User Guide

CM.com - Chat bot user manual

September 2020

Version 0.1

By: Saskia Beukeveld and Derk-Jan Hartman

Contents

Introduction

Good to know: setting up a Chatbot the first time might not be simple. Since you don't have experience yet on how consumers respond. Do you want advice? We can sure help you. We have brought many chat-bots to life and the scenarios and experience will help you make a swift start. When this interests you, please book a demo and we can take a look how, together, we will make you successful. To try out the capabilities of the bot, please scan this QR-code and meet our WhatsApp Coffee-flow bot !	

You might be wondering; do I need a bot with artificial Intelligence or a scripted bot:

- AI: Artificial Intelligence bots with Natural Language Processing are offered by many communication companies and can be a wonderful interaction with your customers. However, there are also some limitations. Most of these systems only support a very select few languages. They also tend to require a lot of input to train them before you can really start using them and require continuous assistance in training even after it is in use. This can require a significant budget.
- Scripted bot: are great if you want to have assistance with repetitive tasks. This is a simple technology many people are familiar with from Voice call centers, but with powerful additions. It works in any language, is simple to configure and much cheaper.

This document is all about setting up a CM-scripted bot. If you believe an AI bot is more relevant for you, we can help you with that too. Please visit CX.com for more information (CX, is a fully owned entity of CM.com).

About the CM-Chatbot

A scripted conversation to get specific information. The user must answer a series of questions for the bot to get all the required information. Handover to support agent when the right need has been identified. Flows with a path to a defined goal.

- It can handle a strict script of sending replies
- Can send and receive messages, images, video's and locations
- Menu options like 1, 2, 3, questions and answers flows (like we know them from call-centers, to navigate you to the right team or search capabilities).
- Hand over the conversation to a human agent in CM-Customer Contact
- Make API calls and trigger webhooks.

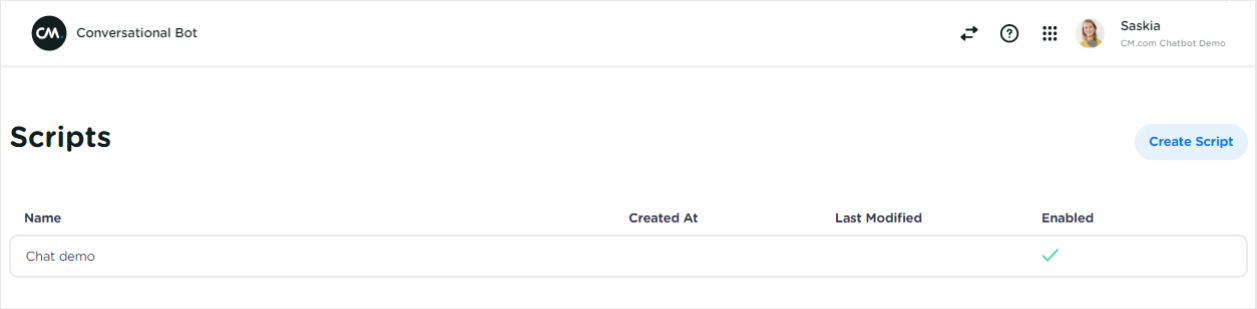
What the bot is not

- NLP/AI/self-learning
- Simple. The bot allows for quite some complexity, but it doesn't require it.

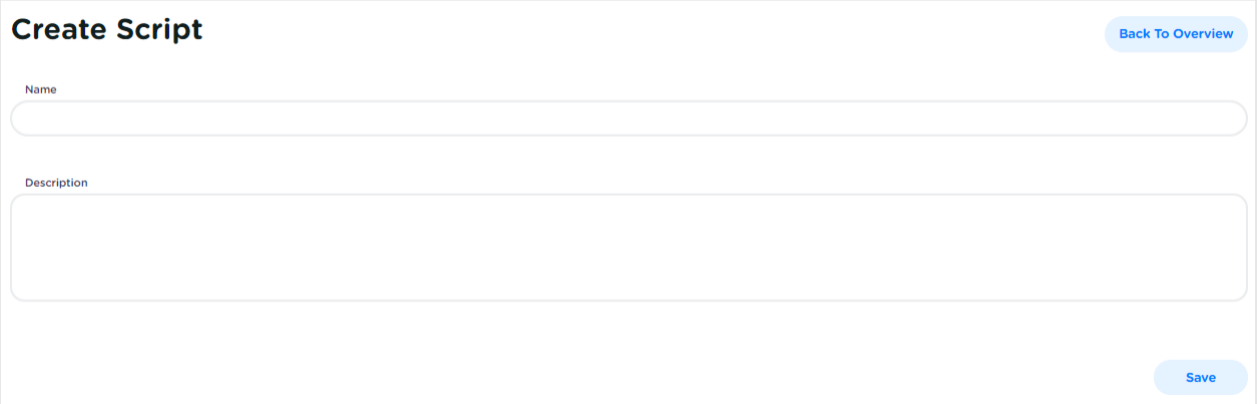
CM.com has a wide range of use cases live: status updates, delivery notifications, information gathering, team routing, making a reservation. If you need assistance with building your building your flows please contact your representative.

Getting Started

After applying for a bot, you will receive a link from CM.com, which will allow you to create and edit bot scripts.



Once you have clicked on Create Script, a new script is added. Fill out the Name (mandatory) and description. The name and description are for administrative purposes and have no impact on your bot.



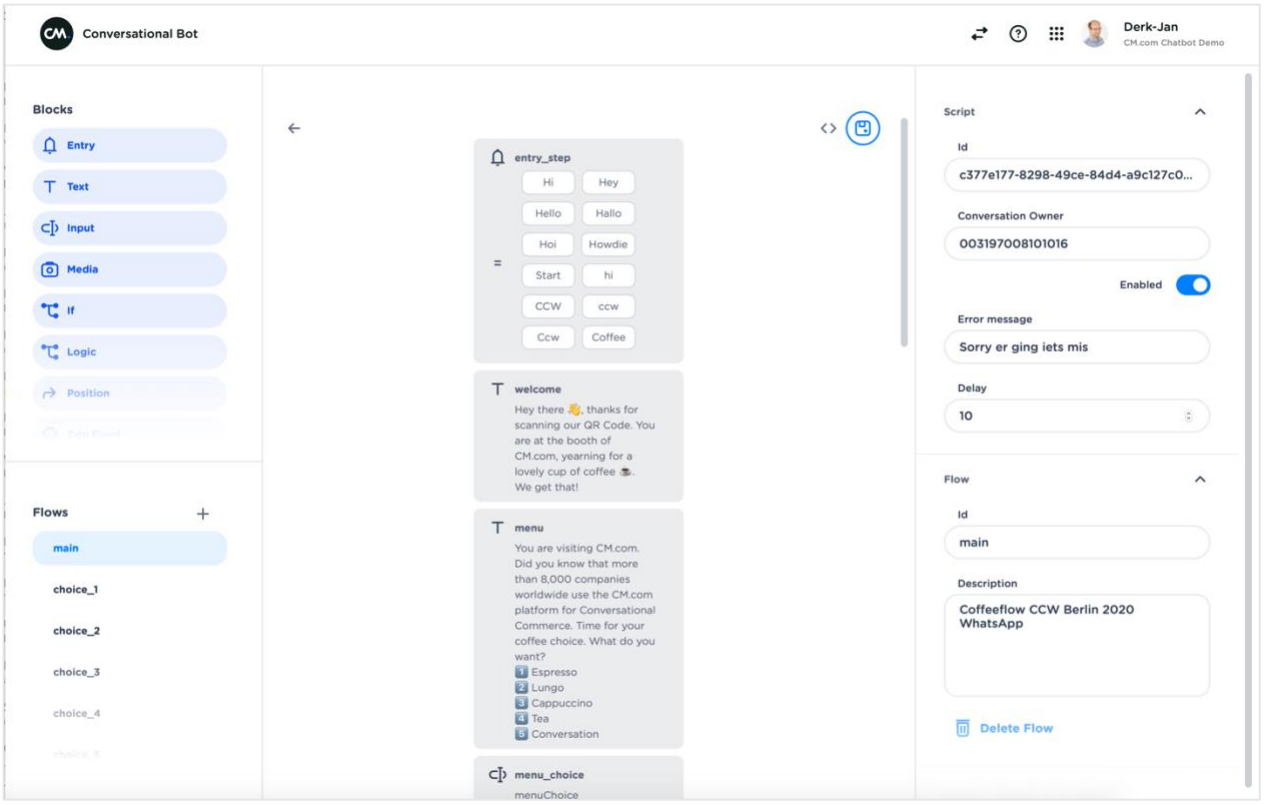
Multiple scripts

When you are using chatbots for multiple purposes in your organization, it is useful to create a script per area. This makes it easier to add ownership, implement flows and to make changes at a later stage.

Advice: the variety of features per communication can vary widely. SMS only supports text, RCS takes care of combined messages (i.e.. Text and images in one message) and Apple Business Chat offers features like Apple Pay.

When you offer customer communication via two or more communication channels it can be confusing when you add those to one chat-bot script. We strongly advice to use one script per communication channel, even when they support the same process (like looking up an order or asking for initial information). With help of the duplication feature, you can quickly copy and adjust flows and maintain the channel unique features. For CM.com software like customer contact this is not at all an issue. No need to (extensively) train your agents regarding this topic.

Navigating the bot builder interface



The interface consists of the several important elements.

On the left you find “Blocks”. Blocks are the building blocks for your bot interactions. You can drag a block into the “Flow builder”, which is the central area of the screen. At bottom left, you can manage your “Flows” and if you select a flow here, it will display in the Flow builder in the center.

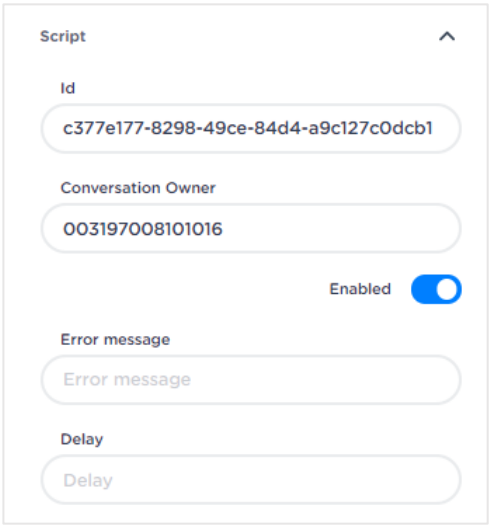
At the top of the Flow builder, you will find a back arrow, which will take you back to the Scripts overview. The <> symbol allows you to access the underlying YAML representation of the scripts, which is useful for advanced users of the bot builder software. Next to that is the Save button. Make sure to regularly save your scripts as you edit them.

Finally, on the right side you have the configuration section, which shows the configuration options of the selected context, generally the selected flow or the selected block inside the flow.

Configuring a script

After opening a script, you will notice a configuration section on the right side of the interface, where you define principles which apply to the entire script.

You can modify these parameters at any time, and they will take effect immediately after saving. If you need to return to this section, simply select any flow in the Flows section to make this configuration section appear.



Script ID	This is an identifier for the script. You generally do not have to modify or use this ID.
Conversation Owner	Depending on the communication channel this is the sender of the conversation. • SMS: MSISDN (a phone number that is enabled by operator(s) to receive inbound messages on) • WhatsApp: Your Business Profile WhatsApp phone number • Apple Business Chat: Apple Business Chat ID
Enabled	A quick toggle to disable and/or enable the entire script and the bot actions it will take.
Default Delay	This specifies the number of milliseconds to wait between the steps in your flow. When you test your flow, or when you receive the feedback it runs too fast, increase it. A very low (thus fast) number could have the side-effect of messages not being in order on the end-user's side. Our advice is to have a minimum of 250 milliseconds. It allows the consumer to read the message before the next message arrives.
Default Error Message	This message will be sent to the consumer in case of a fatal error during processing of the script.

Note: When you have created a flow in your test environment and copied it to the script for production ensure that you update the Conversation Owner.

Build your script flows

Your bot script is likely divided into a collection of understandable flows. Each flow consists of a collection of steps of different types, your building blocks.

Sequence of elements: Your account >> Script >> Flows >> Steps

Each step will have the option to add a **description.** This field is used by the script designer and allows to add a small note for the script designer. It will not be visible to end-users.

CM.com offers the following Step-kinds. In the following chapters each of them will be explained thoroughly.

Interpretation of incoming messages;

1. Entry Step
2. Input Step

Actions and sending message;

3. Text Step
4. Media Step
5. Location Step
6. Logic Step
7. Position Step
8. Set variable Step
9. Terminate

Interact with External Systems;

10. Hand over step
11. Web request
12. Webhook

You can use a variety of different steps in your flows for handling different message types, script flow and other logic.

Generic properties for each type of Step:

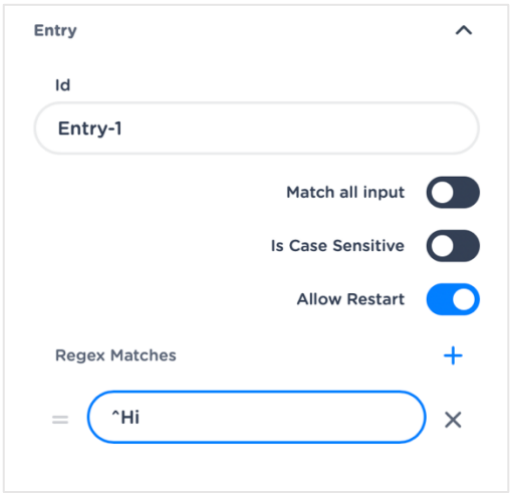
- **id**: A value that uniquely recognizes that step within the script. This is used when you want to connect flows with each other, and have a flow start at a specific step.

Incoming Messages

Depending on how customers contact you, you will receive incoming messages. Incoming messages can either start off a new Flow or be used in between replies of an existing Flow.

Entry Step

Our Chatbot needs to be triggered by consumers messages to start. What those triggers are is defined at a so-called **Entry Step**. In this step you need to define which words or sequences will start this flow. A flow should have only one Entry Step and it should be the first step in a Flow.



You can have multiple flows with an entry step, and the order of the definitions of the flow determines the order in which the script will try to match the Entry steps and thus the starting point of the script. To match a possible starting sequence, one or multiple regular expressions can be configured which each allow the Entry step to start the script (so ANY of the regular expressions could match / concatenated with OR).

Lessons Learned

- Please think about how you welcome your customer. People are looking for a personable and welcoming experience, so your bot needs to show its human face.
- You might want to mention a 'escape' keyword in your welcome. Having a separate flow with an Entry which responds to the word 'Help' or 'Stop' to direct people to a human agent or an email address is good form.
- This element needs training, when the bot is live you will likely continuously add new matches in this step as you learn what is that your customers are looking for

The CM.com coffee flow starts with "hi". Although when looking at our data, we also learned that we want to start the same flow with other words. Hence we added to our example: coffee, COFFEE, hey, hello, hallo.

When possible for the communication channel, add the "entry" words in your start message of the link. Example: https://wa.me/3197008101016?text=Coffee

Properties of the Entry step:

Match all input: Set this to make sure that any input by the consumer which does not match any other Entry step, will start this flow. Each script should probably have one flow with an Entry step that has this option set.

Is Case Sensitive: Toggle this option to make sure that even the casing of the matching needs to match for this Entry Step. Defaults to false.

Allow Restart: Allows the script to restart the script in a new session in case a previous session was already present. Defaults to true.

Regex Matches: One or more regular expressions to match. Regular expressions require some technical skill but are very powerful and allow for more advanced matching logic. Named capture groups are allowed and will be available as session variables on a successful match.

Using regular expressions

Regular expressions allow for powerful string matching that is too rich featured to detail here. There are many online resources allowing you to learn about regular expressions and MDN is one of them. Some useful simple regex examples are:

- `^Hi`: Input starts with Hi
- `Hi$`: Ends with Hi
- `^Hi$`: Begins and ends with Hi
- `(Hi|Hoi)`: Hi OR Hoi anywhere in the input

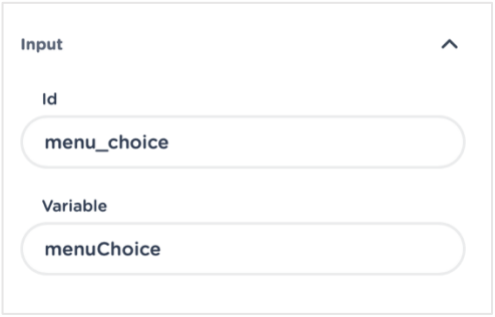
Using regular expressions in the Entry Step opens the possibility to use so-called named capture groups: a specific pattern within the matched sequence can be extracted or 'captured' and will be stored in a session variable that will have the same name as the capture group.

This will best be explained using an example: `^Order (?<orderCode>\d{5})$`

In the above example, all sequences starting with "Order", followed by a space and exactly 5 digits will be matched. The named capture group is specified between brackets `(?<orderCode>\d{5})` and the name of the capture group is "orderCode". As a result, when the end-user starts the chat with the text "Order 12345", a new session will be started and a session variable 'orderCode' with value '12345' will exist within the session.

Input Step

After sending a message to your customer, when you expect an answer in return, you need add the Input Step to your flow. This step will wait for a response by the customer before continuing with its flow.



We can receive following types of input:

- Text
- Media
- Location

This received input is stored in the named variable you specify for this Input step. You can then reply with a new message or take an action. Input steps are often followed by a Logic step to process the value of this variable and to either start new flows or by something like a Web request.

Coffee flow:

What do you want to drink

1. Lungo
2. Cappuccino
3. Tea

User enters { 2. }

Based on this input you will start a flow specific for Cappuccino

Variables by input

When using an InputStep the literal value that is sent by the end-user will be stored in the variable name as specified in the configuration of the Input step.

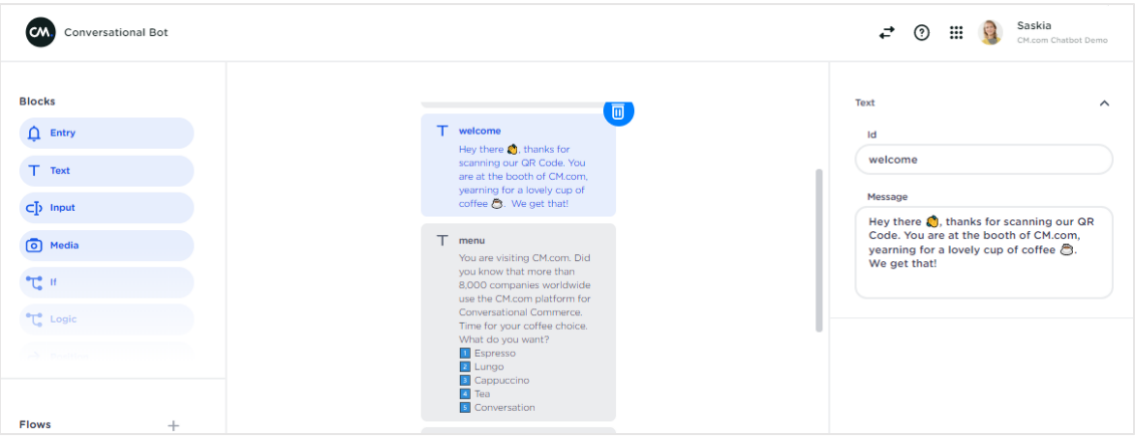
Media and Locations

It is also possible to capture media and location items. By default, these will be converted to text reflecting this input. It is possible to retrieve the full details of these elements, but this is not yet exposed in the interface of the Bot builder. Please contact your CM contact for support in configuring these for your flow.

Actions and sending messages

Text Step

When you want to send a message to the consumer with plain text, you use the Text block.



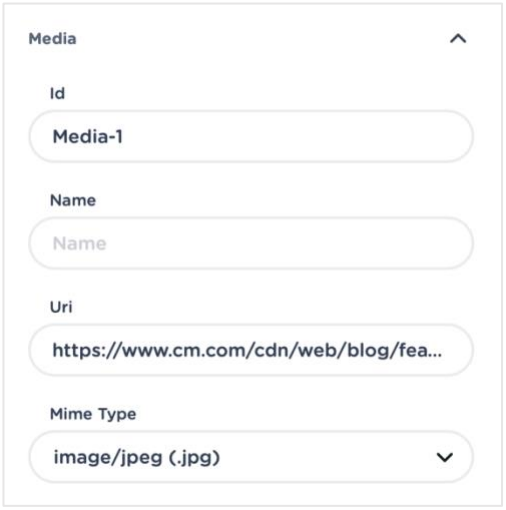
The following applies:

- You can use Unicode emoticons, like the text 'Hey there 👋!'
- It works for any Unicode language (like Arabic, English and Chinese)
- You can create new lines with the use of enter
- You can have as many Text blocks as you want
- Include URLs as you wish to link to other pages/content
- You can use variables inside messages. To learn more about variables, see the chapter on variables.

Advice: to make messages readable on a mobile phone, we suggest sending relatively small messages, so your customer does not have to scroll. Rather use additional, new, text blocks.

Media Step

Since you are using these great communication channels with all their rich features, you will want to share a picture, movie or sound. It is possible when you have selected the right item in the Chat-bot flow, pull in the **media step.**



When you want to send a media file, make sure you configure a LINK to a public location where the media item is stored. These links will then be displayed as 'preview' in the chat of the consumer and can be downloaded.

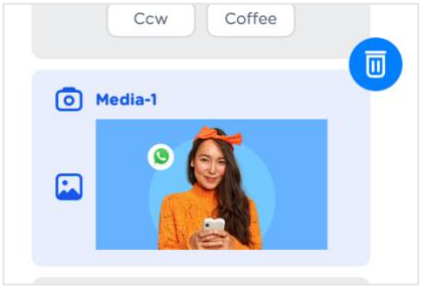
You should also name the image and select the appropriate filetype.

Note: it is not (yet) possible to combine images with text in one message. Since the Chat-apps don't support this yet for businesses (June 2020)

Every type of channel has different limitations upon which exact media type you can use, and maximum file sizes do apply.

Common types to share are:

- image/jpeg
- image/png
- audio/mpeg3
- video/mpeg4
- image/gif (animated gifs and stickers are **not** supported at Business Messaging/WhatsApp at the moment, June 2020)



Location Step

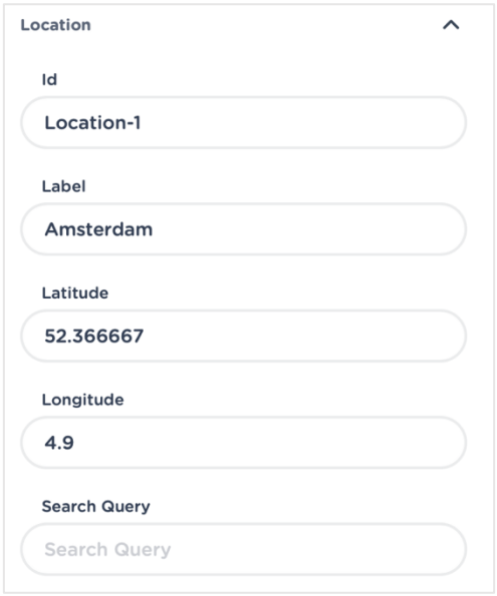
Maybe you want to inform your customer of your store location where they can pick up their product. It is possible to send them a message with the coordinates of that location.

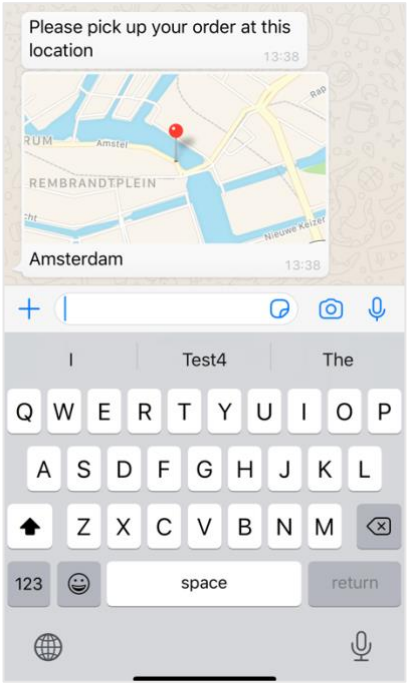
Properties of the Location step:

Label: A textual description of this location. This information will be shown below the map bubble in the application of the consumer.

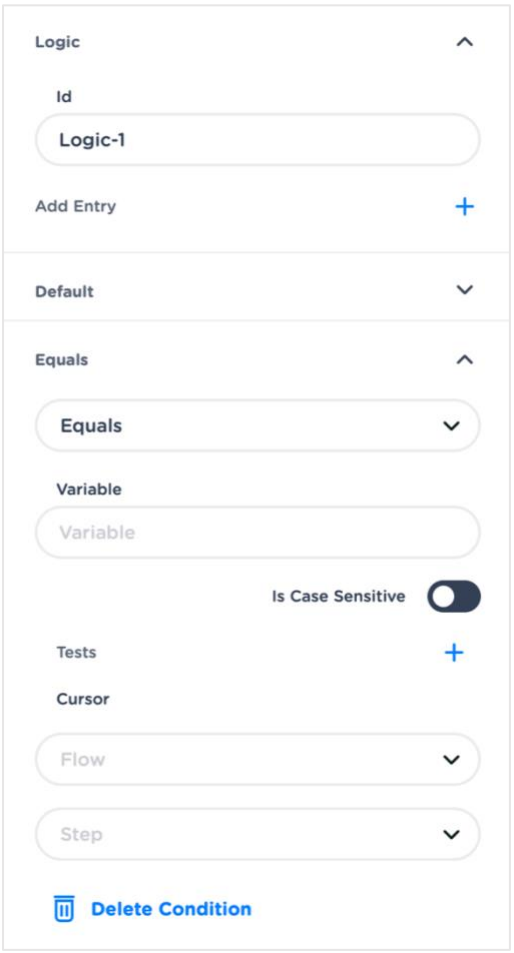
Latitude: Latitude degrees of the coordinate. Specified in decimal format. For instance: 52.366667

Longitude: Longitude degrees of the coordinate. Specified in decimal format. For instance: 4.9





Logic Step



Use the Logic step to evaluate conditions; for instance the value of the input you received from a customer.

The result of a Logic step is a Cursor action. With a Cursor action you instruct the bot to go to a 'different position' within the botscript. For details about Cursor actions, see the chapter on the Position step.

Reasons to use the logic step can be:

1. You want to split the flow because of input by the user. For instance in a menu where you have multiple options, 1, 2, 3, 4 etc.
2. Validate the input provided by the user. For instance to check if something really is an email address.
3. To evaluate the output returned by a previous HTTP request step.

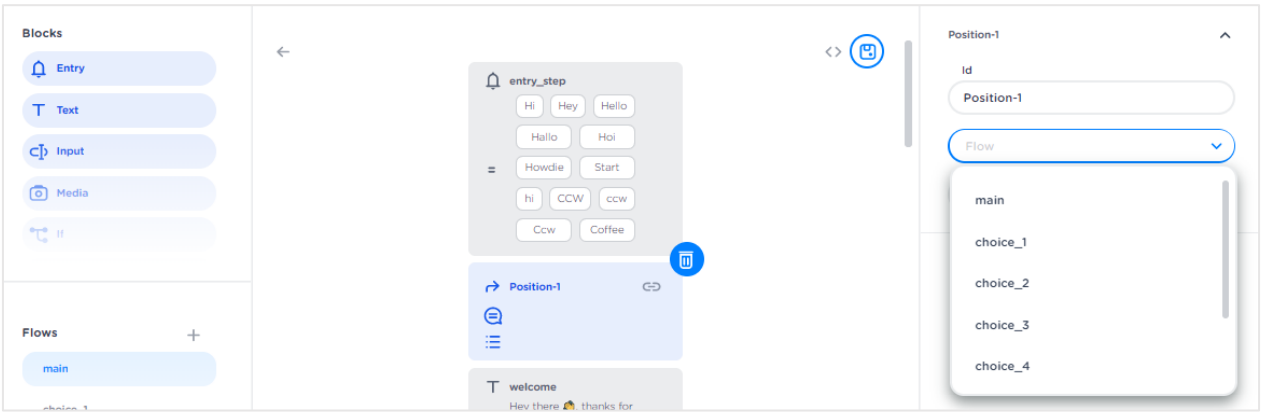
Logic knows the following types of conditions and you can use multiple. They are evaluated in the order added:

1. Default: executed in case none of the other conditions in the Logic step are matched. You probably should have one Default for each Logic step.
2. Equals: check if a variable exactly matches your conditions.
3. Starts With: does the variable begin with a certain value. (for instance all entries starting with "Fin" will match the need to speak to someone from your Finance department – avoidance of typos).
4. Regex: Provide one or more regex patterns which to match.
5. And/Or/Not: Combine multiple sub conditions with AND, OR and NOT operators.
6. Email Validator: A ready-made validator for email addresses.

Position Step

The position block can be used to switch to a different point in a Flow or another flow. This action is useful if you have complex flows and you want to reuse parts of a flow. Simply put the reusable parts in a separate flow and add a Position step to jump you to the reusable part.

This action is also used inside the Conditions of the Logic step to instruct the bot on where to go next.



A position step or action has two basic parameters:

Flow Id: Which flow to go to next

Step Id: Which step within that Flow to go to.

The values used as IDs here are the values of the ID which the UI starts off each step and flow with. Combined as a position these are called the **Cursor** inside the script.

Set Variable Step

This step is an easy way to introduce an extra variable and give it a value.

To learn more about variables, read the chapter on Variables.

Terminate Step

For the bot it is important, to know when a flow ended. We have added the Terminate step to each flow. When the consumer reaches this point in the flow, the terminate step informs the system the conversation ended. New flows of course can be started, while others have been terminated, but they start fresh.

You can remove the Terminate step when needed, but be careful that you are certain that the flow is ended at later point.

Interact with External Systems

Hand over step

With the implementation of a chatbot you want to improve the workflow of your agents, reduce the repetitive tasks or alike. Obviously, the bot does not know it all. When this is the case, your human agents are there to help.

Answering conversations by human-agents is done in CM.com's Customer Contact software. The ultimate inbox for all communication channels, including the flows pre-worked by your bot.

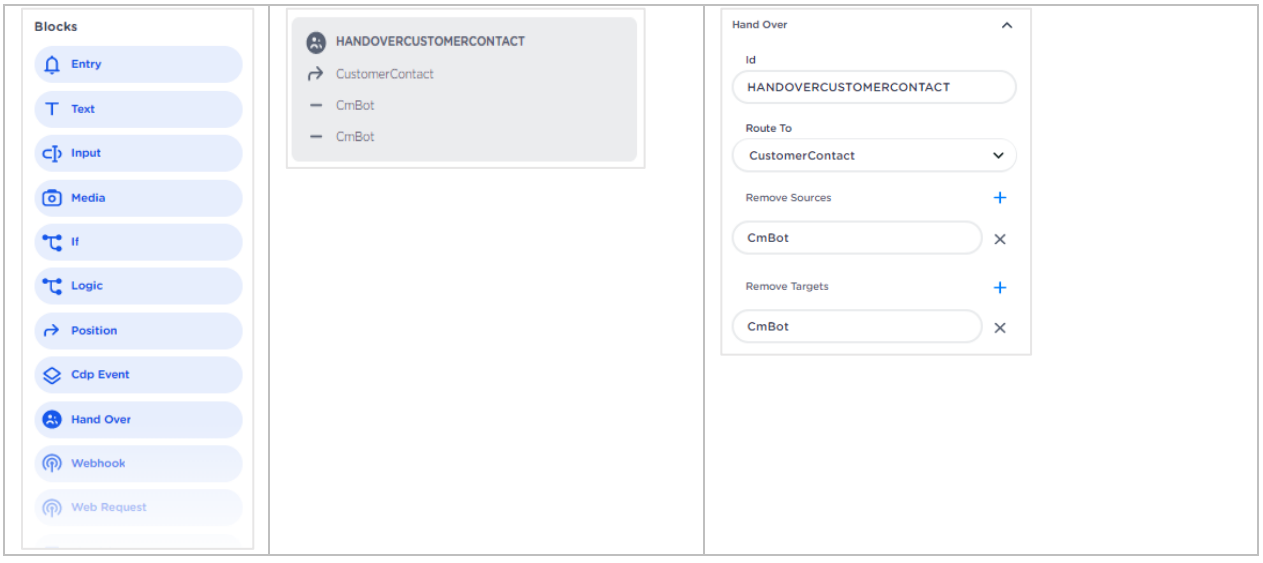
Within Customer Contact we distinguish two types of modes.

1. Hand-over, this is done when the bot has no answer/your customer is at the end of a flow
2. Take-over, when an agent notices a customer is stuck in the bot flow and you want to help them further

Both are possible with the software, but the action for the Hand-over lies on the Bot side, whereas only the agent in Customer Contact knows if he needs to interfere with the chat. Hence Take-overs are initiated from Customer Contact.

When to use a handover:

- Bot has no answer
- Bot gathered all information and a Human can support the consumer now
- Human noticed the communication was abandoned, and wants to take ownership of the chat



The following applies:

- Once a handover is performed, the bot itself lost control. Until a human responds and takes ownership of the conversation no messages will be send to the consumer.
- All conversations handled by the chatbot are visible in Customer Contact. At any time you can interrupt and take over the conversation from there. This is great when you want to test the effect of improvements about the generic chatbot implementation, but also the effect of changes in the flow(s).
- After 24 hours the default routing will be restored and the bot will automatically start answering again.

Properties

Route To: Specifies the primary destination to which messages should be routed from this point forward. This system will get a signal that a Hand over should take place.

Remove sources: Tell the router to stop allowing this system to send messages.

Remove targets: Tell the router to stop allowing this system to receive messages.

A common setup for a handover of the CM bot is to transfer the conversation to Customer Contact to allow follow up by a human agent:

Route To: CustomerContact
Remove sources: CmBot
Remove targets: CmBot

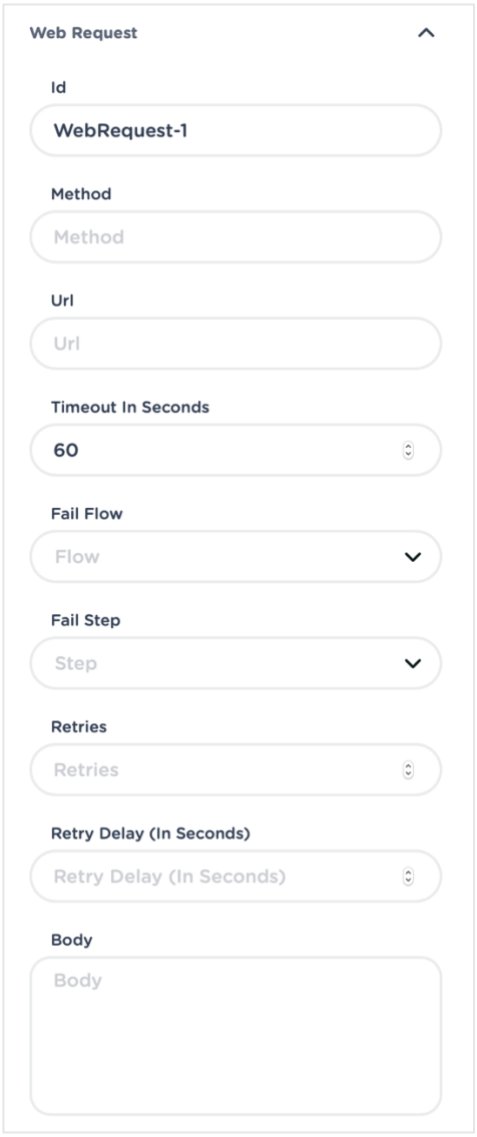
Web Request

A Web-request can be used to send or gather information from an external source. This information you can use in the rest of your flow. Example use cases for a Web Request are:

- Looking up delivery information of a package
- Looking up prices of your products
- Finding information related to the customer profile (i.e.. a company-id and name)

The following applies for a Web-request:

- Web-requests are performed in the Flow your customer is active in. Since we need to send or look-up information externally this step slows down the execution. Usually consumers are not annoyed by this, but when there is a large demand (thus many people reaching out to you at the same moment) this step gets impacted hardest.
- A Web-request should only be performed if the script execution needs information from the reply in the following steps. If you do not need to know the response of the web request, you might want to consider a fire-and-forget Webhook step instead.
- You need to specify any headers you require.
- You can use variables in the URL, body and the headers of the request.



Request properties

Url: The web address of the request to be made

Method: HTTP method used for the request. Supported methods are: "POST", "GET", "PUT", "PATCH", "DELETE"

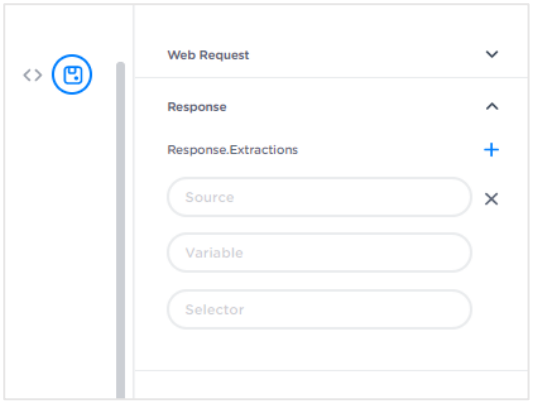
Timeout: Number of seconds before the request will be considered to have failed. Remember that the customers will be waiting for this time before it might see any new feedback, so don't make it too high.

Fail Flow and Fail Step: Indicate the position that the bot should move to when it encounters a problem with a web request.

Retries: Number of times the engine will retry the web request before it fails definitively.

Retry Delay: Number of seconds between retry requests.

Responses



Add a collection of so-called extractions. These allow you to pull information from the response of the web request and put them into a variable.

Each extraction is made up from a source, a destination variable and a selector. You can have multiple extractions.

Source: The source of the extract. One could specify the following options:

@body.json The response body, handled/casted as json. You can specify a json path query as 'selector'.

@body.xml The response body, handled/cased as xml. You can specify an xml path query as 'selector'.

@body.text The plain text value of the response body.

@status_code Numeric HTTP response code of the request.

@is_success Boolean indicating success or failure of the request.

@retries Number of times the request was retried.

Variable: The session variable name in which you want to store the extracted value.

Selector: Required only for JSON and/or XML extractions, leave empty in any other situations. You can specify a selector to extract a specific element from the receive JSON or XML body. For example, to extract the name attribute of the first object inside a returned array use the selector `[0].name`

Webhook

The Webhook step can be used to perform a web request in a fire-and-forget manner. The webhook will be offloaded using a queue and be performed in the background. No guarantees are given that the request will finish in a timely manner. The remaining options are similar to those of a Web request step.

Variables

Every conversation has variables, and these exist until the end of the conversation.

Fixed variables

Certain variables exist for each conversation. These variables cannot be changed. The following variables are provided by the system.

- **$chatId:** A fixed hashed Guid of the combination of channel, conversationClientId and conversationHostId. Can be safely used for external communication from within the chatscript.
- **$conversationClientId** This is the identification number (msisdn, apple chat id, etc) of the end-user.
- **$conversationHostId** This is the identification number of the party that hosts the chat.
- **$channel** The channel that was used of the message that started the session.
- **$startSequence** This is the literal text that was sent to start the current session.

Session variables

Steps like Input or Set Variable can add a variable with a value. You name these variables in the UI.

As described in the section about regular expressions, it is also possible to name variables from regular expressions, to capture input from the consumer.

Another common place to define variables and their content is when parsing the response of a web request.

Using variables

Variables can be used in many of the steps in fields indicating the use of a variable. For instance the Logic step has a field “variable” where you can specify the variable name that you want to apply the Logic to. It is also possible to use variables in many other text fields.

If a field is not specifically used for naming a variable, but is a text value, then you can use a variable by wrapping the name of the variable inside {} brackets. The bot will replace the name variable on the fly, with the value of the variable.

For instance: say the consumer replied to one of our messages with the word ‘Nothing’ and we stored this reply in the variable “menuChoice”. The bot cannot find a good match for this answer and instead we reply to the consumer with the following message:

Uhm... I did not quite understand what you meant with '{menuChoice}'. Let's try again

The consumer will receive the bot reply: Uhm… I did not quite understand what you meant with ‘Nothing’. Let’s try again