

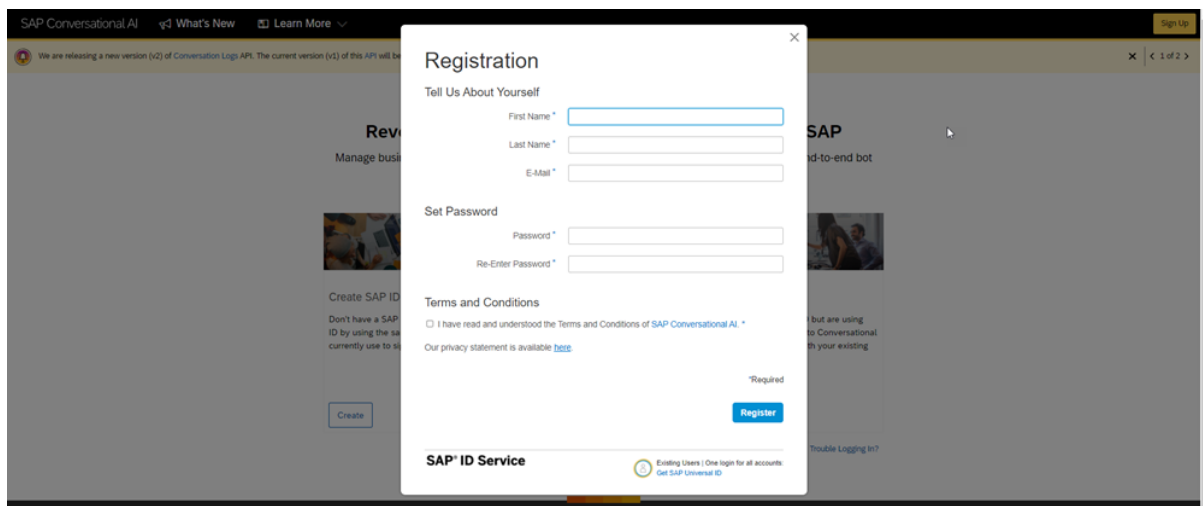
# EDUNET FOUNDATION - Class Exercise Notebook

## LAB 1 – Create a new bot project

### Create SAP Conversational AI account

Go to <https://cai.tools.sap/>, and click Sign Up in the upper-right corner.

Follow the instructions for creating an account.

A screenshot of the SAP Conversational AI registration page. A modal window titled "Registration" is centered on the screen. The form is divided into sections: "Tell Us About Yourself" with fields for First Name, Last Name, and E-Mail; "Set Password" with fields for Password and Re-Enter Password; and "Terms and Conditions" with a checkbox for agreement and a link to the privacy statement. A "Register" button is at the bottom right of the form. The background shows the SAP Conversational AI website with a "Sign Up" button in the top right corner.

SAP Conversational AI | What's New | Learn More

We are releasing a new version (v2) of Conversation Logs API. The current version (v1) of this API will be deprecated in the near future.

Rev...  
Manage busi...

Create SAP ID  
Don't have a SAP ID by using the se... currently use to s...

Create

SAP ID Service

Existing Users | One login for all accounts  
Get SAP Universal ID

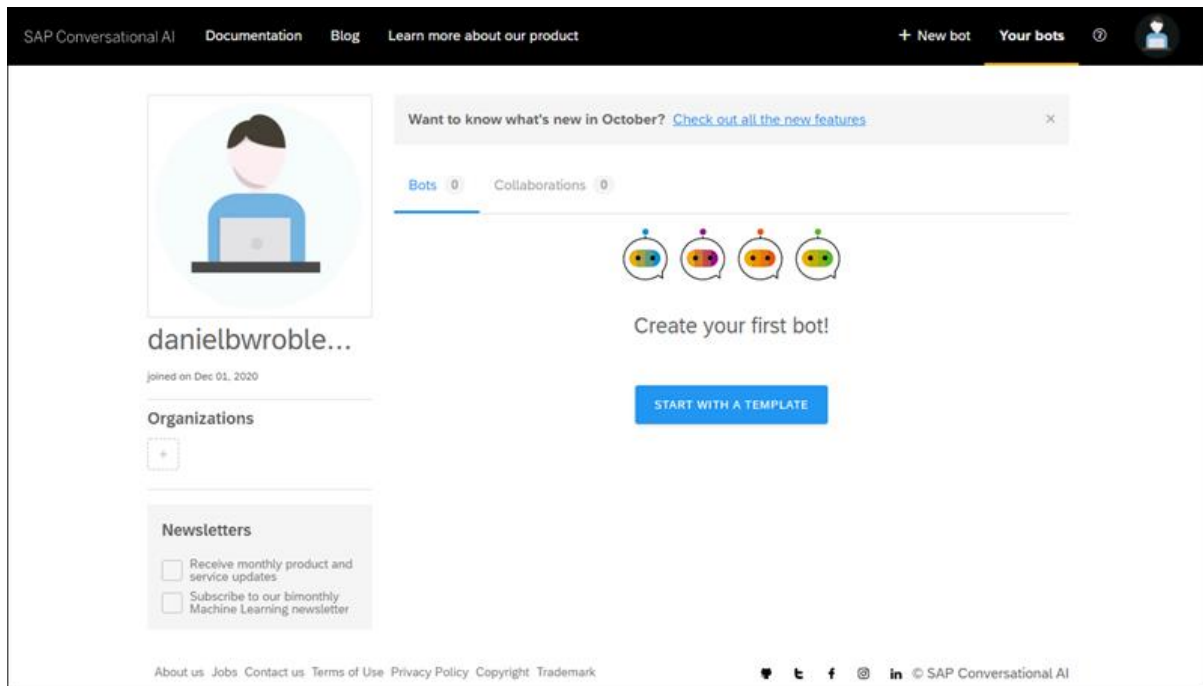
Register

You will get an email with a link to validate your account.

## Create new bot project

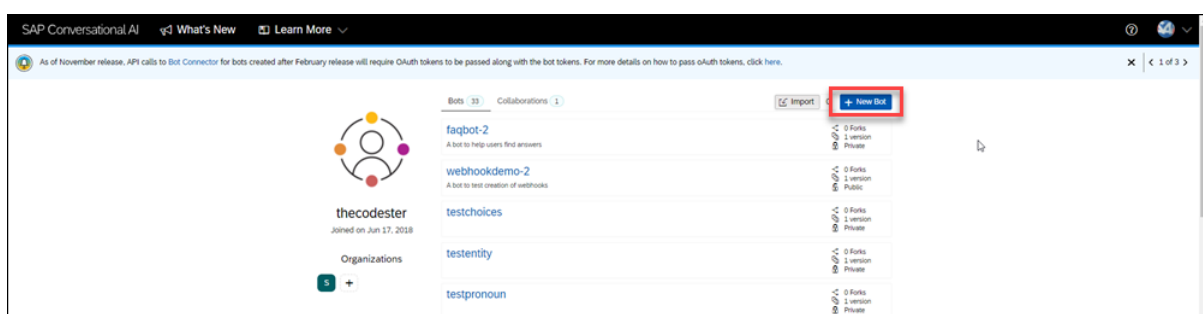
Do one of the following:

If you just validated your account by clicking the link in your email, click Start with a Template.



If you are returning:

- Go to your home page by opening the URL <https://cai.tools.sap/> and sign in.
- Click New Bot.



Fill in the following:

- Select **Perform Actions**.
- For the predefined skills, choose Greetings.
- In the Create your bot section, enter the following:

Field Name	Value
Bot name	<input type="text" value="my-first-bot"/>
Description	A bot that likes to tell jokes and have a little fun

- For **Data Policy**, select the following:
  - Non-personal.
  - Store.
  - Non-vulnerable.
- For **Bot visibility**, select **Public**.

SAP Conversational AI


[What's New](#)


[Learn More](#)

As of November release, API calls to Bot Connector for bots created after February release will require OAuth tokens to be passed along with the bot tokens. For more details on how to pass OAuth tokens, click here.


< 1 of 3 >


1 What do you want your chatbot to do?


  
Perform Actions  
using conversational flows


  
Retrieve Answers  
from FAQ documents


2 Select predefined skills for your bot

  
Greetings

  
Small Talk

  
Weather

  
Customer Satisfaction

  
Set Alarm

3 Create your bot

thecodester / my-first-bot



Description (optional)  
A bot that likes to tell jokes and have a little fun

Topics (optionally categorize your bot with up to 6 topics to improve your bot training)

Click **Create a bot**.

## The 4 stages of a bot's life

< thecodester / my-first-bot-2

 v1  development

Created today at 15:41

A bot that likes to tell jokes and have a little fun

Add Topics

Train

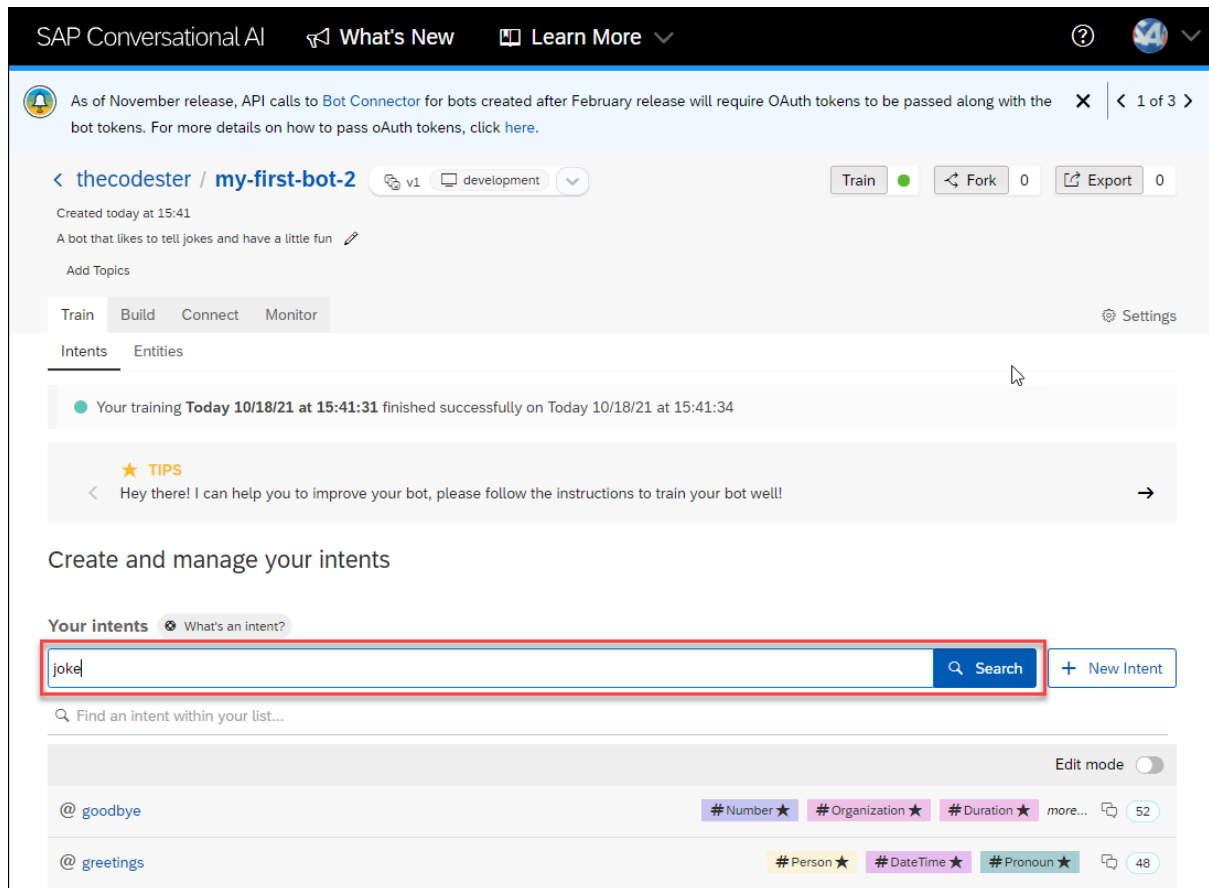
Build

Connect

Monitor

## Fork intent

1. Go to the **Train** tab.
2. In the **Search** box, enter `joke`, and press **Search**.



SAP Conversational AI | What's New | Learn More

As of November release, API calls to Bot Connector for bots created after February release will require OAuth tokens to be passed along with the bot tokens. For more details on how to pass OAuth tokens, click [here](#).

< thecodester / my-first-bot-2 | v1 | development | Train | Fork 0 | Export 0

Created today at 15:41  
A bot that likes to tell jokes and have a little fun

Add Topics

Train | Build | Connect | Monitor | Settings

Intents | Entities

● Your training **Today 10/18/21 at 15:41:31** finished successfully on Today 10/18/21 at 15:41:34

★ TIPS  
Hey there! I can help you to improve your bot, please follow the instructions to train your bot well!

### Create and manage your intents

Your intents | What's an intent?

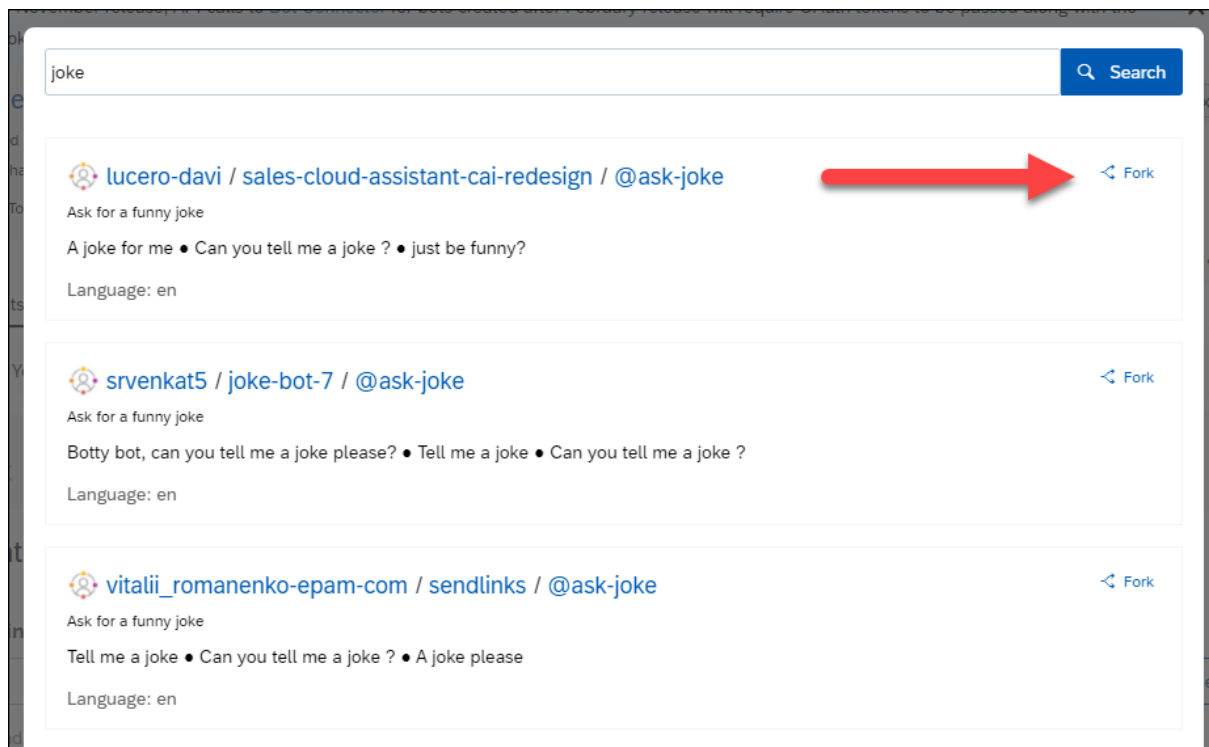
joke | Search | + New Intent

Find an intent within your list...

Edit mode

	# Number ★	# Organization ★	# Duration ★	more...	
@ goodbye					52
@ greetings	# Person ★	# DateTime ★	# Pronoun ★		48

You will get a list of intents to fork.

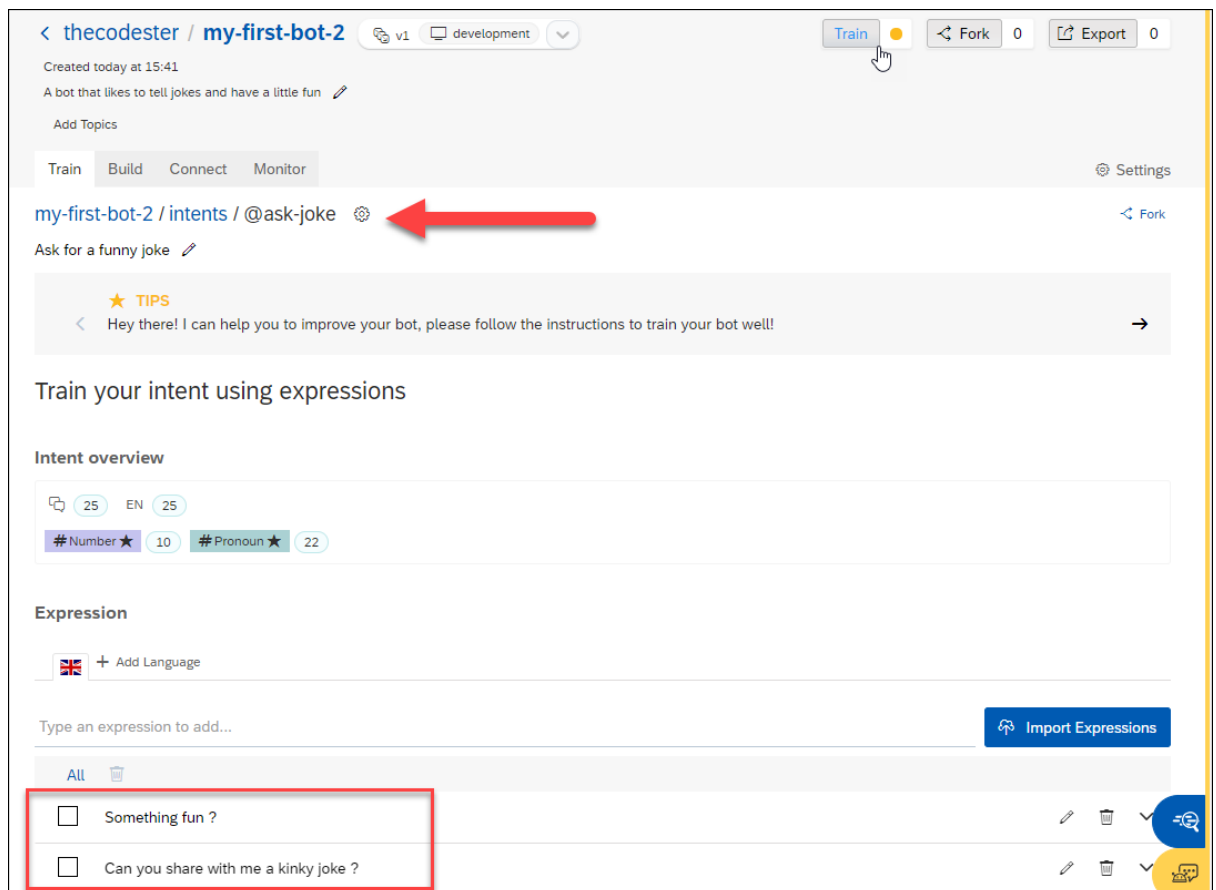


3. Click **Fork** for the first `@ask-joke` intent.

There are many existing intents and the choices are always changing.

4. Explore the intent by clicking on it.

In the intent we forked, there are 2 expressions to discern if someone wants to be told a joke. In yours, you may receive more expressions.



## Create new intents

You will be able to reuse many, many intents created by others. But there are times you will want to create your own.

1. Click the **Train** tab.
2. Create an intent for recognizing good reactions to jokes.
  - Click **Create**.

< thecodester / my-first-bot-2 v1 development Train Fork 0 Export 0

Created today at 15:41  
A bot that likes to tell jokes and have a little fun

Add Topics

Train Build Connect Monitor Settings

Intents Entities

Your training **Today 10/18/21 at 15:41:31** finished successfully on Today 10/18/21 at 15:41:34 but is outdated. Additional changes in your dataset have been made during the training. [Please try again.](#)

**TIPS**  
Some of your intents have less than 50 expressions. Ensure your intent has at least 50 expressions to improve your dataset. Add expressions with different sentences but the same meaning and be sure to have only one sentence by expression

### Create and manage your intents

**Your intents** What's an intent?

Search and fork an intent from the community

+ New Intent

- For the name of the intent, enter **laughs**.
- For the description, enter **A natural reaction to our awesome jokes.**

**Let's create your intent**

thecodester / my-first-bot-2 / laughs

Your intent description (optional)

A natural reaction to our awesome jokes.

Matching Strictness

0 50 100

Cancel Create Intent

- Click **Create Intent**.
3. Create a second intent, this time for bad reactions to jokes.
- Click **Create**.
  - For the name of the intent, enter **lame**.



- For the description, enter **You can't succeed every time.**
- Click **Create Intent.**

## Add expressions

1. Click on the `@laughs` intent.
2. In the expression field, enter a sentence you want your bot to understand, then press **Enter**.

The screenshot shows the Dialogflow console interface for a bot named 'my-first-bot-2'. The breadcrumb navigation indicates the current path is 'theCODester / my-first-bot-2 / intents / @laughs'. The page title is 'my-first-bot-2 / intents / @laughs'. Below the title, there is a description: 'A natural reaction to our awesome jokes.' and a 'Train' button. The 'Expression' field is highlighted with a red box and contains the text 'I haven't laughed that much in a long time!'. To the right of the expression field is an 'Import Expressions' button. Below the expression field, there is a message: 'No expressions added. Start by adding expressions above to build your intent.'

## Test bot

We want to now test to see if the bot can detect when a user says something matching the intent we created.

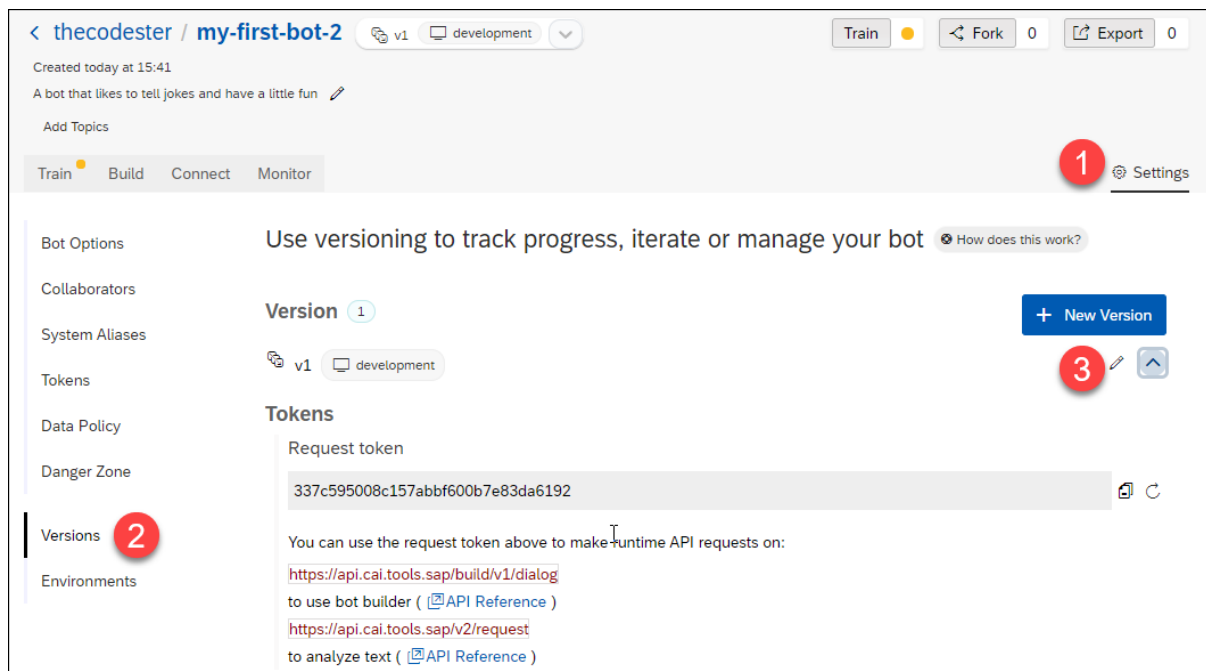
1. Click the **Train** button at the top-right, so that the bot trains itself based on the latest intents and expressions (sometimes referred as your dataset).

The button indicates the current status of training: Yellow is out of date, blue is training, green is up to date..



By default, your bot does not train itself but you must click Train for it to understand the latest intents and expressions.

You can change the setting so the bot automatically trains itself at **Settings > Version > (current version dropdown)**, and scroll down to .



**NLP**

Context management

☐ **Resolve pronouns**  
Map pronouns to entities in the conversation history

☐ **Resolve descriptions**  
Map superlatives to list replies in the conversation history

**Training Mode**

☐ **Automatic**  
*Training is automatically triggered by any change to the bot*

☒ **Manual**  
*You decide when you want to update your training*

**2. At the bottom of the panel, open the Expression Analysis tab**

Create and manage your intents

Your intents What's an intent?

Search and fork an intent from the community

Find an intent within your list...

Find an intent within your list...

Edit mode ☐

@ lame	# Pronoun ★	# Number ★	4
@ laughs	# Number ★	# Pronoun ★	4
@ ask-joke	# Pronoun ★	# Number ★	4
@ goodbye	# Number ★	# Organization ★	# Duration ★

Expression Analysis  
Shift+Alt+E

**3. Enter something that you think should match the intent, like:**

Botty bot, can you tell me a joke please?

If the intent is successfully trained, the test will show that the bot recognized the intent (as well as entities within the expression).

The screenshot displays the SAP Conversational AI interface. On the left, a sidebar contains navigation elements like 'Learn More', 'Fork', 'Export', and 'Settings'. The main area is titled 'Expression Analysis' and shows the following details:

- Language:** English (UK flag)
- JSON View:** Toggle switch (off)
- POST:** `https://api.cai.tools.sap/v2/request`
- User said:** Botty bot, can you tell me a joke please?
- User is referring to:** A red box highlights the entity `@ask-joke`.
- In the expression there are:**
  - `#Pronoun ★` you
  - `#Pronoun ★` me
  - `#Number ★` a
- See more in the JSON view**

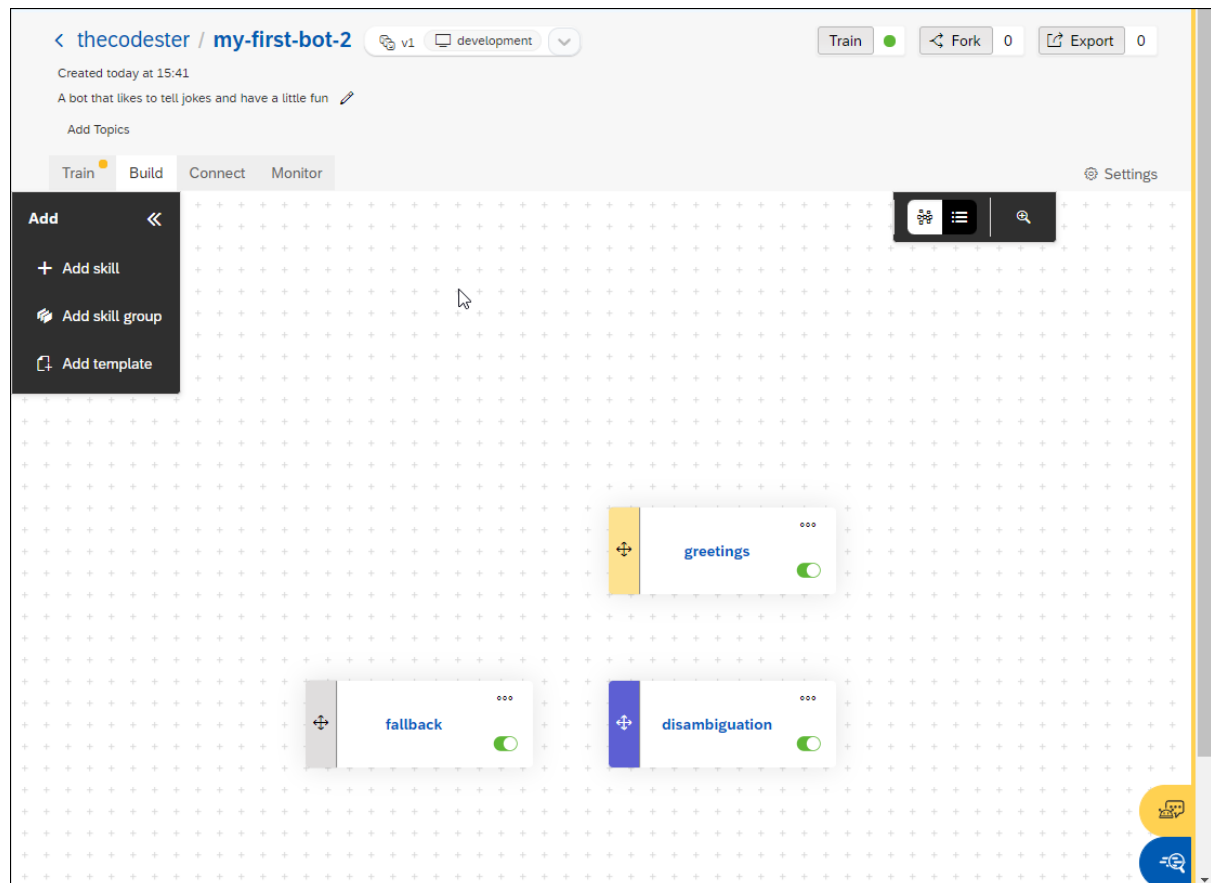
At the bottom, a chat input field contains the text 'Botty bot, can you tell me a joke please?'. A red arrow points to the microphone icon on the left of this input field.

If the bot did not recognize the intent (or selected the wrong intent), you will need to train your bot some more by adding additional expressions (Step 6).

## Manage conversation flow with skills

Now that your bot knows how to understand people who talk to it, it's time to give your robot some skills.

Open the **Build** tab.



Inside, you'll find the **Bot Builder**, which helps you construct the conversation flow of your bot.

### *What is a skill?*

Each skill represents one thing that your bot knows how to do. Your skill can be complicated (e.g., manage payment by credit card) or quite simple (e.g., answer a basic question).

Just like intents, you can create a skill from scratch or inherit skills from other bots you've created or from other people's bots.

In our project, choose the predefined skill **Greetings**.

The **Greetings** skill – like all skills – has 4 tabs:

- **README.md:** A description of the purpose of your skill
- **Triggers:** The conditions that must occur – generally the intents that the user must express – for the skill to be executed
- **Requirements:** Information that must be collected in order for the skill to be executed
- **Actions:** The action to take (basically, this is the skill)

If you navigate through the tabs, you'll see that the **Greetings** skill is structured as follows:

- It is triggered if either the intent `@greetings` or `@goodbye` is matched.
- It has no requirements because it does not need to collect additional information. That means that it will execute actions directly after being triggered.
- It has two possible actions: If the `@greetings` intent is matched, it sends a random welcoming message chosen from a list. If the `@goodbye` intent is matched, it does the same thing, but picks the message from a different list.

## Create new skill

Now let's build a skill to tell a joke.

1. Go back to the **Build** tab – where you see the canvas.
2. Click **Add skill**

There are several types of skills:

- **Business** skills reflect the core purpose of your bot.

- **Floating** skills complement your bot's core business skills (for example, small talk).
  - The **Initialize** skill (there can only be one in a chatbot) is triggered when the conversation with the user starts.
3. Call your skill tell-me-a-joke, set the type as Business, and click Add.

**Add Skill**

**Name**  
 tell-me-a-joke 241  
 ex. book-flight, checkout

**Type**  
 Business Floating Initialize  
 Business skills reflect the core purpose of your bot.

**Activation** ☒  
 Your skill is active. If you deactivate it, your bot will no longer use it when chatting.

**Title**  
 Your bot automatically uses this skill title when disambiguating during chat. If you don't specify one, your bot will display: tell-me-a-joke

Type a short title for this skill's quick reply button...

Cancel Add


4. Click the new tell-me-a-joke skill, and open the Triggers tab.
5. Add 3 If statements, one for each of the @ask-joke, @laugh, and @lame intents.



- 
- Trigger this skill only if: [How do triggers work?](#)
- If  [Save](#)
- | User says  | entity is detected | in the conversation    |
|------------|--------------------|------------------------|
| @lame      | #Cardinal          | _sentiment             |
| @laughs    | #Color             | _source                |
| @ask-joke  | #Datetime          | _language              |
| @goodbye   | #Distance          | _processing_language   |
| @greetings | #Duration          | _memory                |
|            | #Email             | _skill                 |
|            | #Emoji             | _skill_occurrences     |
|            | #Interval          | _conversation_language |
|            | #Ip                | _client_info           |
|            | #Job               |                        |
|            | #Language          |                        |
|            | #Location          |                        |
- [+ New Intent](#)
- You can easily use the entities enrichment, ex: #location.lat
- [View more about conditions](#)

- It should now look like this:





Trigger this skill only if: [How do triggers work?](#)






If @ask-joke is-present  

OR

If @laughs is-present  

OR


If @lame is-present  





6. Go to the Actions tab.
7. Click New Action Group, then Add Condition.
8. After the If, select the @ask-joke intent, and click Save.
9. Click Choose Message Type, choose the Text format, and type in a really good joke, like:



I just flew into town, and boy are my arms tired.

When all requirements are met, craft your bot's response [What's an action?](#)



If @ask-joke is-present  





**Add a new message (randomly picked)**

I just flew into town, and boy are my arms tired.

<> [Markdown syntax is disabled.](#)

≡

 Choose Message Type  Connect External Service  Connect Fallback Channel  Update Conversation

You can also set a delay (optional) between two messages, up to 5 seconds. This might be useful when the messages your bot sends are quite long and need time to be read by the user.


10. Click Save.

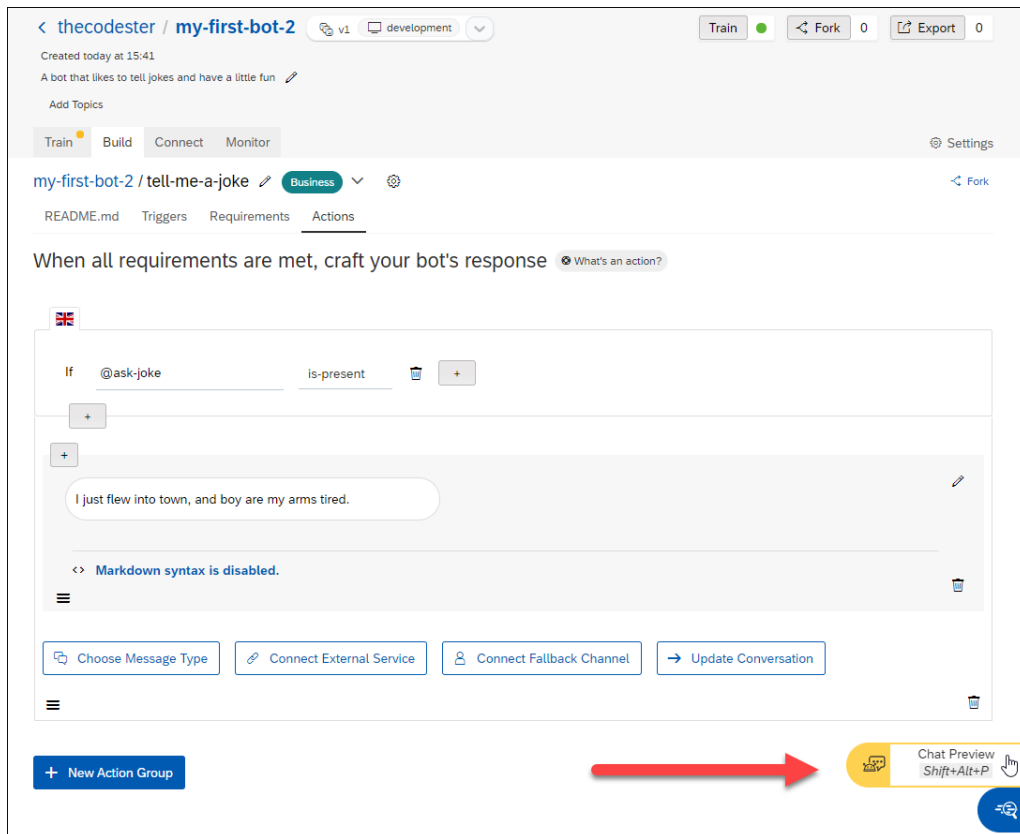
## Test by chatting with bot

Before releasing your bot to your audience, chat with it in real situation. This will help show you how the bot will behave in a “real” conversation.

Double-check if there are any errors in the conversation flow (e.g., bad answers, fallback errors, weird behavior) and correct them before sending the bot out.

Chat with your bot as often as you can while you build it, which will make it easier to find problems.

Click on the bottom-right yellow button Chat Preview and start sending messages. Use the clear icon  at the top of the panel to refresh the chat.



- If you say **Tell me a joke**, the bot will tell you a joke.
- Try other phrases and see how the bot responds. If the bot does not respond properly, then try improving your intent with additional expressions.

