this point you have a basic, but functional chatbot to assist online learners. In the next lab, we'll see how to connect it to our Watson Discovery collection to have our newly defined Courses node dynamically retrieve and issue responses.

Lab 4: Adding Discovery to the Chatbot: Part-2

Objective:

- How to use IBM Cloud Functions to make programmatic calls from dialog node.
- Learn about IBM functions.

Pre-requisite:

You should have completed the previous lab-Adding Discovery to the Chatbot:Part-1 successfully. If not, then please consider doing the previous lab first before moving further with this lab.

In the previous lab you defined an action. At the end of this lab, your chatbot will be able to call a cloud Function to receive a response from Watson Discovery and present course recommendations to the user.

1. Log in to [IBM Cloud](#) and then search for Functions in the search bar. Select Functions. After the page is loaded, make sure the top right hand side shows the name of your organization_lab4. If your organization is abcd@gmail.com, then you should see the following, <abcd@gmail.com_lab4>. Click Actions.

The screenshot shows the IBM Cloud Functions dashboard. The left sidebar is titled 'Functions' and includes sections for Getting Started, Overview, Pricing, Concepts, CLI, iOS SDK, Documentation, Actions, Triggers, APIs, Monitor, Logs, and Namespace Settings. The 'Overview' section is currently selected. The main content area is titled 'IBM Cloud Functions' and describes it as a Functions-as-a-Service (FaaS) platform based on Apache OpenWhisk. It features a large green 'f' icon with a brain-like pattern inside, surrounded by code snippets like '</>' and '< />'. Below the icon, there's a section titled 'What's New:' with a bulleted list of recent changes: 'IAM enablement', 'Namespaces can now be explicitly managed and show up on the dashboard', 'Manage Namespace Settings', 'Updated action runtimes', 'Increased maximum memory for actions to 2048MB', and 'Added support for monitoring the performance of your actions using metrics.' At the bottom, there's a section titled 'Save costs, scale and integrate.' with three small icons: a blue bell, a network of nodes, and a sunburst with an 'f'.

2. Click connectDiscovery action under lab4 package.

The screenshot shows the IBM Cloud Actions interface. In the top navigation bar, there are links for Catalog, Docs, Support, Manage, and thor iron's Account. On the left sidebar, under the Navigation Menu, the 'Actions' option is selected. The main content area displays a table titled 'Actions' with one item: 'connectDiscovery'. The table has columns for NAME, RUNTIME, WEB ACTION, MEMORY, and TIMEOUT. The 'NAME' column shows 'connectDiscovery', 'RUNTIME' shows 'Node.js 10', 'WEB ACTION' shows 'Not Enabled', 'MEMORY' shows '256 MB', and 'TIMEOUT' shows '60 s'. There is also a 'Create' button at the top right of the table.

3. Click Endpoints on the left sidebar.

The screenshot shows the detailed view of the 'connectDiscovery' action. The top navigation bar includes links for Catalog, Docs, Support, Manage, and thor iron's Account, along with a namespace dropdown (@gmail.com_lab4) and location (Dallas). The left sidebar has sections for Code, Parameters, Runtime, Endpoints (which is currently selected), Connected Triggers, Enclosing Sequences, and Logs. The main content area shows the 'Code' section with 'Node.js 10' runtime. It also contains a note: 'Currently Docker, Java (.jar), and Compressed (.zip) Actions are unable to be edited from the IBM Cloud Functions UI. These Actions can still be invoked.' At the bottom right of the code section, there are 'Change Input' and 'Invoke' buttons.

4. Click copy icon to copy the URL for the action and make a copy of the REST API URL. Next, click API-KEY under REST API.

- Click on the eye icon to obtain your CF-based API key and make a copy of the credentials.

You're now all set to begin integrating this function with Watson Assistant.

Integrate Discovery with our Chatbot

- From your Dashboard, click on the Watson Assistant service you created in Lab 3. (Click on the name, not the icon next to it.) and then click Launch Watson Assistant.

Name ▲	Group	Location	Status	Tags
Filter by name or IP address...		Filter by group or org	Filter...	Filter...
> Devices (0)				
> VPC Infrastructure (0)				
> Kubernetes Clusters (0)				
> Cloud Foundry Apps (1)				
> Cloud Foundry Services (0)				
Services (2)				
Discovery-na	Default	Dallas	Provisioned	--
Student Advisor	Default	Dallas	Provisioned	--
> Storage (1)				
> Cloud Foundry Enterprise Environments (0)				
> Apps (0)				

2. Under Skills tab, click Student Advisor skill. Then, click the Options tab.

The screenshot shows the IBM Watson Assistant interface. At the top, there's a navigation bar with 'IBM Watson Assistant' on the left and 'Cookie Preferences' on the right. Below the navigation bar, the URL 'Skills /' is visible, followed by a search icon, a 'Save new version' button, a 'Try it' button, and a more options menu. The main area is titled 'Student Advisor'. Below the title, there are tabs: 'Intents', 'Entities', 'Dialog' (which is highlighted with a blue box), 'Options', 'Analytics', 'Versions', and 'Content Catalog'. Under the 'Dialog' tab, there are three buttons: 'Add node' (highlighted with a blue box), 'Add child node', and 'Add folder'. The dialog tree itself starts with a node 'Student Advisor' which has four children: 'Welcome', 'Enrollment Cost', 'Professional Certificates', and 'Courses'. Each child node has a detailed view below it showing its responses and context sets. At the bottom right of the dialog tree area, there's a blue message icon.

3. Click Webhooks. In the URL field, specify the REST API URL for the action. Append a ?blocking=true parameter to the action URL to force a synchronous call to be made (e.g. https://eu-gb.functions.cloud.ibm.com/api/v1/namespaces/YOUR-ORG_YOUR-SPACE/actions/lab4/connectDiscovery?blocking=true). Finally, click Add authorization.

IBM Watson Assistant

Cookie Preferences ? ☰

Skills /

Student Advisor

Intents Entities Dialog Options Analytics Versions Content Catalog

Webhooks

Disambiguation Autocorrection System Entities

Webhooks

A webhook is a mechanism that allows you to call out to an external program based on events in your dialog.

Webhook setup

Specify the request URL for an external API you want to be able to invoke from dialog nodes. Watson will call this URL when configured to do so from a dialog node. [Learn more](#)

URL

`https://eu-gb.functions.cloud.ibm.com/api/v1/namespaces/` igmai

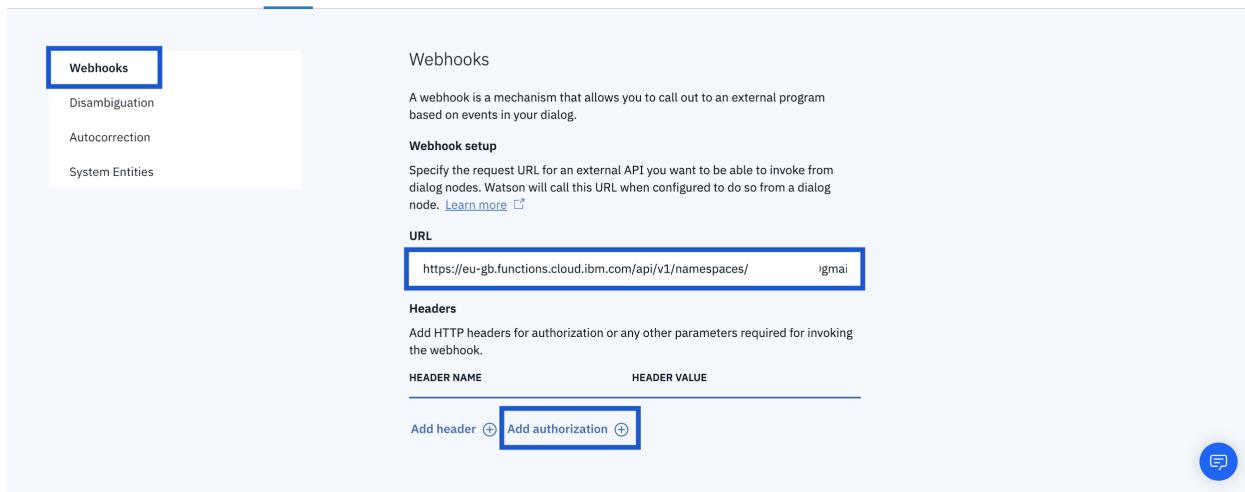
Headers

Add HTTP headers for authorization or any other parameters required for invoking the webhook.

HEADER NAME	HEADER VALUE
Add header +	Add authorization +

Next step

Save new version Try it



4. We'll use the CF-based API key you obtained from action endpoints. The part before the colon(:) is your User name. Copy this to the User name field. Similarly, the part after the colon(:) is your password. (e.g. Your action API key = User name:password). Finally save.

IBM Watson Assistant

Cookie Preferences ? ☰

Skills /

Student Advisor

Intents Entities Dialog Options Analytics

Disambiguation Autocorrection System Entities

Basic authorization

User name
Enter user name

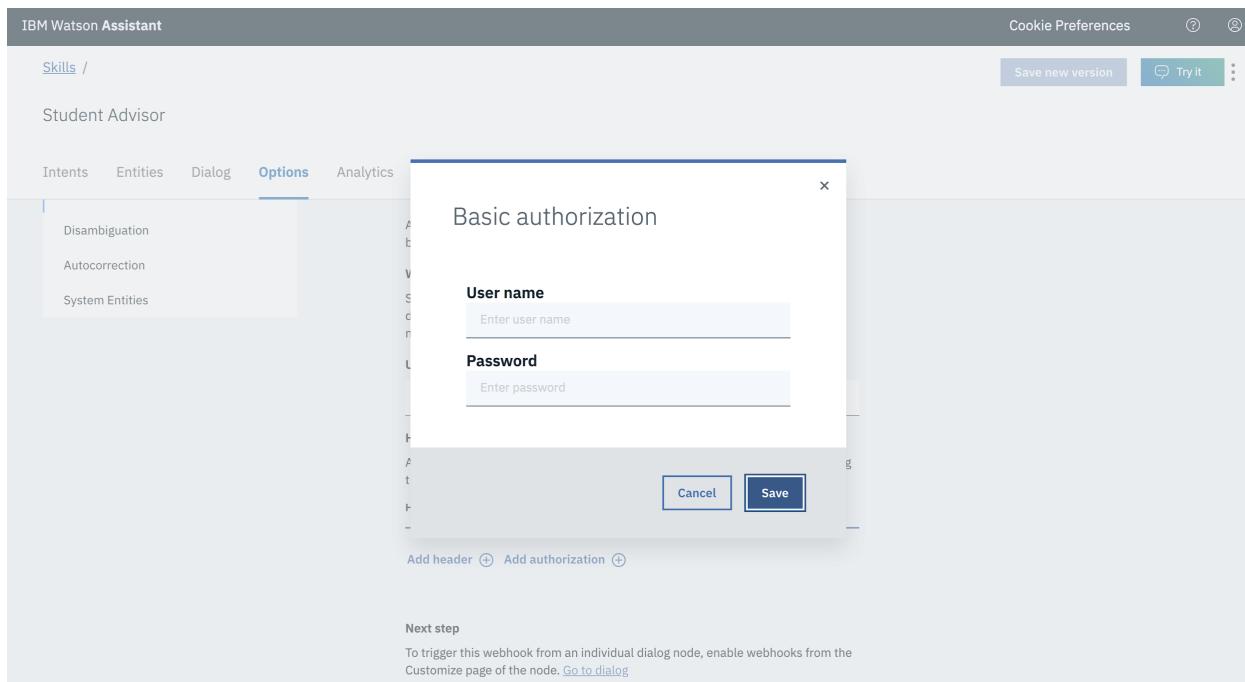
Password
Enter password

Cancel Save

Add header + Add authorization +

Next step

To trigger this webhook from an individual dialog node, enable webhooks from the Customize page of the node. [Go to dialog](#)



5. Find Courses node under the Dialog tab. Click the 3 dots icon in the Courses node and click Add child node. Then, name the new node (e.g., Discovery).

IBM Watson Assistant

Cookie Preferences ? @

Skills /

Student Advisor

Intents Entities Dialog Options Analytics Versions Content Catalog

Enrollment Cost
#Enrollment_Cost
1 Responses / 0 Context Set / Does not return

Professional Certificates
#Professional_Certificate_Recommendation
6 Responses / 0 Context Set / Does not return

Courses
#Course_Recommendation
1 Responses / 0 Context Set / Does not return

Chitchat
3 Dialog nodes / Does not return

Anything else
anything_else
1 Responses / 0 Context Set / Does not return

Add child node
Add node above
Add node below
Add folder
Move
Duplicate
Jump to
Delete

6. Click the Courses node and then click Customize to customize the node.

IBM Watson Assistant

Cookie Preferences ? @

Skills /

Student Advisor

Intents Entities Dialog Options Analytics Versions Content Catalog

Professional Certificates
#Professional_Certificate_Recommendation
6 Responses / 0 Context Set / Does not return

Courses
#Course_Recommendation
1 Responses / 0 Context Set / Does not return

Discovery
true
1 Responses / 0 Context Set / Return allowed

Chitchat
3 Dialog nodes / Does not return

Anything else
anything_else
1 Responses / 0 Context Set / Does not return

Courses

Customize ⚙️ ×

If assistant recognizes:

#Course_Recommendation ✖️ +

Then callout to my webhook: [Learn more](#)

Parameters

KEY	VALUE
Parameter name (ex. Date)	Parameter value (ex. \$date or \$sys-date)

Add parameter +

7. Scroll down to the Webhooks section, and switch the toggle to On, and then click Apply.

8. Now, we will add a parameter, input as a key and "`<?input.text?>`" as a value.

KEY	VALUE
input	#<?input.text?>

9. Scroll down and set the Return variable to `webhook_result_1`.

IBM Watson Assistant

Cookie Preferences ⚙️

Skills /

Student Advisor

Intents Entities Dialog Options Analytics Versions Content Catalog

Professional Certificates
#Professional_Certificate_Recommendation
6 Responses / 0 Context Set / Does not return

Courses
#Course_Recommendation
1 Responses / 0 Context Set / Does not return

Discovery
true
1 Responses / 0 Context Set / Return allowed

Chitchat
3 Dialog nodes / Does not return

Anything else
anything_else
1 Responses / 0 Context Set / Does not return

Courses

Customize ⚙️ X

Return variable
\$webhook_result_1

Webhook URL Your webhook URL is configured. [Options](#) X

Then respond with

IF ASSISTANT RECOGNIZES RESPOND WITH

1 anything_else Enter a response

Add response +

The screenshot shows the IBM Watson Assistant interface. On the left, a tree view displays several dialog nodes: 'Professional Certificates', 'Courses', 'Discovery', 'Chitchat', and 'Anything else'. The 'Courses' node is currently selected. On the right, a configuration panel for the 'Courses' node is open, showing a 'Return variable' section with '\$webhook_result_1' and a 'Webhook URL' section indicating it is configured. Below this, a 'Then respond with' section is shown, with a single response entry for 'anything_else'.

10. Scroll down and under And Then assistant should, select Jump to option and select the child node you created and then select If assistant recognizes(condition).

Course

Customize

Node name will be shown to customers for disambiguation so use something descriptive.

[Settings](#)

webhook_result_1

Assistant responds

If assistant recognizes

Respond with

1

anything_else

Enter a response



[Add response +](#)

Then assistant should

Choose whether you want your Assistant to continue, or wait for the customer to respond.

Wait for reply

Wait for reply

Skip user input

Jump to

Skills /

Student Advisor

Intents Entities Dialog Options Analytics Versions Content Catalog

Select a destination node (origin: Courses)

#Professional_Certificate_Recommendation
6 Responses / 0 Context Set / Does not return

Courses
#Course_Recommendation
1 Responses / 0 Context Set / Does not return

Discovery
true
1 Responses / 0 Context Set

Chitchat
3 Dialog nodes / Does not return

Anything else
anything_else
1 Responses / 0 Context Set / Does not return

Jump to and...
Wait for user input
If assistant recognizes (condition)
Respond

11. Put true as a condition for the node, as we always want to execute it when giving course recommendations. Add the following sample response, which retrieves a list of relevant courses from Discovery via the Function we defined earlier on.

```
Here are some courses I found for you! </br>
<a href='<? $webhook_result_1.response.result.courses.get(0).link ?>' target="_blank"><? $webhook_result_1.response.result.courses.get(0).name ?></a> <? $webhook_result_1.response.result.courses.get(0).description ?>
<br> <a href='<? $webhook_result_1.response.result.courses.get(1).link ?>' target="_blank"><?
$webhook_result_1.response.result.courses.get(1).name ?></a> <?
$webhook_result_1.response.result.courses.get(1).description ?> <br> <a href='<? $webhook_result_1.response.result.courses.get(2).link ?>' target="_blank"><? $webhook_result_1.response.result.courses.get(2).name ?></a> <? $webhook_result_1.response.result.courses.get(2).description ?>
```

The screenshot shows the IBM Watson Assistant interface. The top navigation bar includes 'Cookie Preferences', a search icon, 'Save new version', 'Try it', and a more options menu. Below the navigation is a breadcrumb trail: 'Skills / Student Advisor'. The main area has tabs for 'Intents', 'Entities', 'Dialog' (which is selected), 'Options', 'Analytics', 'Versions', and 'Content Catalog'. On the left, a tree view of dialog nodes is shown:

- Professional Certificates**: #Professional_Certificate_Recommendation, 6 Responses / 0 Context Set / Does not return
- Courses**: #Course_Recommendation, 1 Responses / 0 Context Set / Jump to / Does not return
 - Jump to Discovery (Evaluate condition)**
 - Discovery**: true, 1 Responses / 0 Context Set / Return allowed
- Chitchat**: 3 Dialog nodes / Does not return
- Anything else**: anything_else, 1 Responses / 0 Context Set / Does not return

The right side shows the configuration for the 'Discovery' node. It has a 'Discovery' title and a 'Customize' button. Under 'If assistant recognizes:', there is a condition 'true' with an 'X' icon and a '+' icon. Under 'Then respond with', the type is set to 'Text' with a dropdown arrow. The response template contains the following HTML code:

```

Here are some courses I found for you! <br>
<a href='?> $webhook_result_1.response.result.courses.get(0).link ?>' target='_blank'><?
$webhook_result_1.response.result.courses.get(0).name ></a> <?
$webhook_result_1.response.result.courses.get(0).description > <br> <a href='?>
$webhook_result_1.response.result.courses.get(1).link ?>' target='_blank'><?
$webhook_result_1.response.result.courses.get(1).name ></a> <?
$webhook_result_1.response.result.courses.get(1).description > <br> <a href='?>
$webhook_result_1.response.result.courses.get(2).link ?>' target='_blank'><?
$webhook_result_1.response.result.courses.get(2).name ></a> <?
$webhook_result_1.response.result.courses.get(2).description >

```

12. Now you're all set. Try running Recommend me a course on databases in Try it out panel to confirm that the node works as expected. If you see a result similar to the image below, you are all set for this lab.

To recap, we created a Cloud Function that connects to our Discovery collection and retrieves documents relevant to the user query. We then invoke that function from the relevant node, when people express the intent of receiving a course recommendation. The result is then displayed to the user in our response.

The screenshot shows the IBM Watson Assistant interface. On the left, there's a navigation bar with 'Skills /' and tabs for 'Intents', 'Entities', 'Dialog', 'Options', 'Analytics', 'Versions', and 'Content Catalog'. Below the navigation is a tree view of dialog nodes:

- Professional Certificates**: #Professional_Certificate_Recommendation, 6 Responses / 0 Context Set / Does not return.
- Courses**: #Course_Recommendation, 1 Responses / 0 Context Set / Jump to / Does not return. This node has a sub-node labeled "Jump to Discovery (Evaluate condition)" which points to the "Discovery" node.
- Discovery**: true, 1 Responses / 0 Context Set / Return allowed.
- Chitchat**: 3 Dialog nodes / Does not return.
- Anything else**: anything_else, 1 Responses / 0 Context Set / Does not return.

On the right, there's a "Try it out" panel with a search bar, a "Save new version" button, and a "Cookie Preferences" section. The panel displays a conversation transcript:

```

Hello. I'm a Student Advisor Chatbot. I can help you with your questions about our site and give you course recommendations.

recommend me a course on database
#Course_Recommendation

Here are some courses I found for you!
Distributed Database Systems The increased capabilities of a collection of logically inter-related databases distributed over a computer network enable scalable data processing
NoSQL Database Systems Unlike traditional relational database management systems, NoSQL databases are capable of storing unstructured data
Databases and SQL for Data Science Much of the world's data resides in databases

Enter something to test your virtual assistant
Use the up key for most recent

```

In short, we managed to dynamically invoke results rather than hardcoding the responses. This is a very powerful and flexible approach. You'll use it again when working on the capstone project.