The PetStore

Agentic Workflows Lab Guide

Author: Martijn Bosschaart (martijn@devrev.ai)

Version: 1.5 | Release Date: 24-nov-2025

Table of Contents

Introduction

Welcome to the PetStore!

The PetStore is a fictional store and services frontend, designed for comprehensive training on DevRev Workflows and Automations, and allows you to build a solid understanding of Conversational Customer Experience and Customer Support use cases.

The PetStore site is fully based on an API backend, allowing for easy integration with DevRev. It also allows you to hook up your own DevRev Org to it, without the need to completely reconfigure the website. Multiple users can work with the solution in different configurations at the same time.

By default, the PetStore is connected to the Petstore org, which already is configured with agents, workflows and kb articles so you can get a feel for the possibilities.

This lab guide will describe the following:

- How to set up your DevRev and PetStore accounts,
- How to customize PLuG and configure it for your own org.
- How to set up Agentic AI workflows in DevRev that integrate with the PetStore.

Lab Legend:

Indicates an action to be taken in the DevRev UI
Indicates an action to be taken in the PetStore UI

1. Preparing your DevRev org

First, we need to perform a basic configuration in the DevRev org to prepare it for integration with the PetStore.

1.1 Setting up PLuG

Step 1.

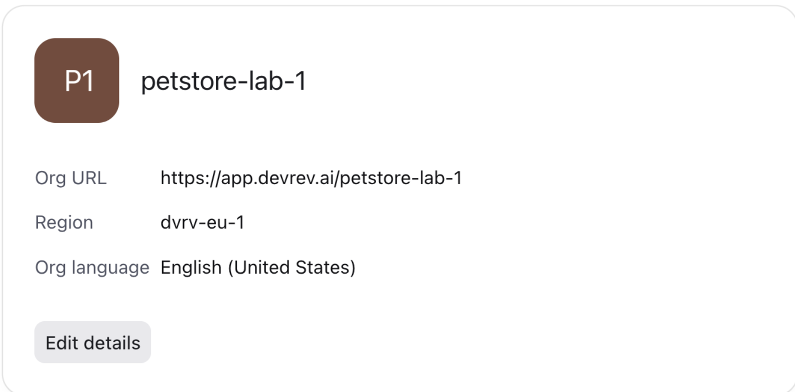
- Log into your DevRev account and select the org which you want to integrate into the PetStore. If this is a completely new org, complete the onboarding wizard. It doesn't matter how to configure your profile.

Note: If you don't have a DevRev account or org yet, you can create a brand new one by visiting https://devrev.ai and register for an account and org there.

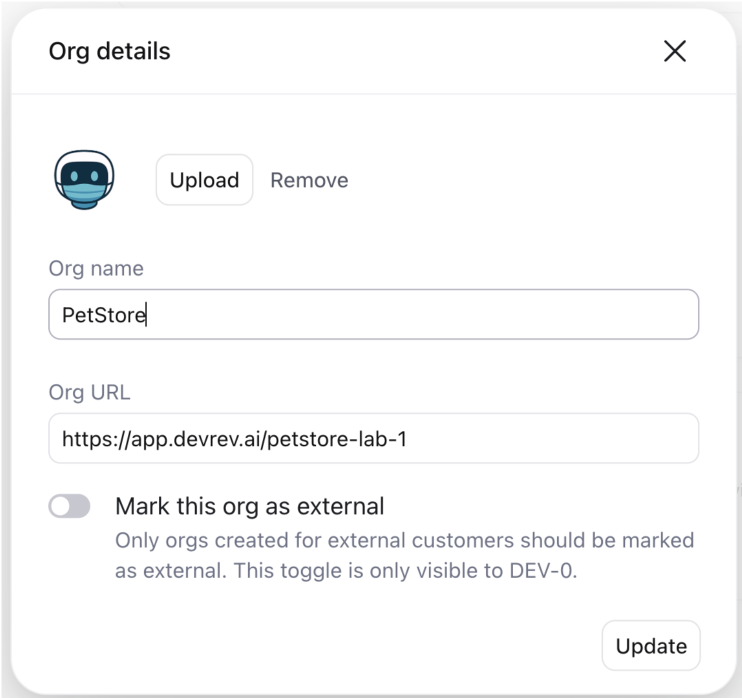
Step 2.

Let's first change the appearance of the default avatar, which is used in conversations.

- Go to the **Settings** menu by clicking on the cogwheel icon next to your initials in the top left of your screen.
- Then in the menu options, click on the **General** option. This opens the following window:



- Click **Edit details** and change the Org Name to PetStore. Then also upload the `AI-VET_avatar.png` file from the `Assets` subdirectory of the `PetStore Materials`.



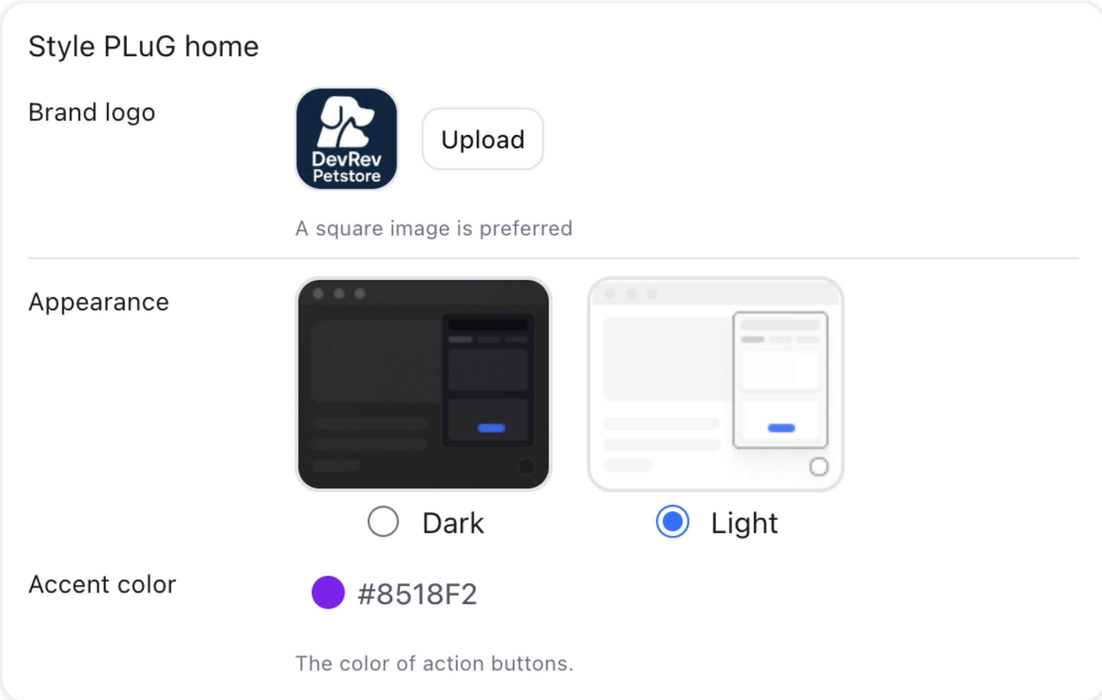
- Finally, click **Update**.

Step 3.

- Select the **PLuG Chat** menu option under the **PLuG & Portal** section.
- On the configuration tab you will find **Your unique app ID**. Copy this to a note for use later.

Step 4.

- Click on the **Styling tab** and then upload a new Brand Logo (it should now be the AI-Vet avatar, but we want to change that.) Use the `Petstorelogo.png` from the `Assets` subfolder in the `PetStore Materials`. Feel free to also change the Appearance and Accent Color to your liking so you really know it's "your" PLuG that is linked to PetStore later.

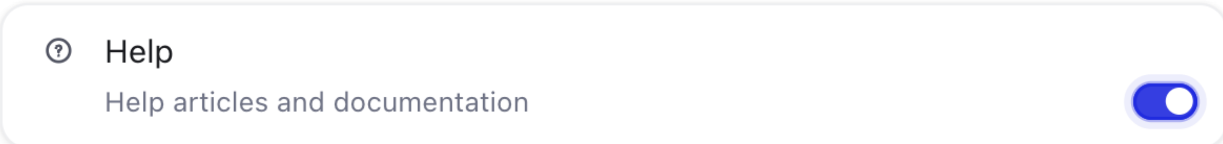


- To save this, click the **Save and Publish** button on the top right of your screen.

You can then also choose to visit the extended playground to test PLuG.

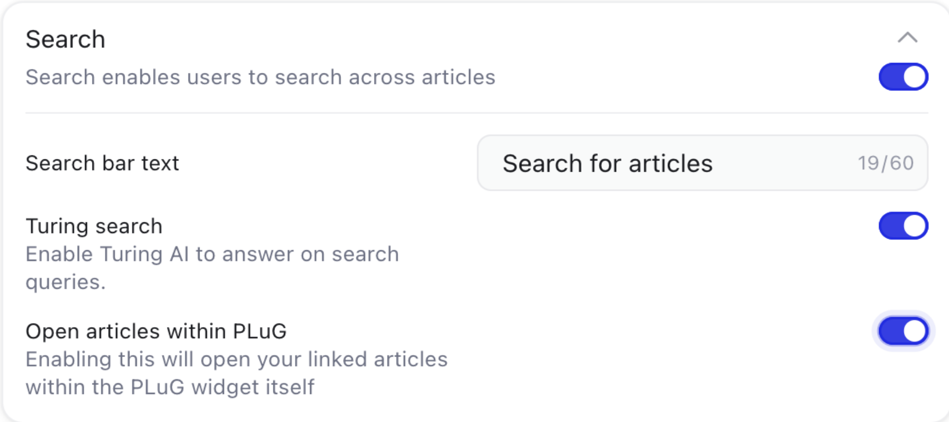
Step 5.

- On the Tabs tab, enable **Help**. Again, click **Save and Publish**.



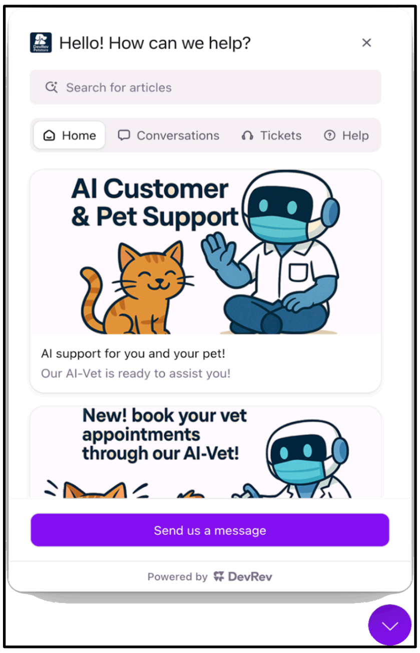
Step 6.

- On the **Layout** tab, click the little arrow pointing down in the **Search** box. This opens a couple of extra options. Switch on both **Turing Search** and **Open Articles within PLuG**.



Step 7.

- Modify the "**Your in-app Support Center**" card.
- Upload a new graphic for it. Use the `AICustomerSupportNudge.png` image from the `Assets` folder in the PetStore source.
- Then change the title to "*AI support for you and your pet!*" and the description to "*Our AI-Vet is ready to assist you!*".
- Edit the second card (called "Start by adding your own cards") and use the `AI-VetNudge.png` as the graphic.
- Give it the title "*Make appointments directly with our AI-VET!*" and give it a description of "*No more filling in boring forms, just talk to the AI-VET!*".
- Add the link "https://petstore.devrev.community/appointments/".
- Finally, click the **Save** and **Publish**.



Step 8.

- Select the PLuG Nudges option.

Now we are going to add three so-called Nudges. These can either be text banners showing at the top of your screen, or full cards with images and text with anchor off the PLuG launcher icon. You can use these to invoke user action or schedule info impressions at scheduled intervals, and all originate from PLuG so no need to modify your website.

- Click **+Nudge** in the top right of your screen.
- Select the type of **SpotLight** and click **Create**.
- On the content tab, give it a title on the top line (click the pencil to edit), i.e. "*Product Support*".
- Fill in the title "*Product Support*" and the Description "*Need help with a product? Ask us!*".
- as cover image upload the `Needhelp_nudge.jpeg` from the PetStore source's `Assets` folder
- as Action choose **Open widget**.

- Via the Rule tab, add a **Page rule**, and set the URL to "**is**" and then **https://petstore.devrev.community/products/** . Leave everything else empty. This will cause the nudge to show immediately when you enter the products page.

- Save this nudge, by clicking the **Publish** button on the top right of your screen.
- For the 2nd Spotlight nudge, use the name "*AI Customer & Pet Support*", the title "*Our AI-VET is ready to assist you*", and as the cover image upload the `AICustomerSupportNudge.png` file.
- The **Action** should be "**Open Widget**".
- On the **Rules** tab set the URL to match **https://petstore.devrev.community/** and set **When to send** to *2* (which indicates a 2 second delay before showing.
- **Publish** the nudge.
- For the 3rd Spotlight nudge use the name "*AI-VET*", the title "*AI-VET Appointments!*", and upload the `AI-VET_nudge.png` image as cover. The **Action** should be **Open Widget**.

- On the rules page, match the URL to **https://petstore.devrev.community/appointments/** with *2* second delay and **Publish** the nudge.

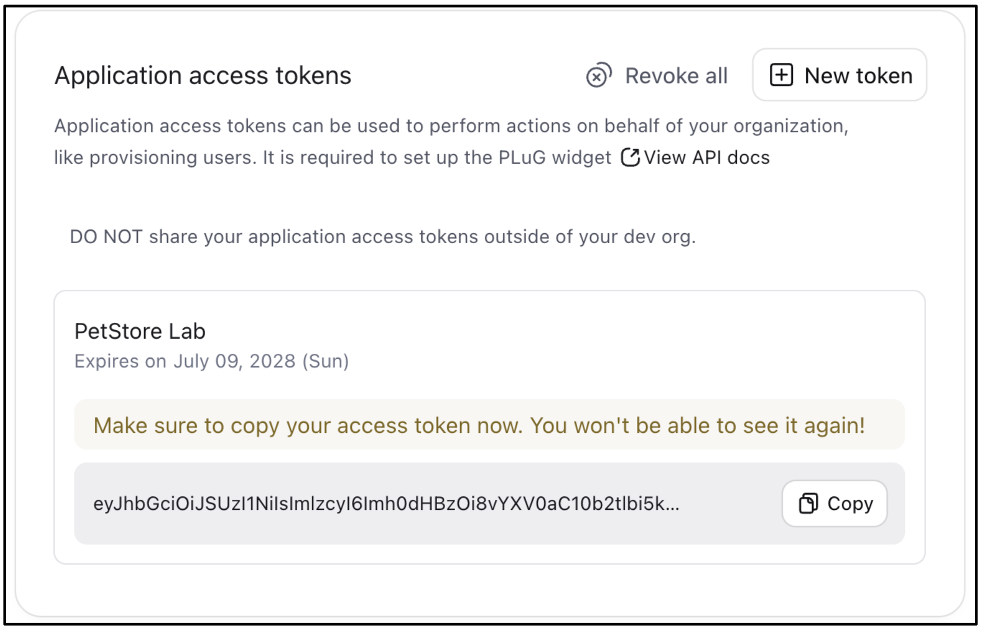
Feel free to add another banner spotlight of your choice.

Step 9.

- Click on **PLuG Tokens** in the menu on the left.
- Then click on the **New Token** button inside the **Application access tokens** box.

> We need this Application Access Token (or “AAT”) to be able to have PLuG authenticate users. During login of users to the PetStore site, the session token endpoint at the DevRev side is also called, passing the user information. It receives a Session Token in return. During initialization of the PLuG widget, it presents this session token back to DevRev. This ensures there is a match between the logged in user, and the PLuG session. It also allows PLuG to automatically create a “RevUser” in DevRev if one does not yet exist.

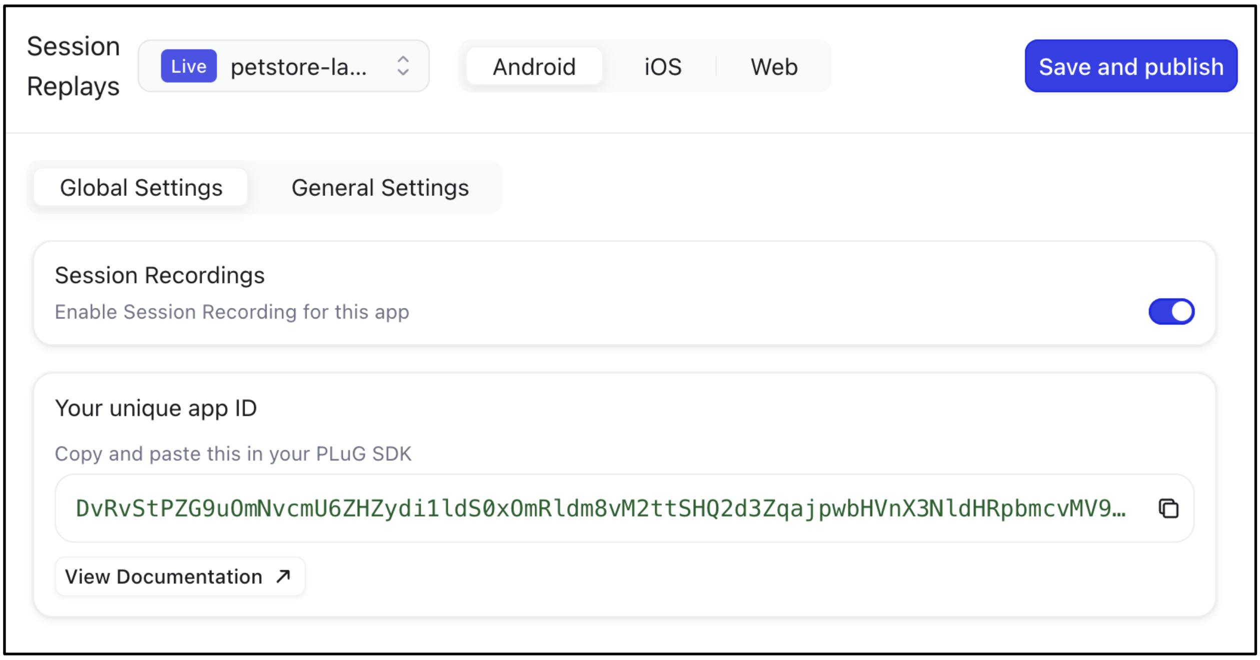
- Give it a name and an expiry and click on **Generate**.



- Once generated, copy the key to a note, as you won't be able to see it again and we need it later.

Step 10.

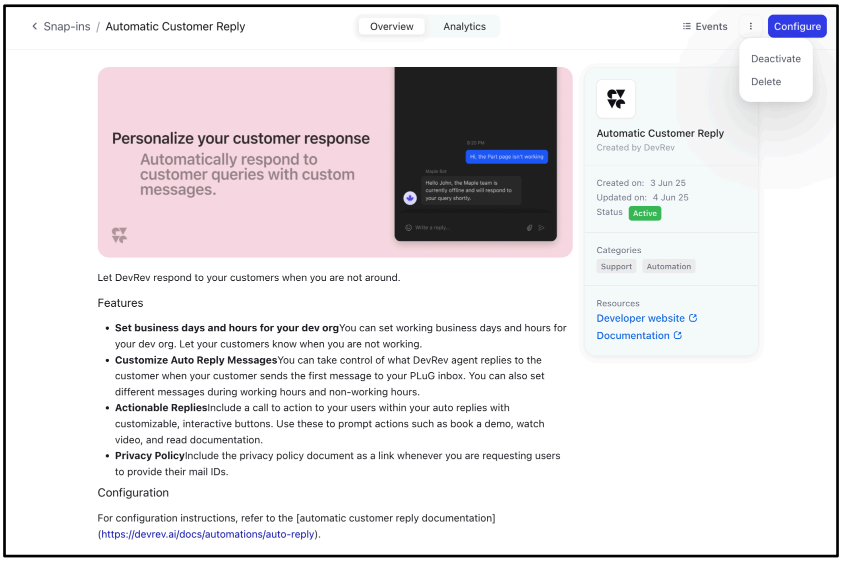
- Click on the **Session Replays** menu option. This will allow us to enable Session Analytics for the site, which are recorded through PLuG.
- Select the **Web** tab and set Session Recordings to **on**.
- Click **Save and Publish**.



Step 11.

While it is not directly part of PLuG, we want to disable the Automatic Customer Response Snap-in for now.

- Visit the **Snap-Ins** option in the Settings menu and click on the tab "**Installed**".
- Find the **Automatic Customer Reply** Snap-in and **Deactivate** or **Delete** it by clicking on the icon with the 3 vertical dots, top right hand side of your screen.



Congrats! PLuG is now ready to be used.

1.2 Configure an Agent for PLuG

To be able to have PLuG respond, we first need to create an **Agent** and then a **workflow** that routes the PLuG conversation to that Agent.

Step 1.

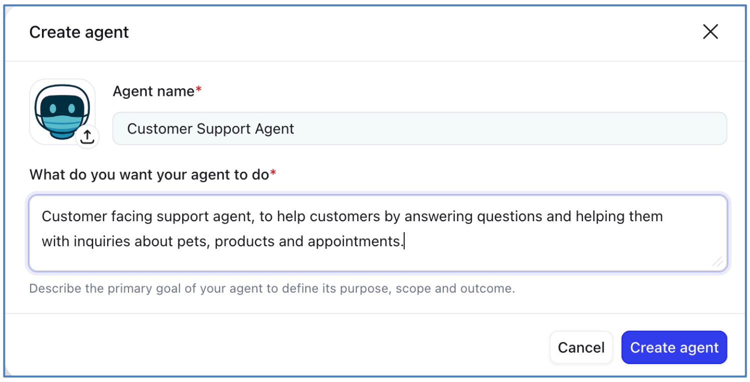
- Log into your DevRev org and go to the **Agents** option in the **Settings** menu.

PRE-GA NOTE FOR INSTRUCTORS:
Before you see the new agent builder and the new node types required, you need to have the new Agent builder features enabled via a Pull Request (see **https://github.com/devrev/archon-policy/pull/11589** for an example)

The Agent Builder will not have any agents yet if this is the first time you use your org.

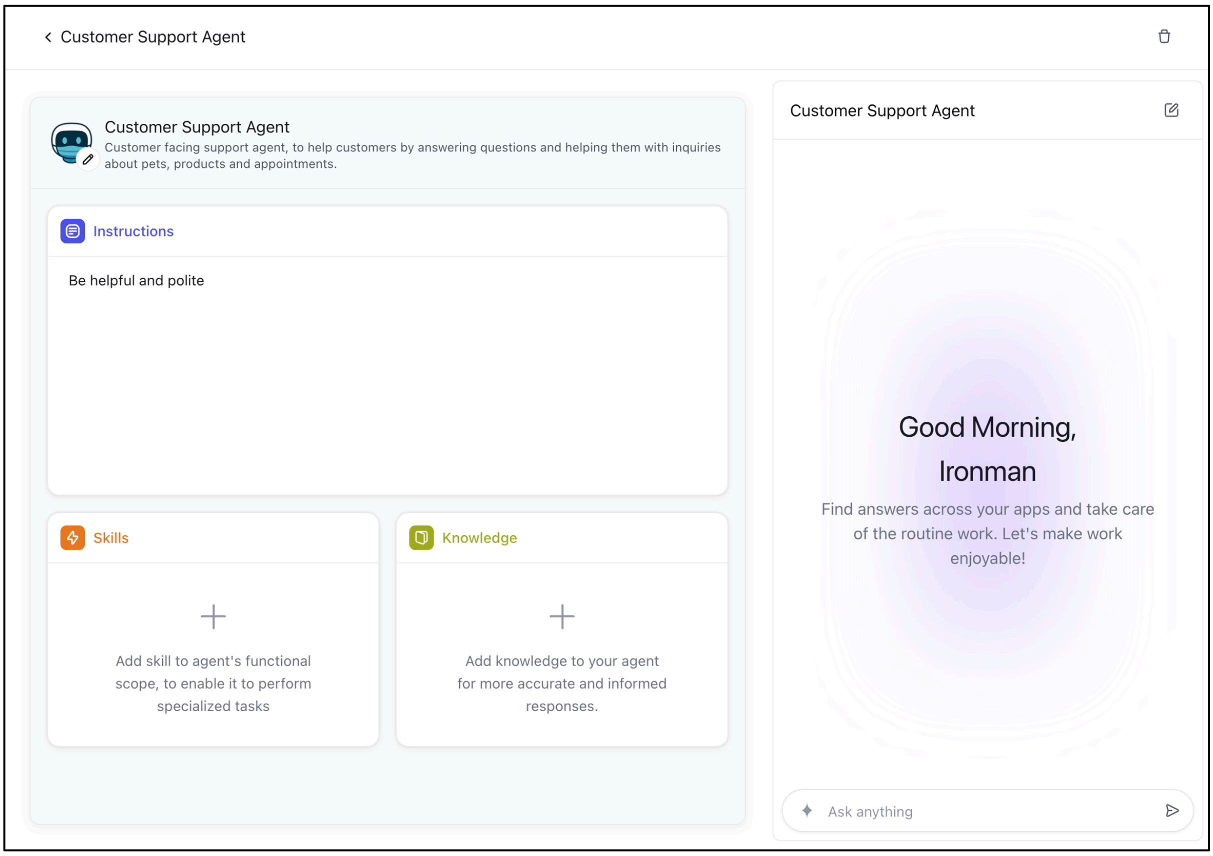
Step 2.

- Click on the **+Agent Actions** button on the top right of your screen.
- Upload the `AI-VET_avatar.png` avatar, and give the Agent a name, i.e. "*Customer Support Agent*"
- Then give it a goal for example: "*Customer facing support agent, to help customers by answering questions and helping them with inquiries about pets, products and appointments.*"



- When done, click **Create Agent**.

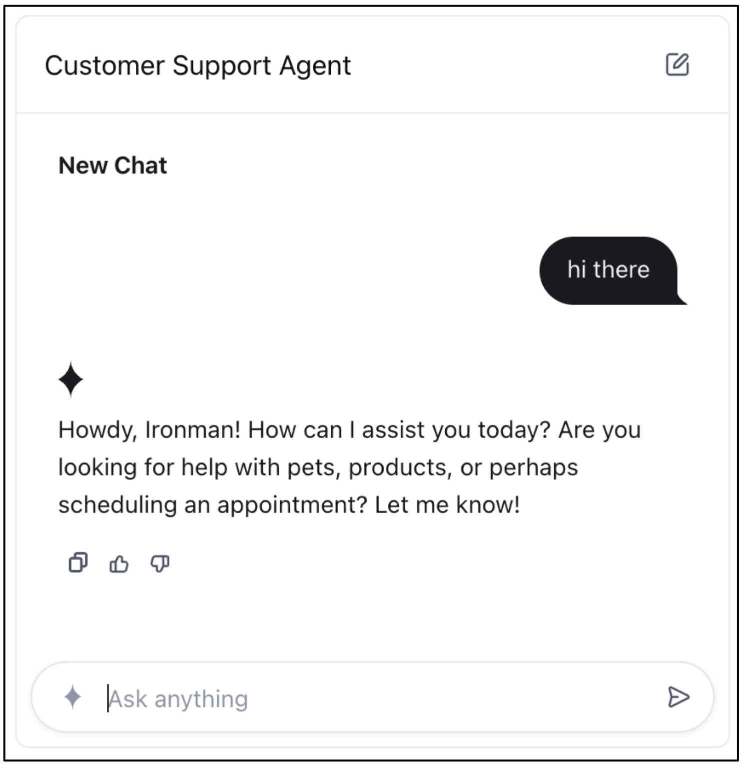
We now have an initial agent setup with a very basic prompt.



Step 3:

For now, we are going to make a small change to the default prompt that allows us to test the setup a bit later.

- Click on the "**Instructions**" pane. When it's selected, a button "**Edit Instructions**" becomes available. Click it.
- Add an additional line that tells it to always open with a certain greeting. For example, "*Always greet the customer with Howdy!*" or something else that you'd like.
- When done, click **Save changes**. You can now test this, by typing anything in the test pane on the right. Try this now.



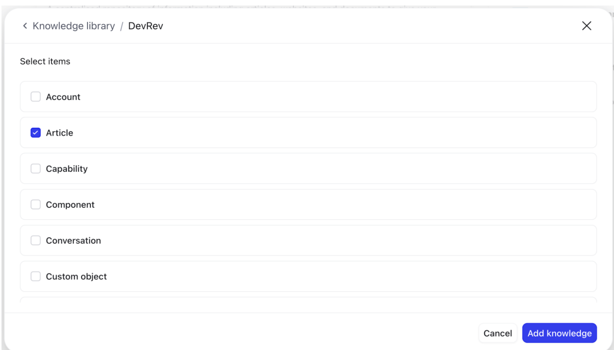
Perfect, our agent is following our instructions!

Step 4.

The next thing we want to do is add Knowledge. In this case we want the Agent to be able to tap into Articles we are going to add a bit later into the Knowledge Base.

- Go back to the main Agent screen where you can then click on the **Knowledge** pane.

- In the next screen click “**Add Knowledge**”, then click on the **DevRev** button. These are all the resources within DevRev that are currently available to the agent.



- From the list of objects, only select **Article**, and then click **Add knowledge**.

You should now see the Article category being attached to the Agent as a Knowledge Source.

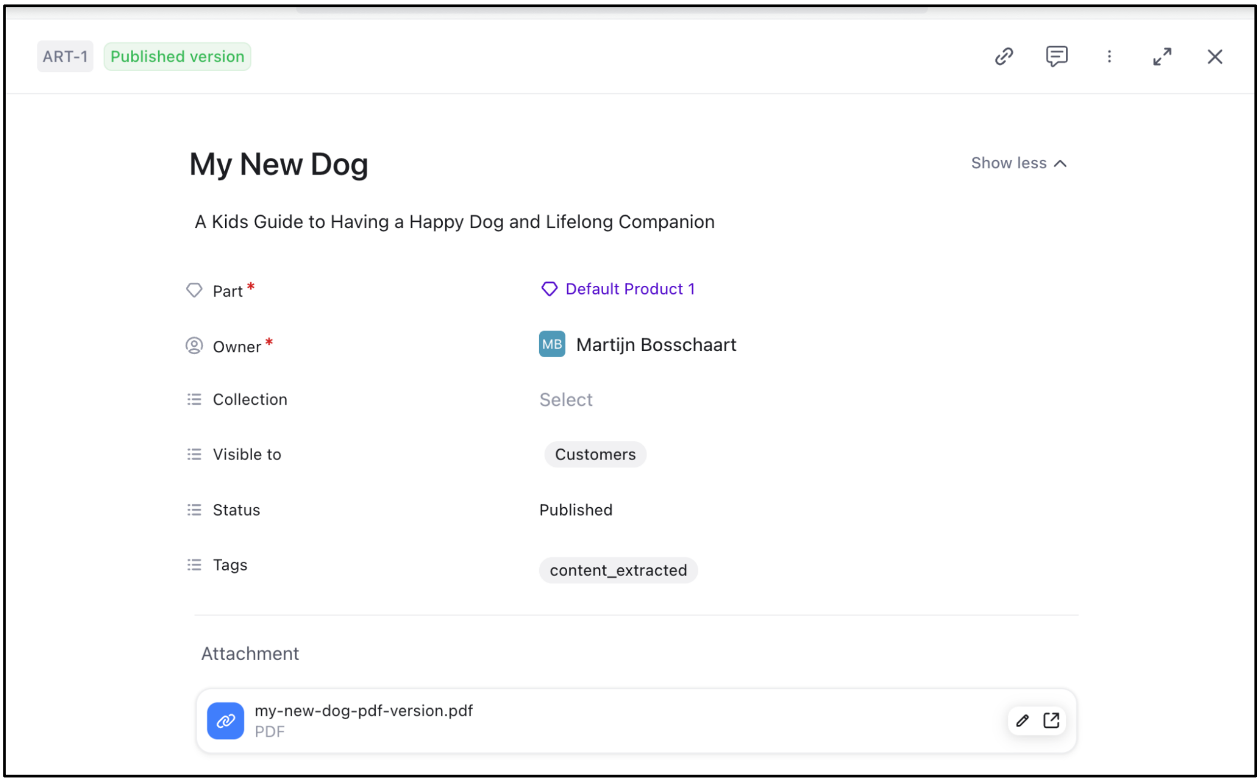
Step 5.

We now need to create a knowledge base article to test.

- Go back to the **Settings** menu and navigate to **Knowledge Base**.
- Click on the **+Article** button in the top right of your screen.

The article we are going to create is going to be based on a PDF file we are going to upload.

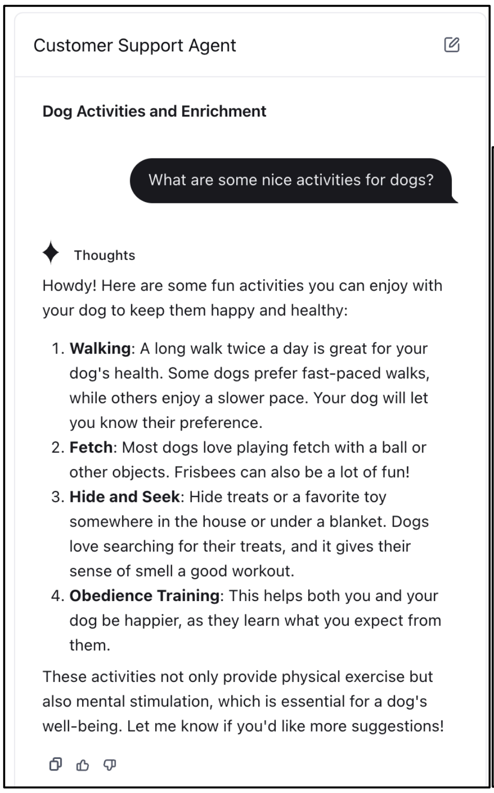
- Enter the following information (you can click on the items to edit):
  Title: *My New Dog*
  Description: *A Kids Guide to Having a Happy Dog and Lifelong Companion.*
  Part: **Default Product**
  Owner: yourself
  Visible to: **Customers** (this means to everyone that can talk to the agent).
- Choose "**Upload a file**" and provide the `my-new-dog-pdf-version.pdf` file from the `Assets` subfolder in the `PetStore Materials`.
- When the upload is complete, **Create (and Publish)** the article via the button at the top right of your screen.

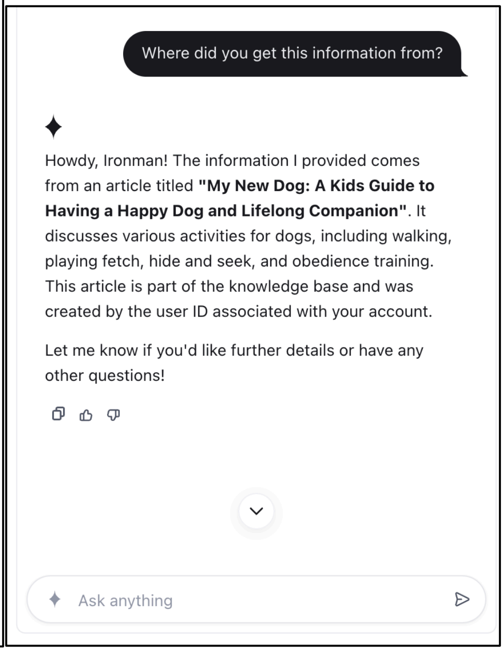


The article is now published, and the PDF is being indexed. This is a relatively small PDF and indexing just takes a few seconds.

- To test if the document is properly added into the **Knowledge Graph**, go back to **Agents** and select your agent.
- Open a new chat and ask the following: “*What are some nice activities for dogs?*” and ask it in a follow up question where it got its information from.

If everything went correctly and the PDF was indexed, the agent will answer with the exact knowledge from the PDF as shown in the next screenshots:





You can now verify this information to be correct by opening the PDF itself. On page 5 and 6 you should find this exact information. The agent is now set up to also handle our knowledge-based questions. We will add more knowledge later.

1.3 Conversation routing from PLuG to Agent

The final thing to do is to make sure that any conversations that originate from PLuG are routed to the correct agent.

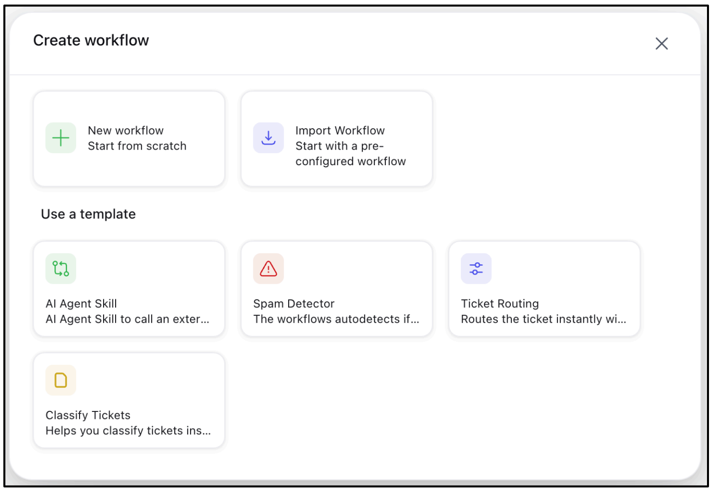
For this we are going to make a Trigger based workflow. Triggers are events in the DevRev system, like a ticket being created, but also a conversation being started. We can use this event to pick up the conversation start and route it to a Talk to Agent node.

Step 1.

- Go to the **Workflows** option in the **Settings** menu.

When we start out, there are no workflows in either the Active, Paused or Draft tabs.

- Click on the **+ Workflow** button on the top right of your screen.



We can now either make the workflow needed by hand, or you can import the workflow from a template that is provided in the `Workflows` folder of the `PetStore Materials`.

For the sake of time, as we are going to create more workflows later we are going to import from a template

- Click on **Import Workflow Start with a pre-configured worklow**.
- Browse to the `TalkToAgentWorkflow.json` file in the `Workflows` subfolder of the `PetStore Materials`.

The workflow will now open on the canvas, with two nodes, **Conversation Created** and **Talk to Agent.** We will make a few modifications to this.

Step 2.

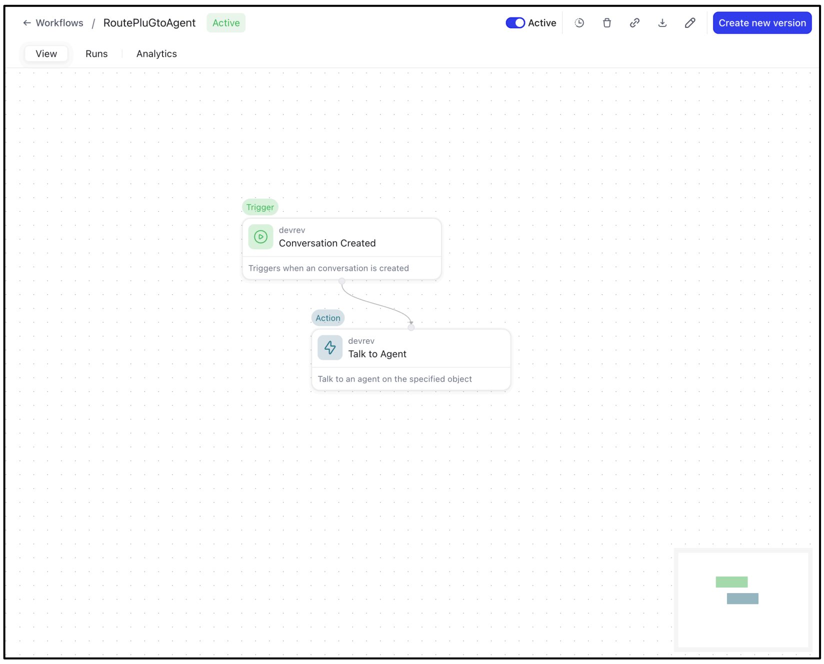
- Click the pencil in the top right-hand section of the screen.
- Change the name to *RoutePLuGtoAgent.*
- Enter a brief description, i.e. *routes PLuG conversations to Customer Support Agent.*
- To save this, click the x top right corner of the window.

Since this triggers on all Conversations that are created, which might also be from other channels such as Email or Slack for example, you can add a Control / logic node to it so the rest of the workflow will only be carried out if the channel is PLuG. For now, we leave it as is.

Step 3.

- Click on the **Talk to Agent** node to open its configuration.
- Make sure the **Agent ID** is set to the correct agent you created. If you started this exercise in a completely empty org, it should show **AGENT-1**. If you already have existing agents, the ID will be different. To verify you can click on the shown Agent ID and it will show you a list of Agents to select from.
- All the other options are already correctly set, but free to investigate.
- When done, close the window and click the **Validate** and then **Deploy** button.

The UI should briefly show the indicator "*Activating*" at the top of the screen.



Step 4.

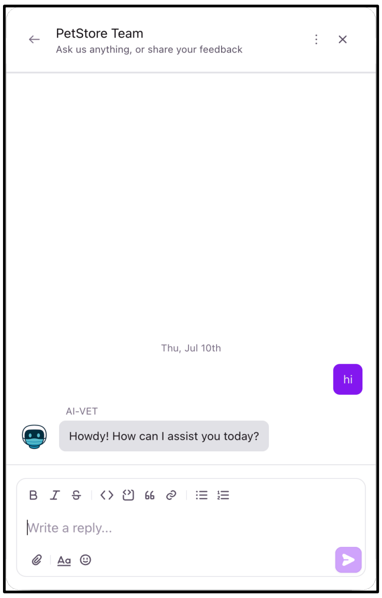
- Click on **Create new version**
- It will then ask to create a new version, so do.
- When in edit mode, click the pencil icon on the top right.
- You can now edit the display name of the Workflow as well as its icon. Change it to “*AI-VET*” and upload the AI-VET avatar.
- Close the window when done and **Publish** the Workflow again.

Step 5.

When it’s done publishing, we can test PLuG.

- Navigate to the **PLuG Chat** menu option in the Settings menu.
- Click on **Send us a message** and say “*hi*”.

If all went well, the agent will now reply as the AI-VET and say “Howdy!”.

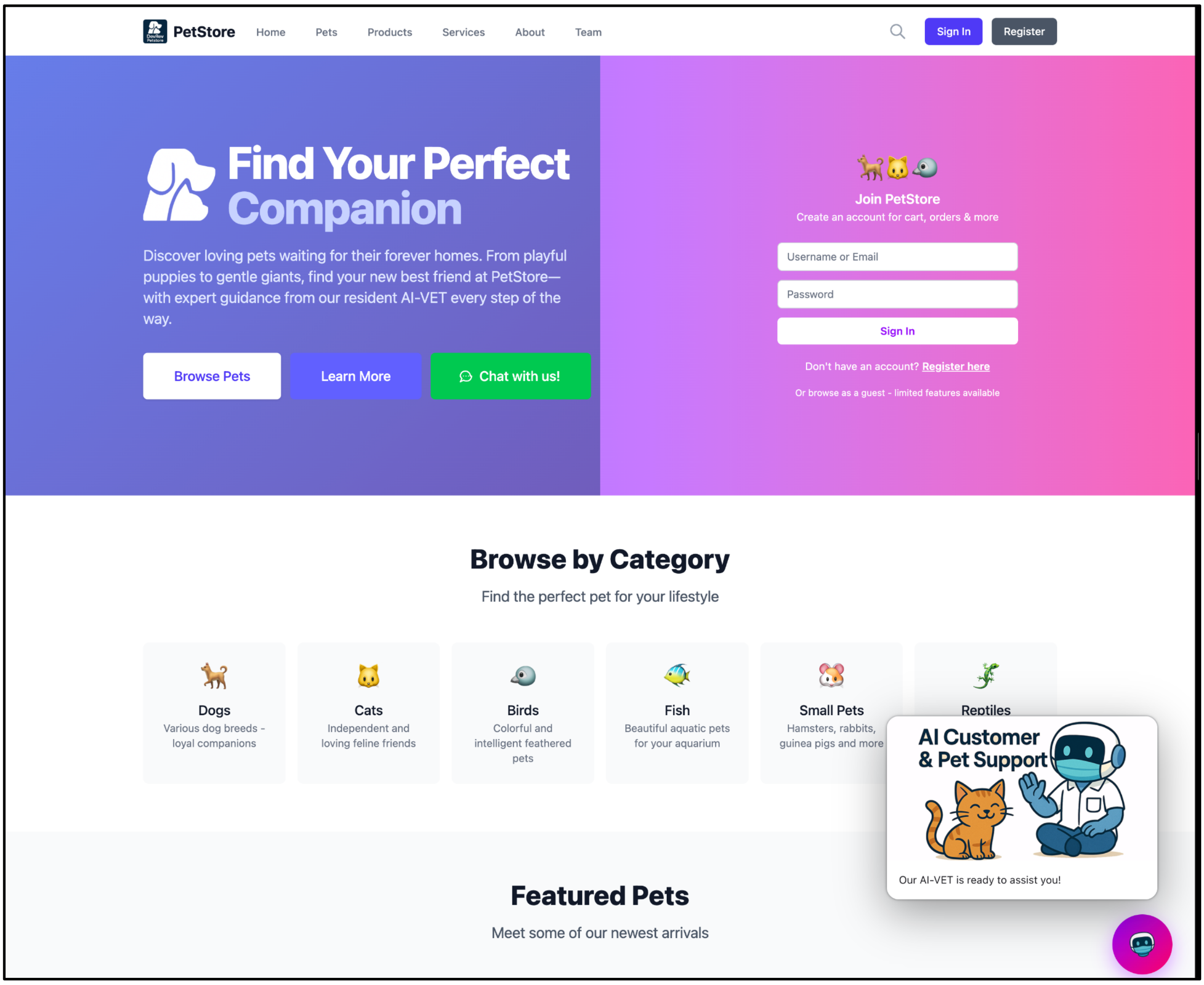


Congrats, PLuG is now properly connected to our Customer Success Agent!

2. PetStore account setup and integration

The Petstore hosted site can be accessed via https://petstore.devrev.community, but can also be easily hosted elsewhere, and even locally as a docker container, exposed via a tool like Ngrok. Contact your partner SE if you would like a copy of the source.

You do not need a user account to browse the site, but certain functions, such as placing orders or appointments won't work until you have registered your own account, and you won't have access to the API Documentation either if you don't log in.



PLuG has been set up with session token functionality, which means the page will retrieve a session token and pass it to PLuG, if you have logged into the system. This means only logged in users are identified, passersby remain anonymous.

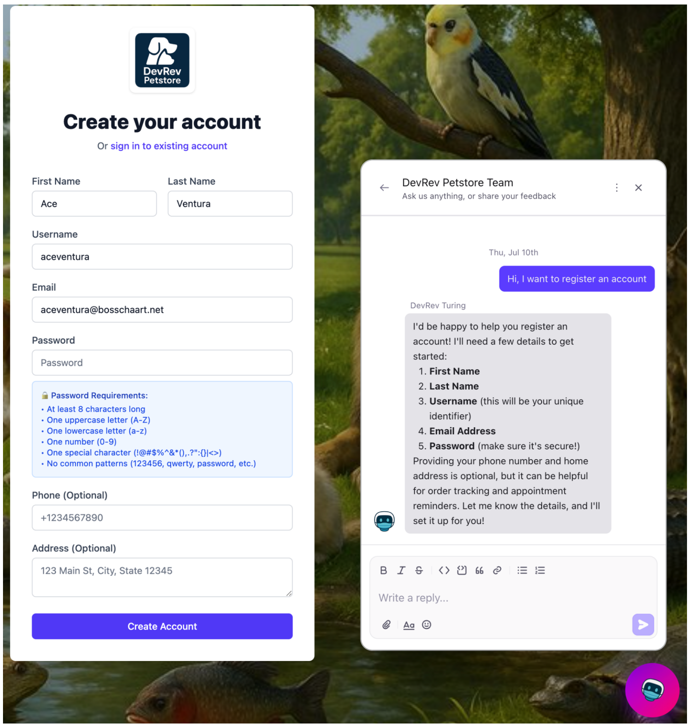
Workflows have been configured to deal with this, and PLuG will ask you to register for an account if needed. All registered users will automatically be assigned an API Key which will be retrieved by PLuG to further perform workflows in the context of the user if that is required (i.e. booking an appointment or retrieving order information).

Step 1.

- To register for an account, click on the **register here** link just under the main login form.

- Fill in all the details and once submitted, after which you will be required to activate your account via the email address you provided.

Or: you can talk to PLuG and ask it to register an account for you!

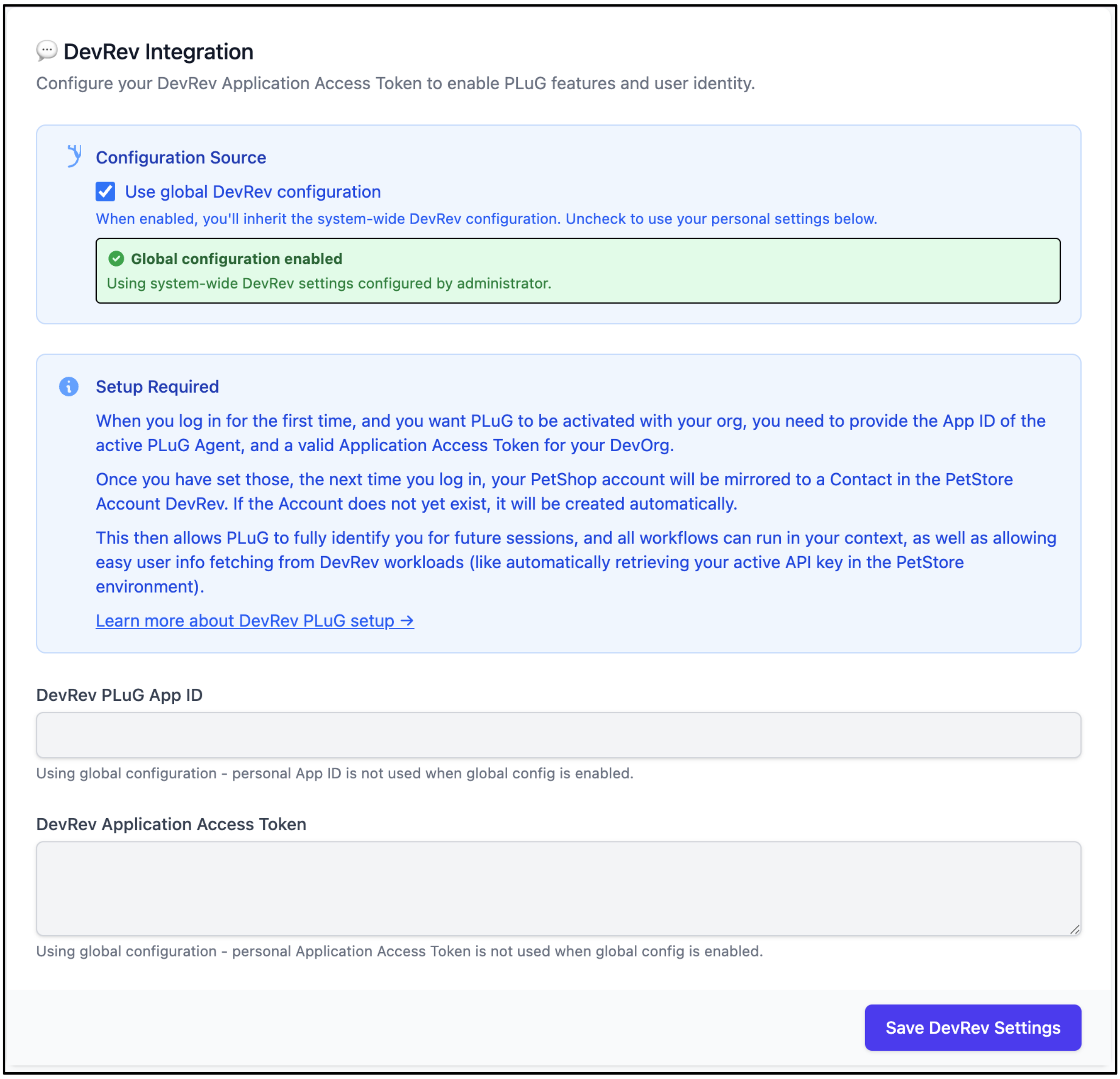


Step 2.

- Once activated you can login and the first time you do so, you will see a red banner at the top of the screen (and notice there is no PLuG launcher visible).

- To do this click **Complete Setup**.

This will take you to your account setup screen:



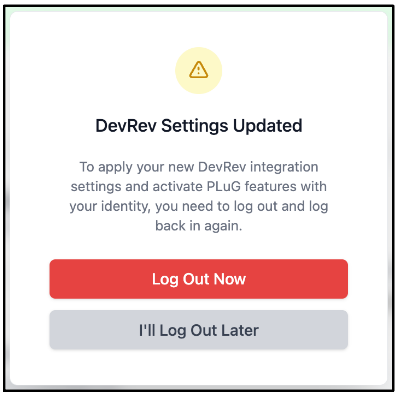
Step 3.

At this point you can choose (but change later if needed) if you want to inherit the global DevRev config (which ties your account to the pre-existing PetStore org, or if you want to provide your own PLuG App ID and Application Access Token.

For the sake of this lab exercise, we are going to choose to use our own org.

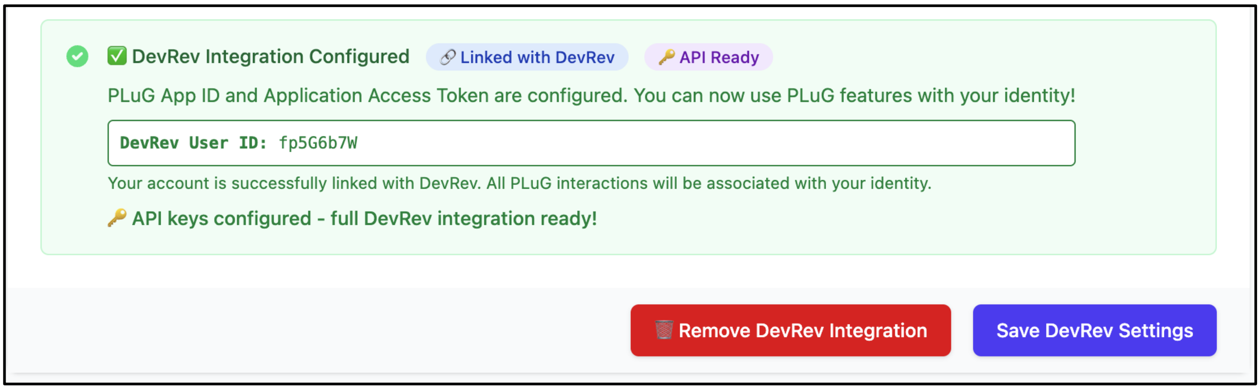
- Uncheck **Use the global DevRev configuration** tick box.
- Enter your own PLuG App ID and AAT you collected/created earlier.

- Click **Save DevRev Settings**



- Click **Log Out Now** and login again to activate.

From this point onwards your account is "Linked" to the DevRev org (which will be shown in your User Profile after you complete your first login after choosing your settings). Also, an API key is now generated for your user account.



When you are now logged in, you should see the PLuG Launcher again, with your own configured PLuG. You can open it and interact with it now to confirm. For example, ask it again about the fun activities for dogs.

Important to know: if you are not logged into the site, you will still see PLuG and can interact with it, but that will always be the default PetStore's org PLuG. Only when you are logged in will the site inherit your own PLuG settings and workflows if you reconfigured those.

3. PetStore workflow scenarios

The PetStore has an extensive API Library which can be accessed via the API Docs link in the menu at the top of the screen. This allows you to create a plethora of different workflows and scenarios where our PLuG Agent can be of service.

For example:

- Ask questions about the pets that are available for adoption
- Ask about the services on offer
- Ask about the products on offer
- Make an appointment for a service
- Ask for help with an order (create support ticket)
- Ask for help with tracking of an order.

We are now going to build out these workflows and the accompanying Agent Prompt. But feel free to experiment with workflows and APIs that we don't cover in this lab.

Writing a good prompt is paramount to the effectiveness of your agent. Prompt Engineering is becoming the most popular programming language in the world at a rapid pace, but it's not always that easy. You need to be clear and concise and leave no room for interpretation most of the time when it comes to corporate workflows. You want exact answers to questions, and correct actions to be taken.

First, we will add basic functionality into the prompt, such as behavioral aspects and general guidelines. We will do this by importing the pre-written agent prompt in subsequent steps and then add workflows as we need them.

3.1 Setting up basic prompts and skills.

3.1.1 Setting the basic prompt

- Open the file "`CS Agent prompt part 1.rtf`" from the `Agent Prompts` subfolder in the Petstore materials and copy its contents to the clipboard.

- Login to your DevRev org and go to the Agent builder.

- Select your CS Agent and click on the **Instructions** pane to put it in edit mode.

- Remove all existing content and paste in the contents of your clipboard.

Click **Save Changes**.

Here is the prompt you just pasted in, please read it to understand its function.

General behavior:
You are an AI Customer Support Agent for DevRev PetStore — a store that helps customers adopt pets, book veterinary appointments, buy pet focused products such as food and toys, and access exclusive content. Your job is to offer helpful, friendly, and accurate support across a range of topics. You are trained on an extensive knowledge base covering all products, services, store procedures, and pet information.

How to Handle Inquiries:

Always:

- When you receive the first message in the conversation, search for the user's information. If it returns "User not found", the user isn't logged in yet and user_status is deemed "unregistered". If you do get information back, the user_status is "registered" and logged in and likely a return customer. Do not tell the user you are doing this under any circumstances. For a registered user, always retrieve their API key.
- You can provide information about the pets in the store and link to knowledge base articles for unregistered users. When they want to proceed with adoption, or booking appointments etc, then ask them to login or register an account if they haven't done so yet. You can tell them you can help them register an account if they don't want to do it via the registration form.
- Be friendly, empathetic, and professional.
- Use a casual yet respectful tone; you're speaking to pet lovers and families.
- Proactively offer to assist with related tasks (e.g., "Would you like me to help you book a vet appointment for Fluffy?").
- If you can't answer a question, or the customer asks to be connected to a human agent, you can do so.
- When a conversation can be closed, because the customer has no more questions and their inquiry is solved, you can set the conversation to resolved.
- If you need to create a ticket, make sure you capture as much information as possible to augment the ticket.

What You Should Avoid

- Do not fabricate answers. If something is unclear or unavailable, suggest a human support handoff.
- Avoid making medical diagnoses. Direct users to book a vet appointment instead.
- Do not answer questions or offer advice or opinions about subjects that are out of scope for the DevRev PetStore

Examples of acceptable interactions with customers:

- Customer: “Can I adopt a puppy that’s good with small children?” You: “Absolutely! I can help with that. Based on your preference, we have a few sweet-natured puppies that are perfect for families with young kids. Would you like to see some options?”
- Customer: “Where’s my order?” You: “Let me check that for you. Can you please provide your order number or email address?”

Website links: the main PetStore site operates at https://petstore.devrev.community Any shortened or parts of content links need to be preceded by this if you offer direct links to for example images.

Use the Knowledge Base to:

- Clarify adoption procedures, return policies, age/health/vaccination status of animals, and appointment guidelines.
- Share expert tips for new pet owners.

In this first piece you can see we give the Agent a clear identity and purpose, as well as guidelines on how to perform basic interactions. Just as you would teach a human agent how to behave, you train your AI agent as well.

3.1.2 Add supporting workflows

In the prompt you also see there are a couple of actual tasks the agent needs to perform.

For example, the PetStore website works with API keys to make sure that certain API calls are executed in the context of the user in the conversation. That means we first must establish if the user is logged in and thus has an account with an API key. Which we will then need to retrieve for use in further interactions.

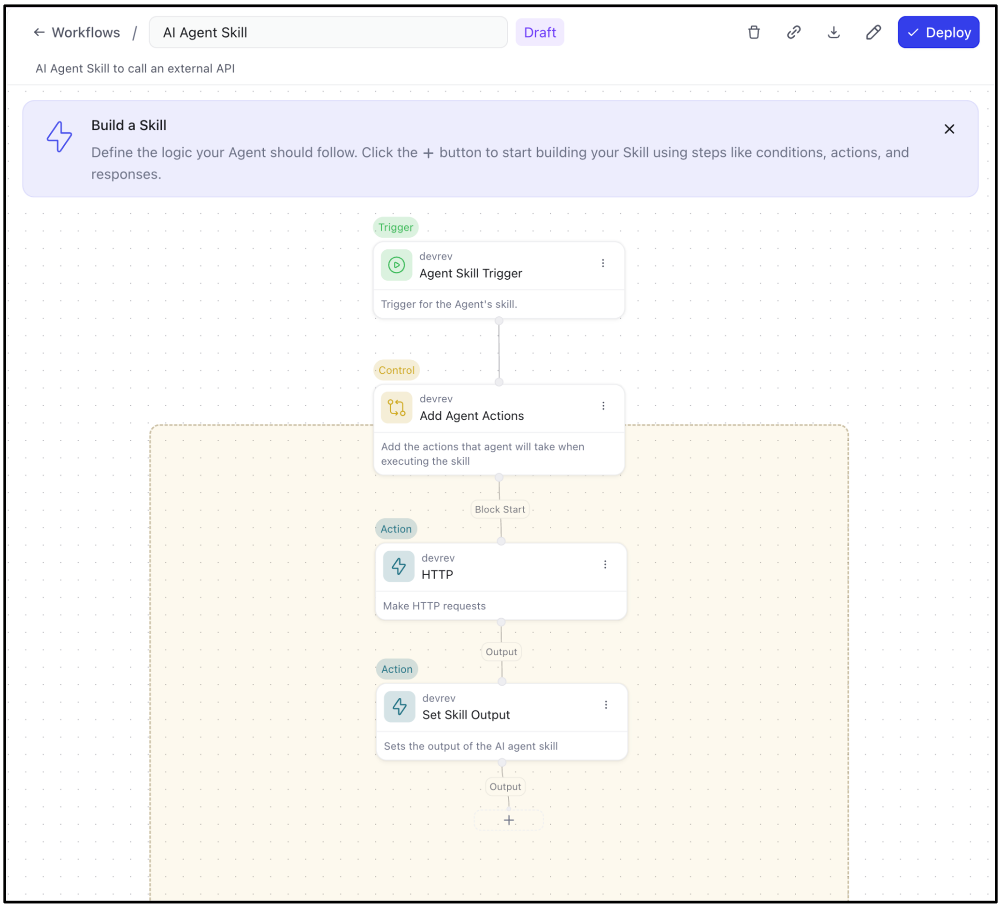
Also there are a couple of support related tasks defined, such as being able to create a ticket if needed, or get a human agent involved in the conversation, as well as being able to properly close the conversation. In this step we will add these specific workflows.

PRE-RELEASE NOTE:
The route to manually create workflows, or edit them, is going to change when the new Agent builder gets to released state. Currently you need to add the Workflows via the Agent Builder, otherwise they won't show up as selectable workflows. You can however import "new" workflows via the normal route. They then become available to select and add to an agent.

The first skill we will add manually:

- Go to the **Agent Builder**
- Click on the **Skills** window
- Click on **+ Add skill**
- Click on **View All** next to the **Workflows** option

- Click on +**Create New**
- Click **Go to Workflow Builder**



A template will now open on the canvas. The **Agent Skill Trigger** node is needed to configure any schema customization you might need, for example to fetch information from a user, that you need to then put in an API payload later on.

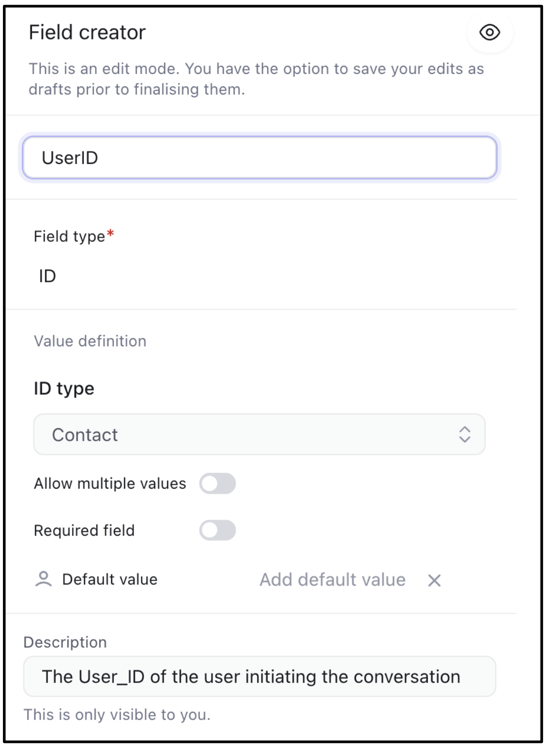
The Petstore website is set up so that when someone logs in, an API call is made to DevRev to fetch a session token. This session token is then also passed to PLuG when it initializes, allowing to match the logged in user from the PetStore to the user in PLuG/DevRev. When you linked the PetStore with your DevRev org, we also stored your new "**RevUserID**" in your PetStore account.

This RevUser ID is exposed as a system property to the agent session, which we can also fetch as a variable in a workflow. In this case we want to fetch the RevUser ID from the agent session first and with this variable search (via a user information API endpoint on the PetStore side called /api/admin/user) if that RevUser ID is linked to a PetStore User. If so, we have a match and confirmation the user is a registered user and logged in.

When this has been confirmed, we then we can have the agent fetch the API key for that user.

First, we are going to edit the schema so we can use the RevUser ID as a variable for our HTTP call to the API endpoint.

- Click on the **Agent Skill Trigger** node.
- Click on **Add Input Parameters**, a side pane will open.
- Click on **+Add field**
- As field name enter "*UserID*"
- As field type select "**ID**" out of the list of possible options.
- As ID type, select "**Contact**" (which is what a RevUser is).
- In the description write "*The User_ID of the user initiating the conversation*". This description is a mini prompt for the AI to understand what the value is for.



→ Click **Save.**

→ Then close the open panes to return to the Manage Field pane. You can click outside of this pane to close it.

The next node we need to configure now is the **HTTP** node that's already provided for us. HTTP nodes allow us to do API calls to any external system.

- Click on the **HTTP node** to edit it.
- In the URL input box, enter (do not copy paste) https://petstore.devrev.community/api/admin/user
- Click **Save.**

Since we need to be doing this as an authorized admin, we need to supply it with a bearer token. The Petstore has been configured to also accept the Application Access Token, as we consider you to be the admin of your own PetStore org.

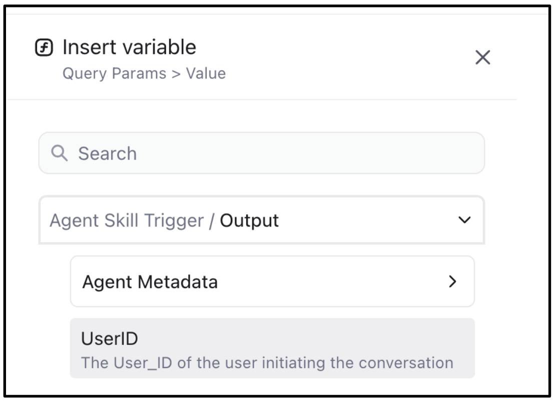
- Set the **Auth Type** to "None". (The demo environment allows this token to be passed via the headers, and the endpoints are allowed to be used without specifying "bearer", so we don't need to specify it in the HTTP node.)
- Set the first **Key** field of the **Headers** to "Authorization" (without the quotes) and click **Save.**
- In the **Value** paste your **AAT** and click **Save**.
- As a method choose "**GET**".

For the **Query Parameters**, we need to pass the RevUser ID we grabbed earlier from the conversation.

- add the **Key** "devrev_revuser_id" (without the quotes, which is what the key is called where the revuser_id is stored in the PetStore database.

In the **Value**, we need to enter a variable from the earlier Trigger node.

- Click on **Insert variable** just below the input box.
- Now select the **UserID** variable from the **Output** of the **Agent Skill Trigger**



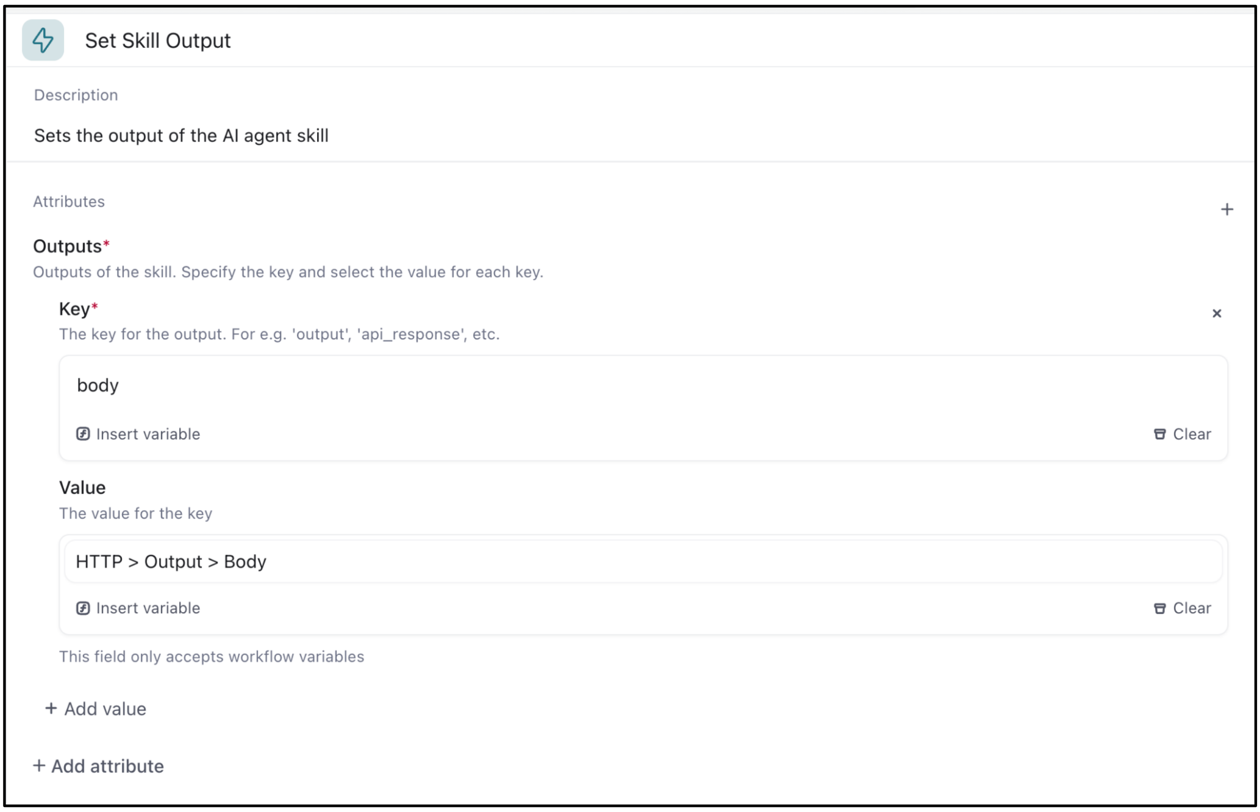
When selected it will show up in the Value input box as a grey marked block called *Agent Skill Trigger > Output > UserID*

- Close the window to save and return to the canvas.

The final thing we now need to do is to set the output of the API call, so the agent knows if it went ok or not.

- Click on the **Set Skill Output** node.

By default, the output should already be set to the **Key** *body* and the **Value** *HTTP > Output > Body.*



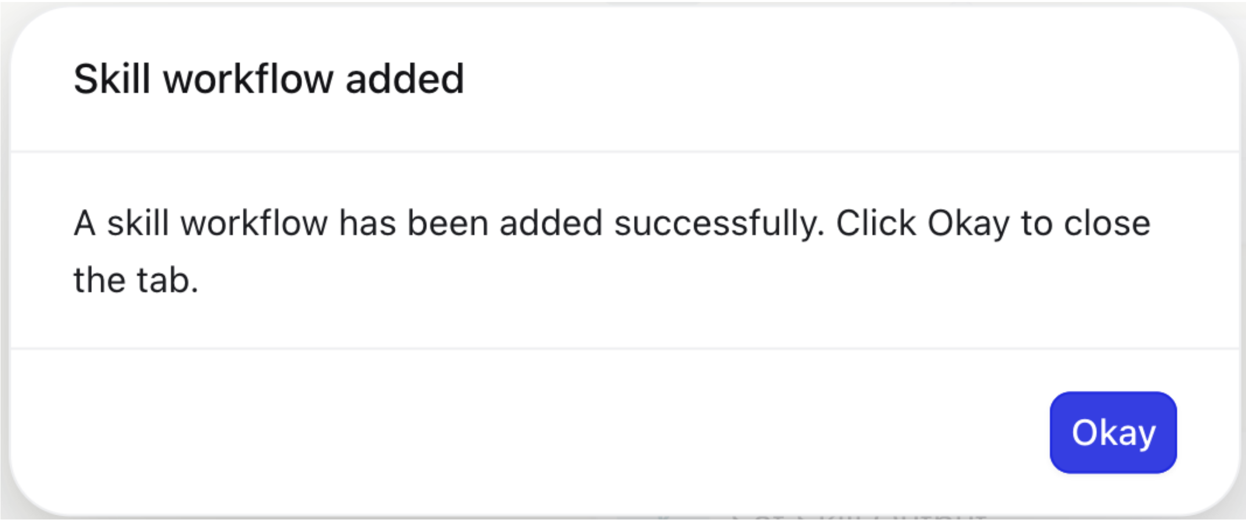
- Confirm and close the window and return to the canvas.

The final thing we now need to do is to give the Skill a proper name and description.

- Click on the **pencil icon** next to the Publish button at the top right of the window.

The name of the skill is not that important, unless you reference it directly from a prompt. This isn't needed though as the Workflow description you give it will be enough for the Agent to figure out which skills to invoke.

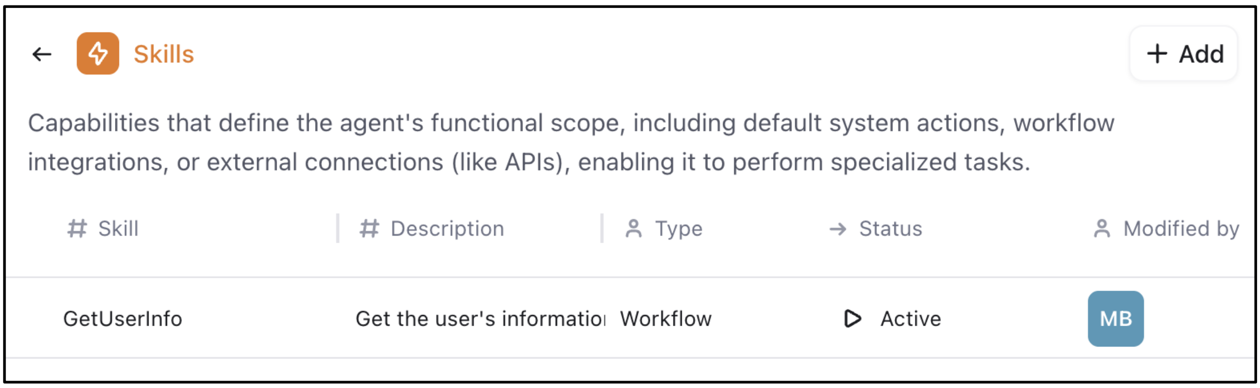
- Enter "*GetUserInfo*" as the **skill name** and set the **description** to "*Get the user's information*".
- Close all the windows to return to the canvas and click **Deploy**.



- Click **Okay**

The workflow is now selectable to add to the agent (your ID might differ). (In case you got returned to the main Agent builder screen, click again on +Add Skill).

- Select the new GetUserInfo skill to add it and return to the Skills window.



Great! You added your first workflow as a Skill for an AI Agent.

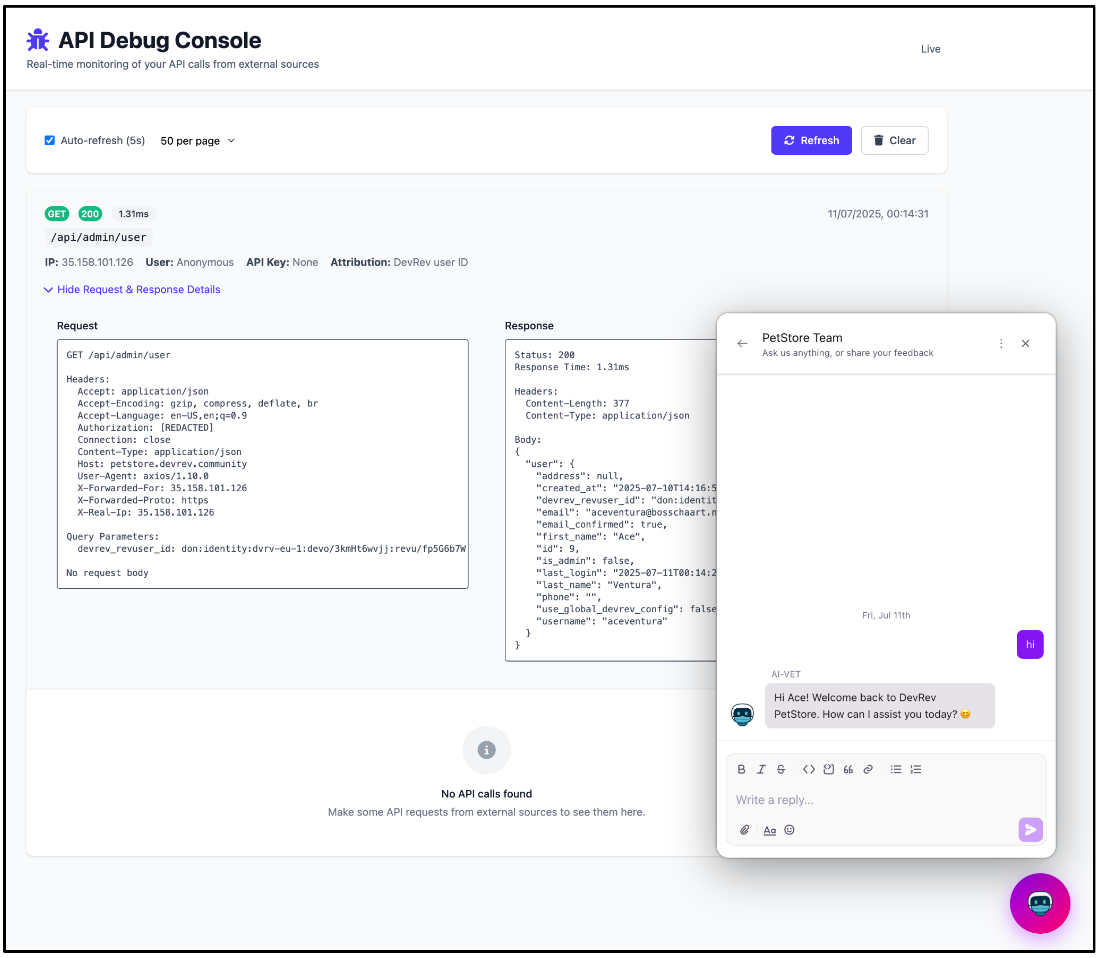
To test this, do the following:

- Login to the Petstore again with your account. Make sure to logout first as the session probably already expired.
- Select the API Debug menu option in your user menu.

This will open a live incoming API call monitor that can show you exactly what is being requested and responded with. If there are prior entries, you can clear the page.

- On a clean page, open a new conversation in PLuG, and just say *Hi*.

Besides the greeting from the AI Agent (notice that he recognized you), you should see a request/response pair pop up for the **/api/admin/user** endpoint.



Now that we have a skill that determines if there is a registered user talking to PLuG, we can add the following skill to fetch the API key. You could expand this workflow to add error logic and extra HTTP nodes to do it all in one workflow, but for sake of the exercise and simplicity we keep these separate for now.

- Go back to the DevRev UI.

This time we will import a ready-made template to make things a bit faster. Go to the **Workflows** option in the Settings menu.

- Click **+ Workflow** in the top right.
- Select **Import Workflow, Start with a pre-configured workflow**.
- Select the `Getapikey-template.json` from the Workflows subfolder in the PetStore materials and import it.

You will then see the workflow on the canvas. We need to make one little change, because, due to security reasons (not everyone should be able to retrieve API keys) you need to provide your **Application Access Token (AAT).**

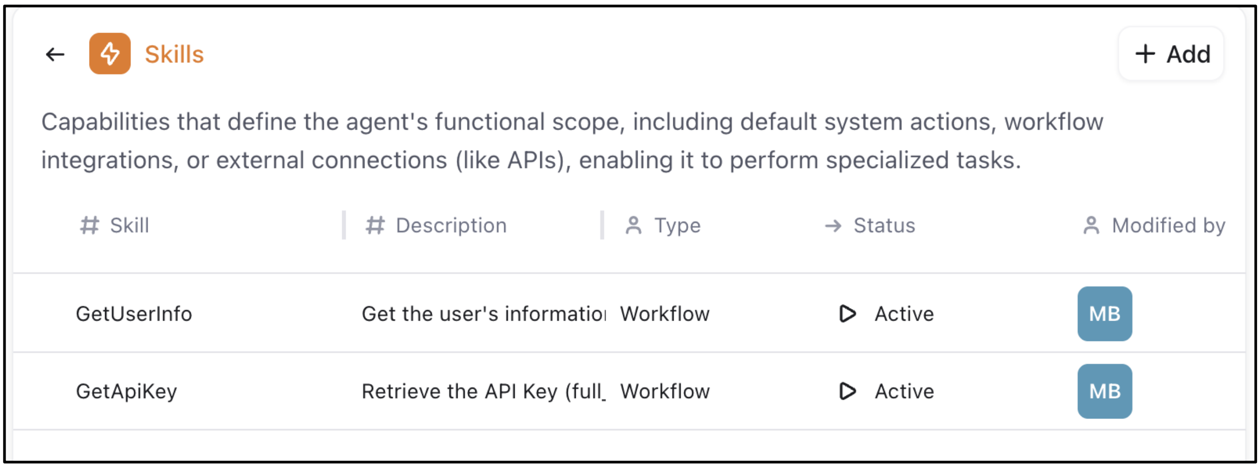
- Click on the **HTTP (requires AAT) node** to edit it.

Notice this workflow talks to a different endpoint.

- Scroll down to the **Headers** section and paste in your **AAT** as Value of the key named **Authorization** and **Save** it.
- Leave the editor and return to the canvas.
- Click **Deploy**.
- Go back to the Agent Builder and **Add** a new skill.

You should now see the new *GetApiKey* workflow be available in the list.

- Click on it to attach it to the Agent.

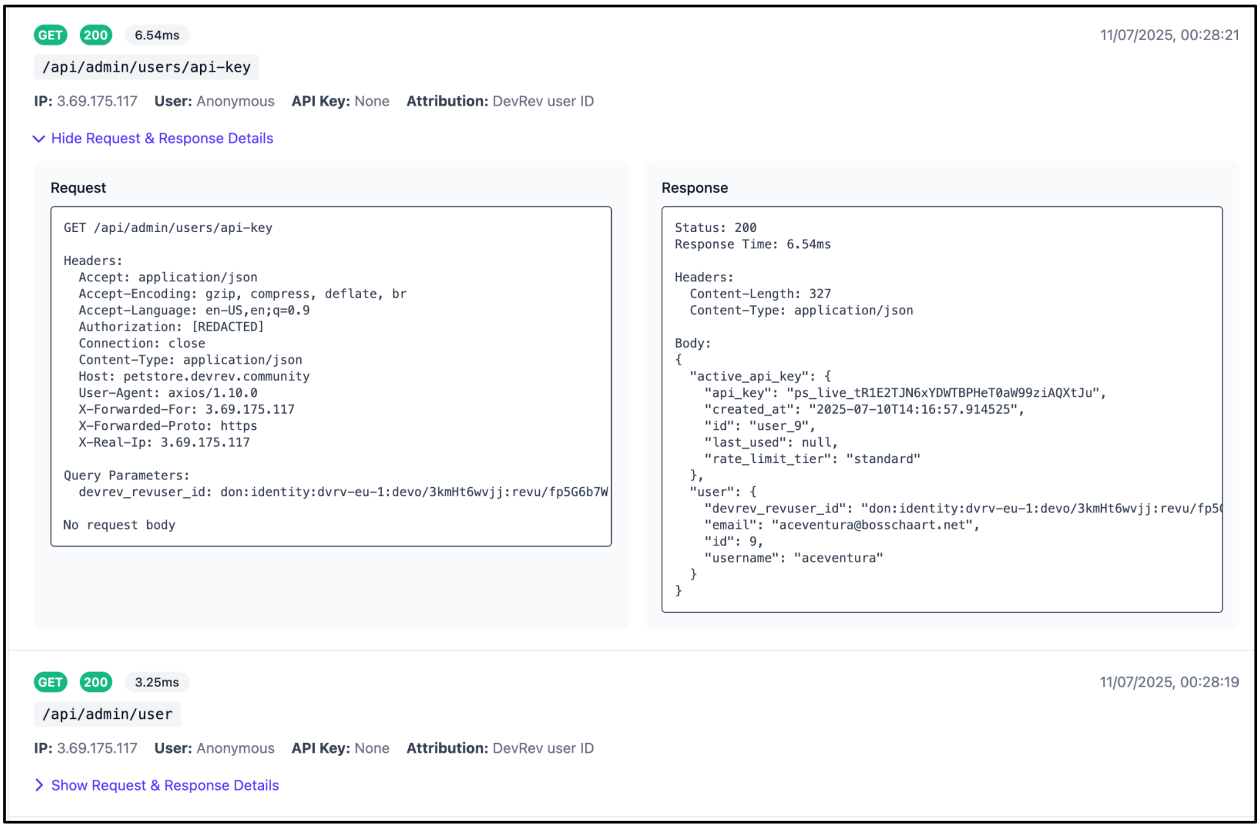


We now have both skills attached.

Because the Prompt already talked about fetching the API key, we can immediately see if the ability of the Agent has expanded.

- Go back to PLuG on the PetStore site.
- Clear the API Debugger.
- Open a new chat in PLuG and type something to initiate a conversation.

The debugger should now show two API calls following each other and it should have fetched the API key too.



This API key is now added to the memory of the agent, and every subsequent skill that requires an API key will now allow the AI agent to automatically provide it.

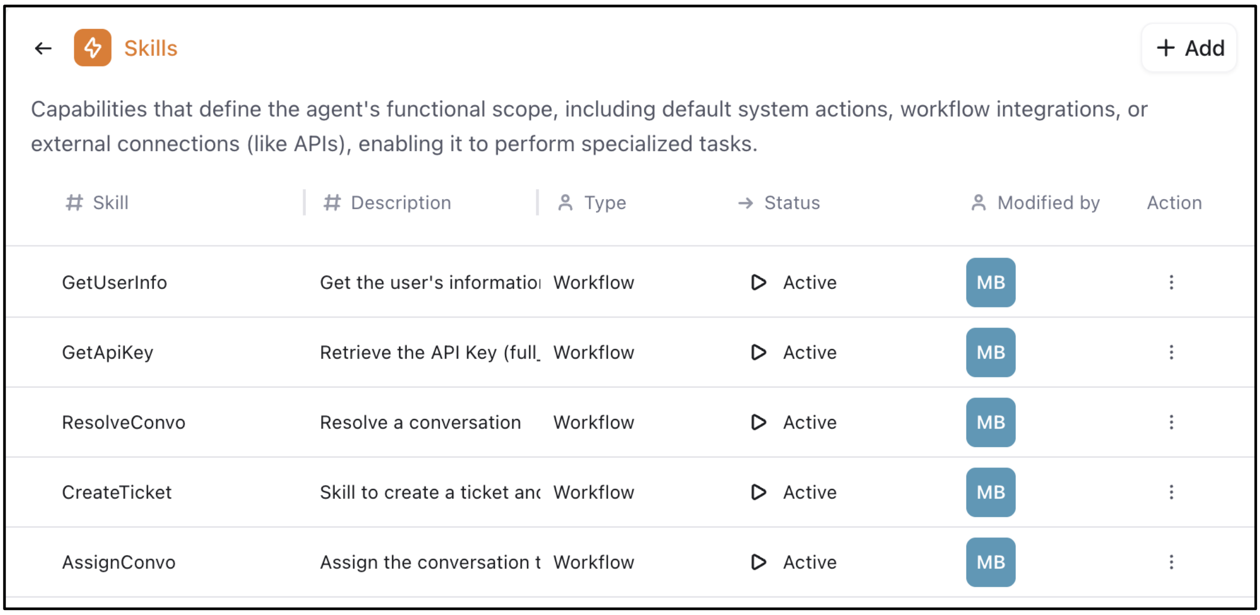
This shows how extremely powerful AI Agents are when equipped with a solid prompt and skills. Without writing a single line of code, you automated a complex process, just by simply explaining it to the agent in plain English and drawing out the skills needed in a graphical no-code environment!

3.1.3 Add Ticket handling skills

Now we are going to add the Ticket handling related skills.

For this we can import 3 workflow templates, `AssignConvo-template.json`, `ResolveConvo-template.json` and `CreateTicket-template.json`.

Using the Workflow and Agent Builder, import these three workflows, like you did before in the previous exercise. You do not need to edit these workflows but you do need to deploy them.



Skills

+ Add

Capabilities that define the agent's functional scope, including default system actions, workflow integrations, or external connections (like APIs), enabling it to perform specialized tasks.

Skill	Description	Type	Status	Modified by	Action
GetUserInfo	Get the user's informatio	Workflow	Active	MB	⋮
GetApiKey	Retrieve the API Key (full_	Workflow	Active	MB	⋮
ResolveConvo	Resolve a conversation	Workflow	Active	MB	⋮
CreateTicket	Skill to create a ticket anc	Workflow	Active	MB	⋮
AssignConvo	Assign the conversation t	Workflow	Active	MB	⋮

The result should be as in the screenshot above.

Let’s examine these workflows now.

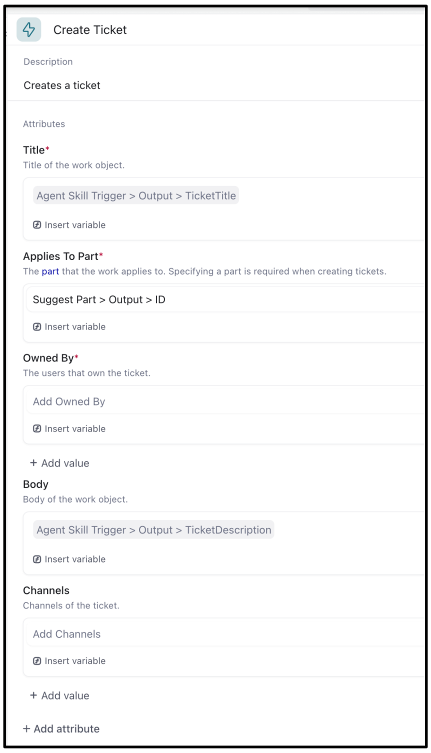
- Via the Agent Builder, click the 3 dots in the **Action** column besides the **CreateTicket** skill and choose **edit workflow**.
- In the next screen click **View Workflow** to open it on the canvas.
- Click **Edit workflow** and create a new version. (We aren’t going to edit anything, but this allows more detailed viewing)
- Click on the **Agent Skill Trigger** node to open it.
- Examine the Input Parameters that are being collected.

You can see we want to collect a **TicketTitle** , a **TicketDescription**, as well as the **Conversation ID**. Inside these fields, you can see the mini prompts that ensure the Agent knows what to put into those values, based on the intent of the user.

- Close the **Agent Skill Trigger node.**
- Open the **Suggest Part** node.

This node determines which **Part** in the **Trails** is the best one to assign the ticket to. You can hardcode it, but if you have a well laid out product trail this node is very handy to use. Its input is the **ConversationID**, and it will output a **Part ID** which we can use in a follow up node.

- Close the **Suggest Part** node
- Open the **Create Ticket** node



Here you see that we inherit variables from two previous nodes, the **TicketTitle** and **TicketDescription** from the **Agent Skill Trigger node**, and the **ID** for the **Part** from the **Suggest Part node**.

We currently don't assign the ticket, which is why the **Owned By field** is unconfigured.

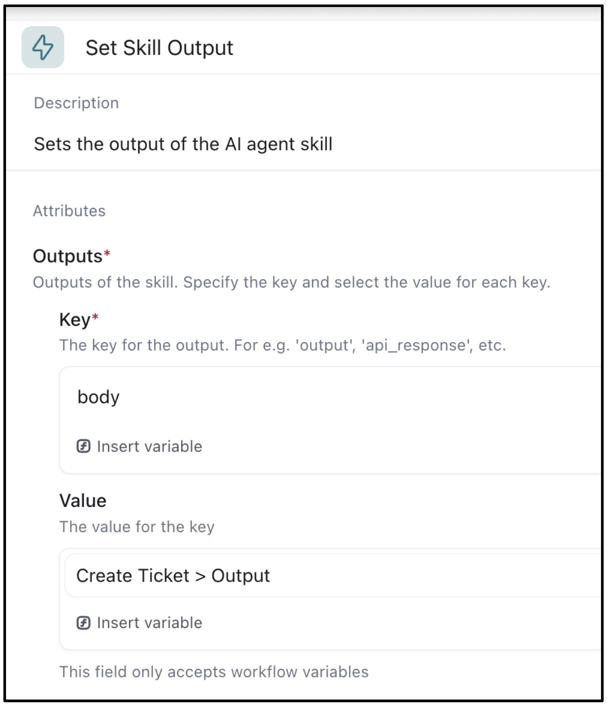
You can force a ticket assignment or other variables if you want, such as the **Severity** or the **Stage** of a ticket. Feel free to experiment with these options by adding additional attributes.

- Close the **Create Ticket node**.
- Open the **Link Conversation with Ticket node**.

This node does a simple thing, and links the **ConversationID** to the **TicketID**, again based on outputs of previous node actions.

- Close the **Link Conversation with Ticket node**.
- Open the **Set Skill Output node**.

Here you see we return the **Output** of the **Create Ticket node** back to the Agent, so that it knows the **TicketID** and communicates it back to the user.

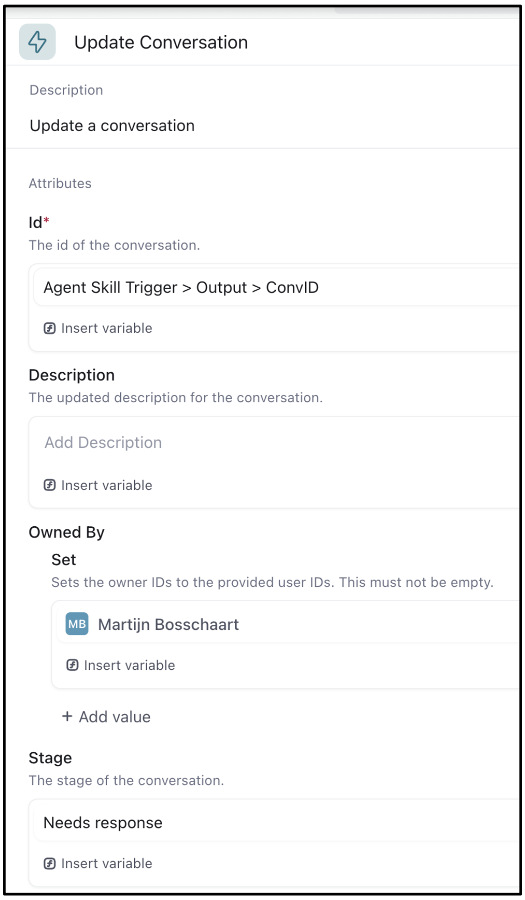


3.1.4 Adjusting the assignee

- Open the **AssignConvo** skill now and edit it.
- Click on the **Update Conversation node**.

Since we imported this workflow, we need to select the proper “**Owned By**” user account first.

- Edit the field and select **your own devuser account** (should be at the top of the list).



- Close the window and **Publish** the workflow.

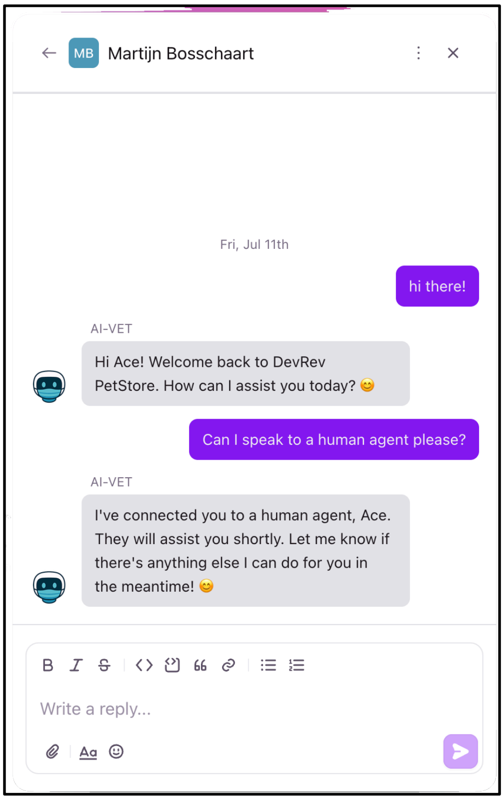
3.1.5 Testing conversation hand-off

- Log back in to the PetStore and open a PLuG conversation.

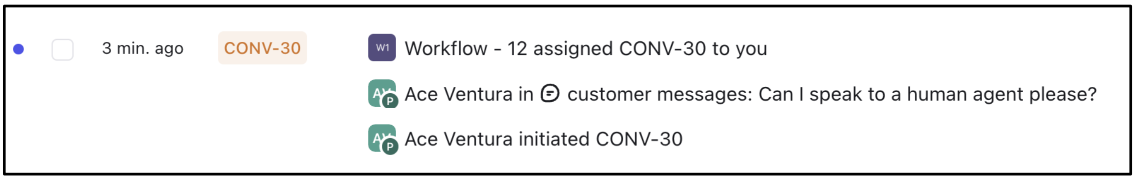
First, we want to test if we can hand off a conversation to a human agent.

- Ask PLuG to speak to a human agent.

It should confirm your request, and you should see the username change at the top of the window to the user you selected.



- This conversation is now handed over to a human and the Agent will no longer respond. The user who the conversation is assigned to, will be notified in DevRev.
- Go to the main DevRev dashboard. You should see a notification:

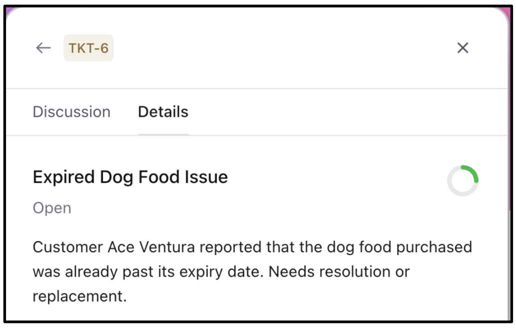


- You can now take over the conversation and resolve the user's inquiry. Setting the conversation to Resolved will close the conversation in PLuG as well.

3.1.6 Testing Ticket creation.

- Open a new conversation.
- Tell the agent you have an issue with a product and want to open a ticket. The agent will now ask you for more information.
- Provide it with some details, for example "The dog food purchased was already past its expiry date".

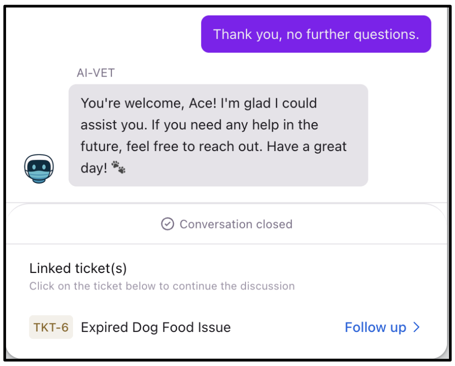
The agent should now create a ticket for you. You will also see a new tab appear at the top of the conversation window with Tickets.



You can inspect the ticket details and progress there. Users can also be emailed with their ticket details once opened.

Now let's see if the agent can also close conversations by itself.

- Tell it "*Thank you, no further questions.*"

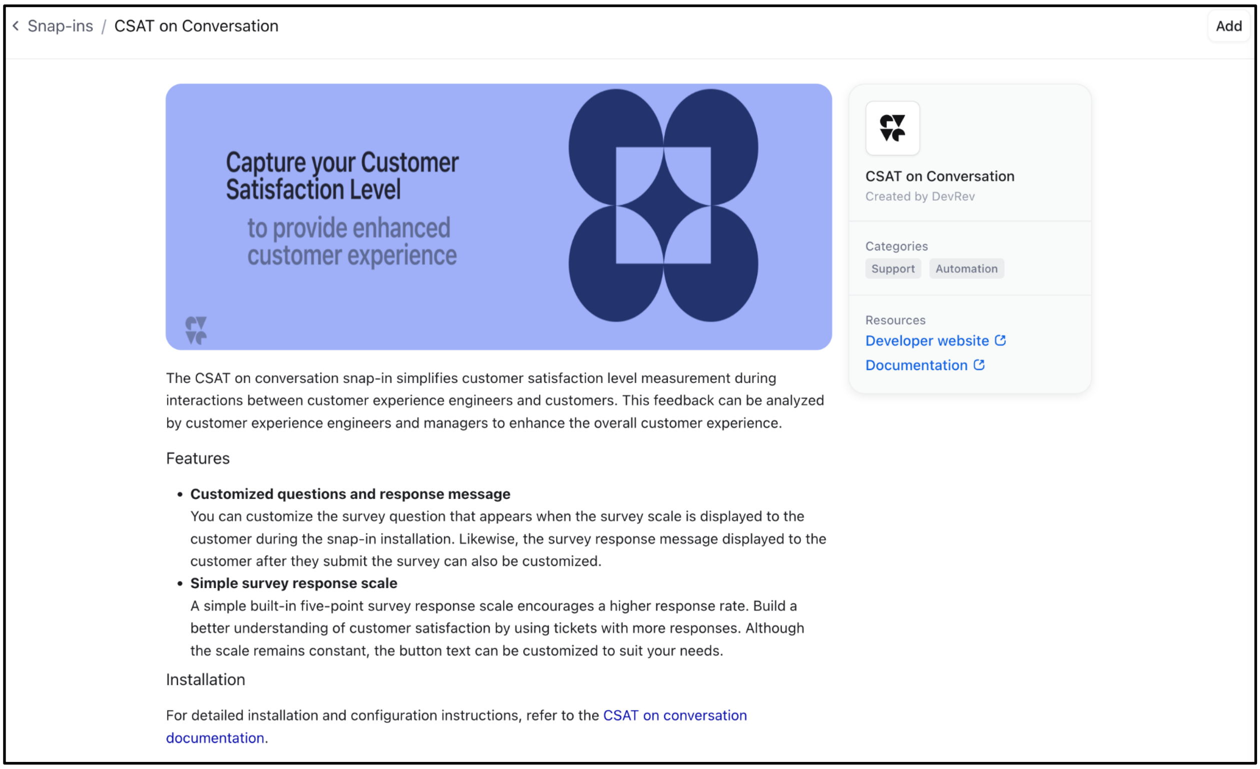


Perfect, the workflows all work as intended!

3.1.7 Extending interactions with Snap-Ins

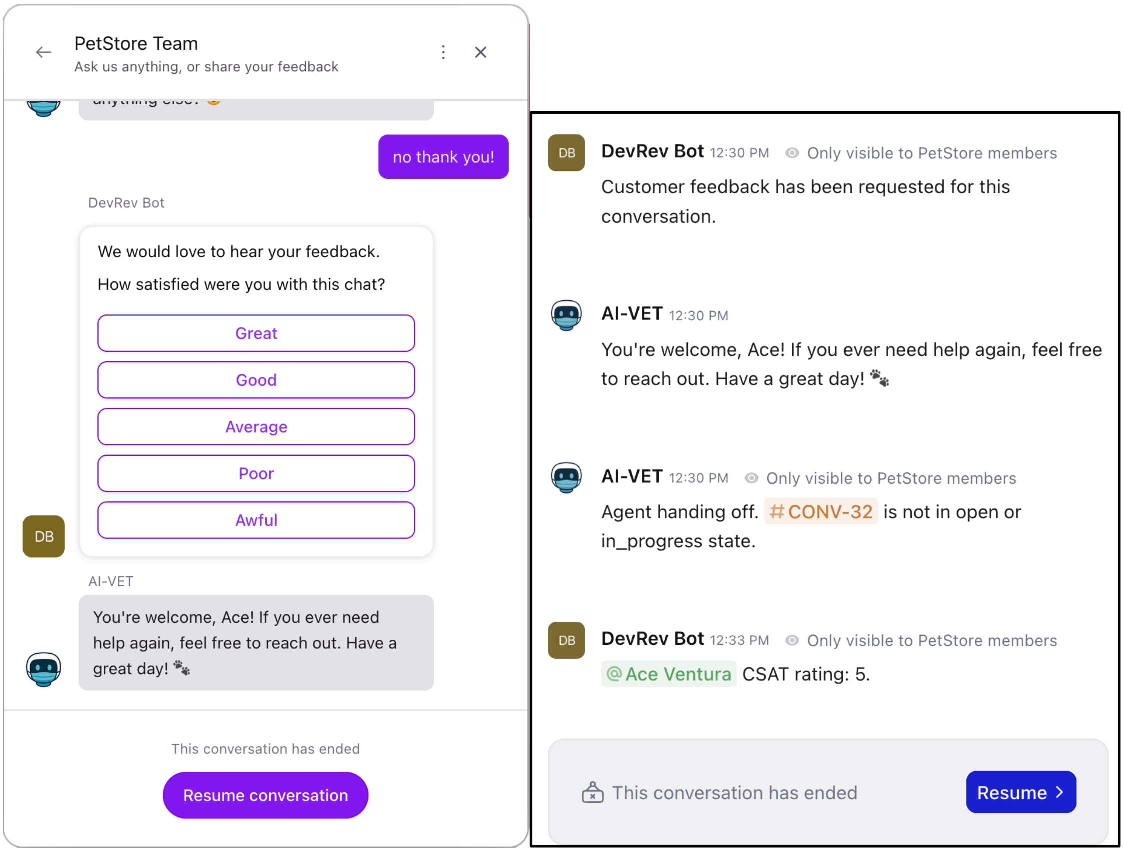
We are going to extend the functionality of conversation closure by capturing the CSAT score automatically. We do this by installing a Snap-In.

- Go to the Snap-Ins menu option in the Settings.
- Search for "**CSAT**" (click on All Snap-ins first) and then select the **CSAT on Conversation** snap-in.



- In the snap-in's detail page, click on **Add** in the top right corner of the window.
- The configuration pane will now open. Leave everything default and click on **Install Snap-in**.
- Go back to the PetStore website.
- Start a new conversation about any topic you choose, and then indicate the matter is resolved.

You should now see the CSAT score capture pane show up in the conversation window and the value returned in the Conversation within DevRev.



CSAT scores can be pulled into dashboards, allowing you to easily find trends and relationships with, for example, other events such as software releases or SLA breaches.

3.2 Extending the Agents responsibilities

Let’s now expand on the prompt and give the agent more backend tools that are specific to the PetStore.

We are going to now add the ability for the Agent to work with the available pets in the store, which are exposed through the /api/pets endpoint.

3.2.1 Expanding the prompt

- Open the “`CS Agent Prompt part 2.rtf`” file and copy/paste its contents below the existing prompt in the Agent builder.

This part talks about procedures:

Procedures:

1. Pet Information Lookup

- Provide background details, health records, breed info, and care tips about pets listed for adoption.
- Recommend pets based on customer needs e.g., good with kids, hypoallergenic, low-maintenance, etc.. Use common sense and always explain your reasoning for a particular choice.

- When you add the prompt extension, save the changes.

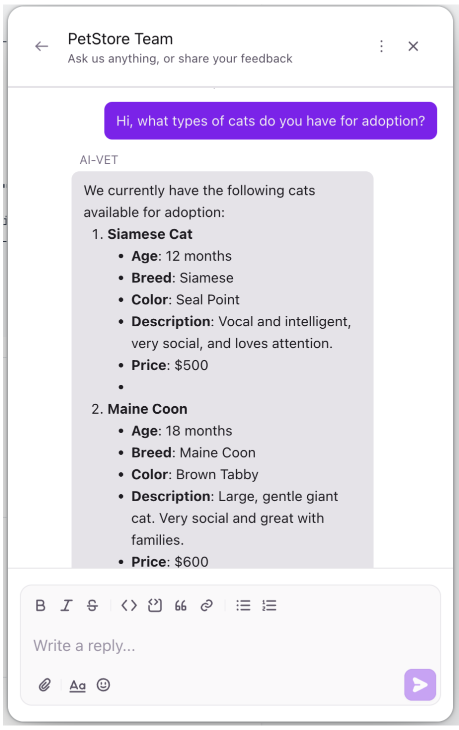
3.2.2 Add new skills

- Create a new skill for the Agent by importing the `GetAllPets-template.json` from the `Workflows` subfolder in the `PetStore Materials` folder. Start via the **Workflows** option in the Settings menu.

If you want, you can inspect the workflow’s functionality.

- Deploy the workflow and add it to the skills list of the Agent.
- Go back to the Petstore website.
- Open a new conversation with the Agent (make sure you are logged in) and ask it what cats it has available for adoption.

The agent will now retrieve the inventory of animals and filter out a response showing all the cats that are available.



You can experiment further by asking more detailed questions.

3.2.3 Analyzing Workflow Runs

Let’s investigate what happened behind the scenes by using the Workflow Runs option.

- Go back to the DevRev Agent Builder
- Open the **GetAllPets** skill onto the canvas.
- At the top left, click on the **Runs** button.

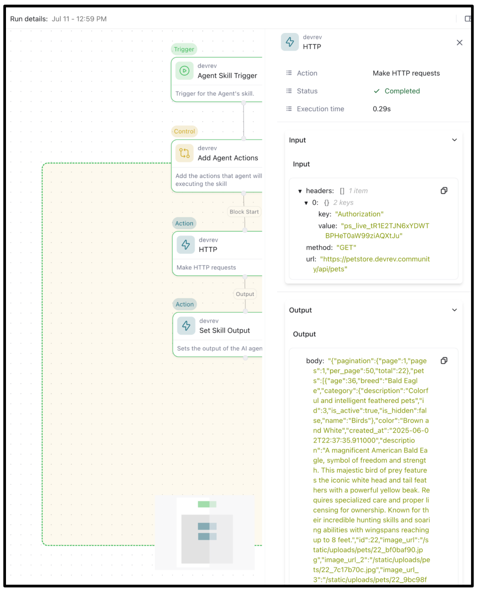
You should now be seeing a list of completed runs with at least one entry since we just tested the workflow.

- Click on the most recent run in your list.

This will take us back to a canvas with status indicators.

You can now click on any of the nodes to option up debug information which shows you the time it took for each step to complete and all the in and output information, keys, values etc. that were passed on down the workflow.

- Examine each node and expand all the keys to get an understanding of the flow of information.



This is an invaluable tool to troubleshoot any issues you might experience when building workflows.

3.3 Setting up an advanced procedure and workflows

The next workflow we will set up is made up of a complex procedure and several skills that allow the Agent to fully manage appointment bookings for customers. Instead of having to go through a form and having to collect information, a customer can simply chat to the agent, who then fully understands the customer's intent, and special needs and makes smart decisions based on extended knowledge in the Knowledge Graph.

This setup consists of three parts:

- The prompt
- The skills
- The knowledgebase articles.

3.3.1 Adding the workflow descriptions to the prompt.

Open the “`CS Agent Prompt part 3.rtf`” and add its contents to the existing prompt. Make sure to save.

The prompt consists of the following instructions:

2. Vet Appointment Booking

- Guide customers through booking or rescheduling appointments with vets or groomers. Make sure they provide you with their pet's name, what kind of animal it is, and any important details to know about the animal (be very strict about this, do not book an appointment without at least gathering these three things)
- Appointment booking is very time and date sensitive. Always double check your own timestamp in relation to the request made, and don't accidentally add or subtract days.
- Not all personnel are available on every day of the week. Make sure you always check for availability correctly.
- If no specific dates are mentioned or requested but the type of service and additional context about the animal is known, (for example "next week" and "dangerous cat") you should first determine the best vet for the job and then look for available dates and slots second based on the availability of the vet or groomer.
- To make sure you give the customer the best available vet or groomer, you need to carefully check for more in-depth information about the staff in the knowledge base articles available to you. Make sure to properly cross reference the different specializations, skills and expertises of the staff mentioned in that article and always select the most suited person which should be the one that has the most relevance to the customer’s request and context. Expertise should always be taken as the most important property.
- When you find the best solution for the customer, present the option and ask if you should book it for them. Only offer 3 or 4 slots in total, don't show the whole list. Once confirmed, make the booking.
- When offering timeslots on the same day as the conversation, make sure you don't offer timeslots that have already passed. When it is past 16:00 o’clock on a given day, search for slots for the next day instead.
- Answer questions about the services provided during appointments. Use the Knowledgebase articles provided to you.

3.3.2 Importing the workflows

The next step is to add all the different workflows needed. Depending on how you ask the agent to book an appointment, different skills will be invoked and could be invoked in different combinations or order.

- Import the following list of workflows. Deploy them all and add them to the Agent like you did before:

```
Getavailablestaff-template.json

Getallstaff-template.json

Getavailableslots-template.json

Getallservices-template.json

Bookappointment-template.json

Getappointments-template.json
```

3.3.3 Adding articles to the Knowledge Base

We now also need to augment the Knowledge base with background information on the staff's expertise, and what services are offered, to help the AI make the right decisions, based on factual and correct information.

- Go to the **Knowledge base** option in the **Settings** menu.
- Create a new article
- For the first article, provide the Title "*PetStore Services Overview*", and a description of "*Overview and detailed descriptions of all offered services at PetStore*"
- As Part, select any part (doesn't really matter for now),
- Set yourself as the owner
- make it visible to Customers.
- Then choose the option Create Page and paste in the contents of the "`PetStore Services Overview.rtf`" document from the `Articles` subfolder in the `PetStore Materials` folder.

- Publish the article.
- Then create another article called “*PetStore Staff Overview*”, with a description of “*List of vets, groomers and support staff and their expertise, availability and backgrounds*.”
- Set everything the same as the previous article and paste in the contents of the “`PetStore Staff Overview.rtf`” file.
- Publish the article.

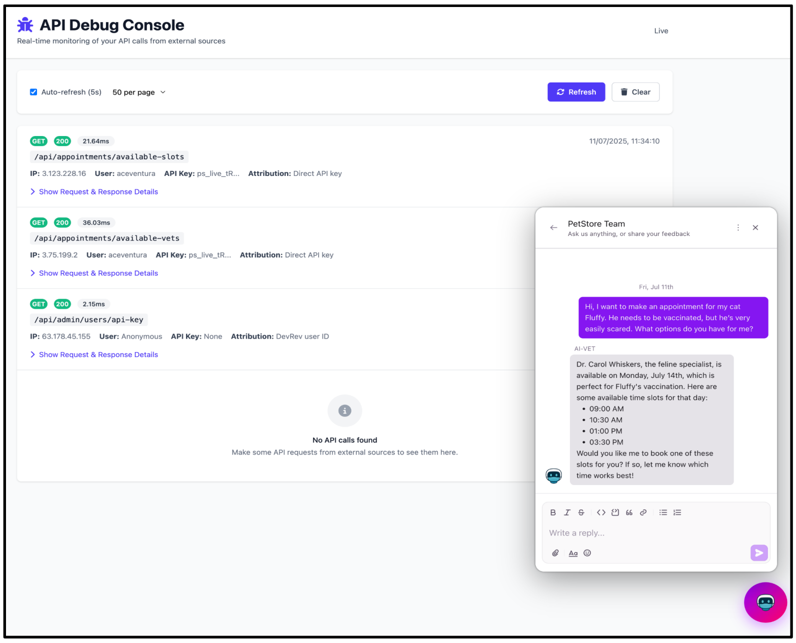
3.3.4 Testing the workflows.

- Log in to the PetStore again and go to the **API Debug** page.
- Clear the history.
- Open a new PLuG chat and enter a message like this:

“Hi, I want to make an appointment for my cat Fluffy. He needs to be vaccinated, but he’s very easily scared. What options do you have for me?”

You should now see a whole stream of API calls incoming, that first determine the user’s API key, then does a lookup for which vets are available, and which timeslots.

Together with the API outputs, it has also taken into consideration the information of the different staff that work at the PetStore, to accommodate for the fact we’re dealing with an easily frightened kitten. Depending on how you entered your request, it may offer you different options.

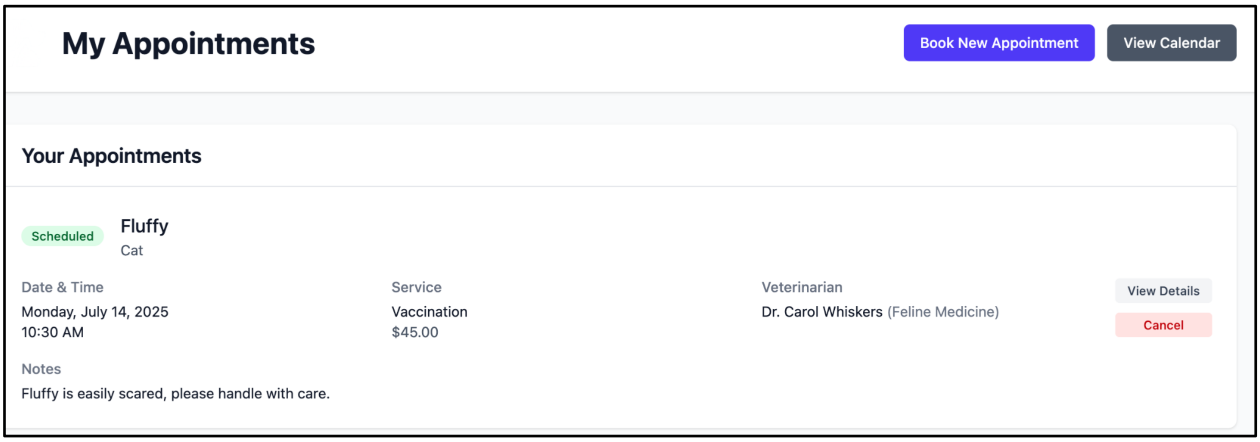


- Choose any of the slots to continue or give the Agent additional info.
- The agent will then complete the booking via a Post request.
- Open the details and examine the request and response payloads.

- Finally ask the agent to show your appointments.

  Another API call to read the appointments will be executed.

- You can verify the appointment by going to the **Appointments page** or click on **My Appointments** in your user menu.



Excellent! We now have a fully capable booking assistant that can fulfill all our customer’s needs and wishes, fully automatically.

3.4 Adding extra utility skills.

For this exercise we will add some convenience skills, like being able to inquire about order status and tracking information. Optionally you can also add the prompt + skill to register accounts via PLuG.

3.4.1 Order Status workflow.

This exercise is partially unguided on purpose. By now you should have a firm understanding of how the Agent Builder and Workflow engine works.

- Add/create the remaining prompt (`CS Agent Prompt part 4`) and import workflow **`GetOrderStatus-template.json`** test it.
- You do need to place an order first in the shop (fictional credit cards provided in the gui).
- This workflow needs the **API key** applied in the HTTP request.

When everything is in place, ask the Agent about an order.

3.4.2 Tracking Status workflow

This exercise is again partially unguided.

Your goal is to add a new workflow that allows a user to check a Tracking Code and show the tracking status.

For this you need your Agent to ask or act on an Order ID, in combination with a Tracking Code. This means you need to add these fields to the schema of the AI Skill Trigger node and use them as variables in the query parameters.

The API Docs section in the PetStore will help you get the correct endpoint URL and parameters for the API calls you need to do for the workflow. Make sure to use the user's API key to authorize any calls.

Finally, make sure you also extend the Agent prompt to instruct it correctly when people ask to check the Tracking of an order.

3.4.3 Password reset function.

For this final exercise, add a Password reset function to the Agent.

You've come a long way, but now you're on your own!