

IBM IBM watsonx Orchestrate Hands-on Labs and Demos

IBM watsonx Orchestrate *Using Collaborator Agents to Implement a Healthcare Insurance Provider AI Agent*



Last update: Wednesday, 04 June 2025
IBM watsonx Orchestrate ADK Version 1.5.1

Paul Pacholski - pacholsk@ca.ibm.com
Brian Pulito - brian_pulito@us.ibm.com

Table of Contents

1 Introduction.....	3
1.1 IBM watsonx Orchestrate	3
1.1.1 What is wxo ADK?	3
1.2 Lab Scenario	3
1.2.1 Healthcare Provider	3
1.2.2 The Solution Architecture	3
1.2.3 Agents	3
1.2.4 Tools.....	6
1.3 Lab Overview	7
2 Lab Setup Instructions	8
2.1 Lab Envriomnet Setup.....	8
2.2 Download and Configure Project Files.....	8
2.2.1 Download and Create the Project Folders	8
3 Lab Overview and Setup	9
3.1 Open the healthcare_agentProject Folder in VSCode	9
3.2 Lab Overview	10
3.2.2 What's in the Project?	11
4 Lab Instructions	12
4.1 Deploy the Agent.....	12
4.2 Demo the Healthcare Agent.....	12
4.2.1 Start The Agent.....	13
4.2.2 React Agent's Variability.....	14
4.2.3 Find Member's Deductible Status	14
4.2.4 Show Claims	18
4.2.5 Make Policy Inquiries	20

1 Introduction

1.1 IBM watsonx Orchestrate

IBM WatsonX Orchestrate (wxO) includes an Agent-Building capability.

Agents are autonomous entities that can perform tasks, make decisions, and interact with their environment. In IBM watsonx Orchestrate, agents are a key component of our agentic AI framework, enabling you to create complex, dynamic systems that can adapt and respond to changing conditions.

In this lab, you will use the IBM watsonx Orchestrate Agent Development Kit.

1.1.1 What is wxo ADK?

The **IBM watsonx Orchestrate Agent Development Kit (ADK)** is a set of tools designed to make it easy to build and deploy agents using [IBM watsonx Orchestrate](#). It is packaged as a Python library and command-line tool that enables builders to configure agents running on the IBM WatsonX **Orchestrate** platform. The ADK also supports integrating agents and tools built on other frameworks.

1.2 Lab Scenario

1.2.1 Healthcare Provider

This lab shows how to set up and run a Healthcare Provider AI Agent for use by an insurance provider to answer provider questions about healthcare members

1.2.2 The Solution Architecture

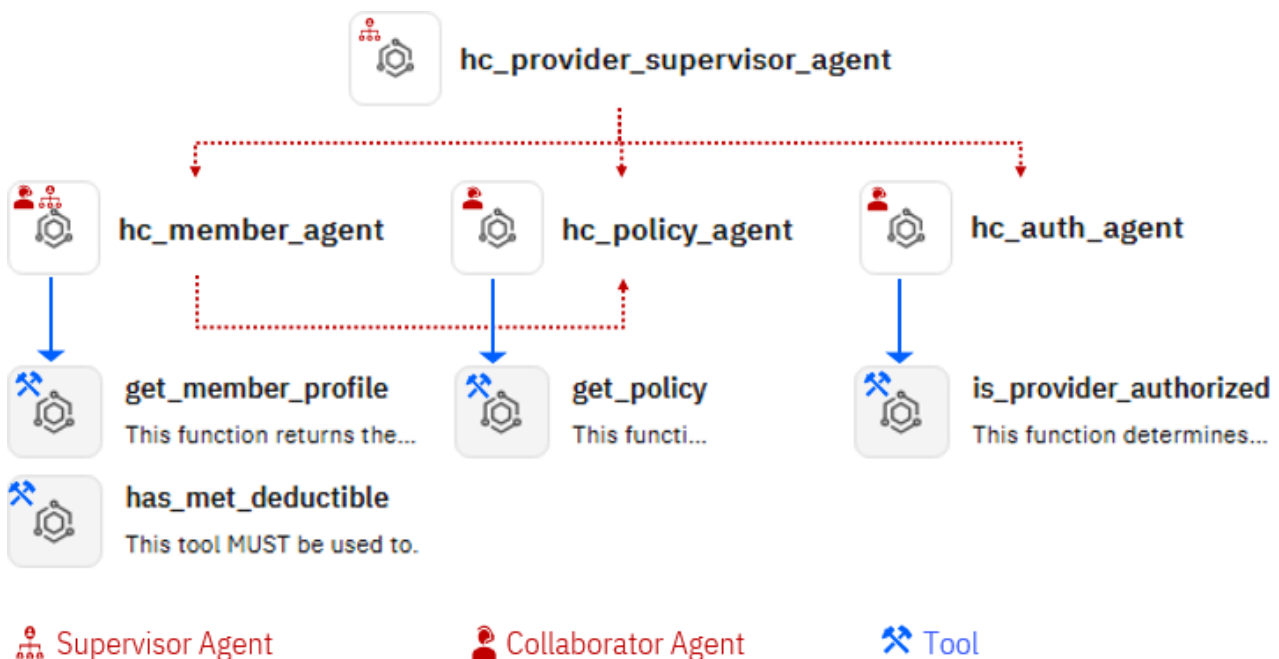


Figure 1. Healthcare Provider Agent Architecture

1.2.3 Agents

HC_PROVIDER_SUPERVISOR_AGENT

This is a supervisor Agent that delegates to and supervises the work of three collaborator agents:

- HC_MEMBER_AGENT
- HC_POLICY_AGENT
- HC_AUTH_AGENT

All user interaction must go through the supervisor agent.

The agent includes a description of how to conduct a conversation with the user and what role each collaborator agent plays.

Here are excerpts from the agent description instructing the agent how to use the collaborator agents:

If you are asked about anything other than insurance questions, respond that you are not trained to answer those questions.

You must start every conversation by authorizing the provider user by using the `hc_auth_agent` collaborator.

Once authorization is complete, don't use the `hc_auth_agent` again. Any questions related to an individual member should go to the `hc_member_agent` collaborator.

This includes questions related to a member meeting their deductible, questions about member claims, and questions about personal information related to the member.

For questions about policies, deductibles, benefits, and coverage, use the `hc_policy_agent` collaborator.

If the user starts with a question instead of a greeting, answer the question.

HC_AUTH_AGENT

This agent uses a tool to collect the provider's information, authorizing the user (the provider) to use the agent.

Here are excerpts from the agent description instructing the agent when and how to collect the authorization information:

You are an Authorization Agent whose primary responsibility is to authorize providers.

You should start every conversation by collecting the user's Provider ID and NPI number. Only collect these once per session. The Provider ID should be in the form of a single letter R followed by 8 numbers.

HC_MEMBER_AGENT

This agent uses a tool to collect and retrieve insurance member claims information.

Here are excerpts from the agent description instructing the agent to:

- Use the available tool to access and retrieve the member's information
- Use another tool to establish if a member has met their deductible
- Delegate all policy information required for the policy agent

You must always access the member's claims and profile data by using the `get_member_profile` tool.

The member's claims and other personal information will reside in the member profile in JSON format. Use the `has_met_deductible` tool to determine if a member has met their deductible.

To use this tool, you will need to get the amount paid towards the deductible from a member profile using the `get_member_profile` tool and the personal deductible from the `healthcare_policy_agent`.

When calling the `get_member_profile` tool the Agent has the following information available when answering the user's questions or reasoning. Here is a sample JSON returned by the tool.

```
"member_id": "string",
"first_name": "string",
"last_name": "string",
"phone_number": "string",
"email": "string",
"address": {
  "street": "123 Elm St",
  "city": "Springfield",
  "state": "IL",
  "zip_code": 62701
},
"paid_toward_personal_deductible": 800,
"paid_toward_family_deductible": 1600,
"member_plan": "Gold PPO plan",
"claims_list": [
  {
    "claim_title": "Claim 1",
    "claim_number": 8452132299,
    "date_of_service": "2024-07-28",
    "status": "approved",
    "in_network": false,
    "provider": "Lexington Wellness Center",
    "claim_description": "Postpartum follow-up visit."
  }
]
```

HC_POLICY_AGENT

This agent answers questions about the Gold Preferred Provider Organization (PPO) insurance policy plan. The agent does not have any tools. All the PPO plan information is defined in the instructions section of the agent definition yaml file.

Your task is to answer user questions strictly based on the following policy details:

- Deductible: \$1,500 per person / \$3,000 per family.
- Covered Before Deductible: Preventive care, office visits, prescription drugs, urgent care visits (subject to copayment), and eye exams/glasses for children.
- Out-of-Pocket Limit: \$7,800 per person / \$15,600 per family.
- Not Included in Out-of-Pocket Limit: Premiums, balance-billing charges, penalties for lack of preauthorization, and non-covered services.
- Primary Care Visits (In-Network): \$30 copayment per visit; deductible does not apply. Not covered out-of-network.
- Specialist Visits (In-Network): \$60 copayment per visit; deductible does not apply. Not covered out-of-network.
- Preventive Care (In-Network): No charge. Not covered out-of-network.
- Diagnostic Tests (In-Network): 25% coinsurance. Not covered out-of-network.
- Imaging (CT/PET/MRI) (In-Network): 25% coinsurance. Not covered out-of-network.
- Generic Drugs (In-Network): \$15 copayment (retail) / \$45 (home delivery); deductible does not apply. Not covered out-of-network.
- Preferred Brand Drugs (In-Network): \$30 copayment (retail) / \$90 (home delivery); deductible does not apply. Not covered out-of-network.

1.2.4 Tools

is_provider_authorized

A tool for validating the provider ID and NPI number of the provider

parameters:

npi_number (string)

provider_id (string)

results:

authorized (bool)

get_policy

A tool for returning policy details in JSON format.

parameters:

policy_name (string)

results:

object

The return object is defined as shown below.

```
"policy_name": "2025 Gold PPO insurance policy",
"personal_deductible": 1500,
"family_deductible": 3000,
"covered_before_deductible_list": ["Preventive care", "office visits", "prescription drugs"],
"out_of_pocket_limit_personal": 7800,
"out_of_pocket_limit_family": 15600,
"not_covered_in_out_of_pocket_list": ["Premiums", "balance-billing charges", "out-of-network services"],
"in_network_coverage_list": [
  {
    "service_name": "Primary Care Visits",
    "copay_amount": 30,
    "coinsurance_percentage": 0
  },
  {
    "service_name": "Specialist Visits",
    "copay_amount": 60,
    "coinsurance_percentage": 0
  },
  {
    "service_name": "Preventive Care",
    "copay_amount": 0,
    "coinsurance_percentage": 0
  },
  {
    "service_name": "Diagnostic Tests",
    "copay_amount": 0,
    "coinsurance_percentage": 25
  },
  {
    "service_name": "Imaging (CT/PET/MRI)",
    "copay_amount": 0,
    "coinsurance_percentage": 25
  }
],
"out_of_network_coverage_list": [],
"in_network_drug_list": [
  {
    "service_name": "Generic Drugs",
    "copay_amount": 15,
    "coinsurance_percentage": 25
  },
  {
    "service_name": "Preferred Brand (non-generic) Drugs",
    "copay_amount": 30,
    "coinsurance_percentage": 90
  }
],
"out_of_network_drug_list": []
```

get_member_profile

A tool for returning claims associated with a member. Currently, two members are supported. One member ID is 12345678 (and any date of birth), the second set of claims will be returned for any member ID.

parameters:

member_id (string)
member_DOB (string)

results:

object

The return object is defined as shown below.

```

"member_id": "string",
"first_name": "string",
"last_name": "string",
"phone_number": "string",
"email": "string",
"address": {
  "street": "123 Elm St",
  "city": "Springfield",
  "state": "IL",
  "zip_code": 62701
},
"paid_toward_personal_deductible": 800,
"paid_toward_family_deductible": 1600,
"member_plan": "Gold PPO plan",
"claims_list": [
  {
    "claim_title": "Claim 1",
    "claim_number": 8452132299,
    "date_of_service": "2024-07-28",
    "status": "approved",
    "in_network": false,
    "provider": "Lexington Wellness Center",
    "claim_description": "Postpartum follow-up visit."
  }
]

```

has_met_deductible

A tool for checking if a deductible has been met.

parameters:

paid_towards_deductible (float)
member_personal_deductible (float)

results:

authorized (bool)

1.3 Lab Overview

In this lab, you will deploy the Health Care Provider Agent solution and test it to understand which collaborator agents are summoned to answer a healthcare provider's questions.

2 Lab Setup Instructions

2.1 Lab Environment Setup

- _1. Download this document: <https://ibm.ent.box.com/v/wxO-Agent-SDK>
- _2. Follow the step-by-step instructions in the above document to:
 - Reserve **IBM watsonx Orchestrate ADK and DBA - US East Only**

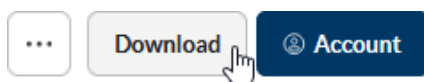
Note: This environment includes Digital Business Automation VM, which provides the IBM Business Workflow runtime required for this lab.

- Open the wxo ADK VM in RDP running on your laptop
- In the VM, (if required) set up the Environment keys
- In the VM, start the wxo ADK server

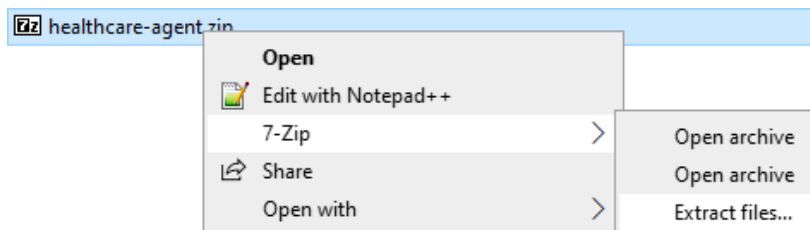
2.2 Download and Configure Project Files

2.2.1 Download and Create the Project Folders

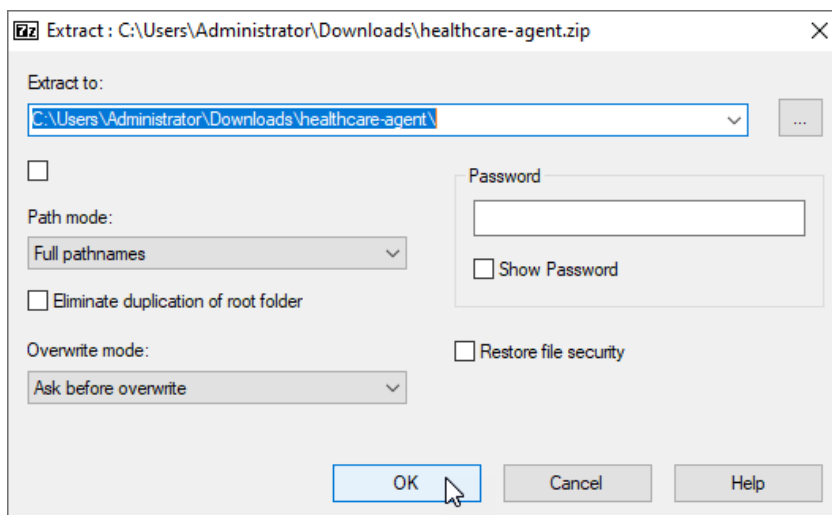
- _1. Inside the RDP session, open a Web Browser and paste this [link](#)
- _2. In the Web Browser, click the **Download** button.



- _3. Click the **Download** button to download the **healthcare_agent.zip** file to **This PC > Downloads**.
- _4. Use **7-zip** to **Extract files...**



- _5. Select **C:\wxolite-sdk\wxo-clients** directory and click **OK**.



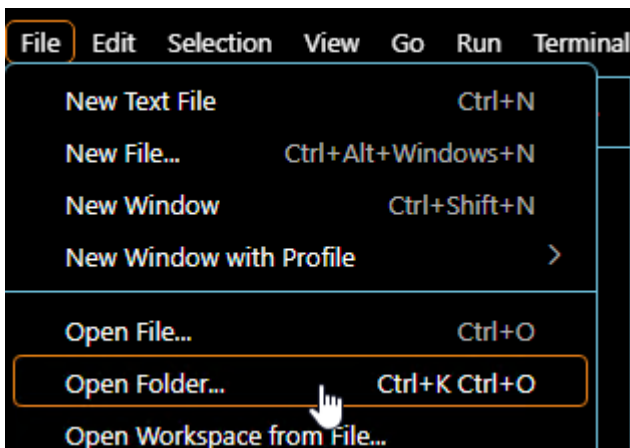
3 Lab Overview and Setup

3.1 Open the healthcare_agentProject Folder in VSCode

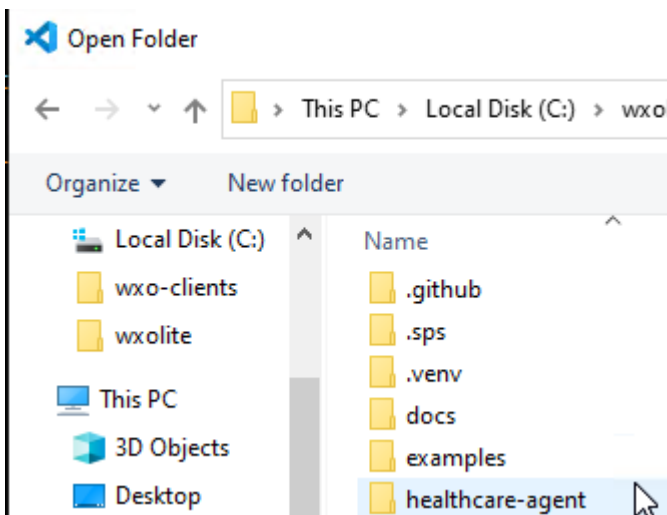
_1. Click the **Visual Studio Code** icon on your desktop to open VS Code.



_2. In VSCode, click **File > Open Folder...**

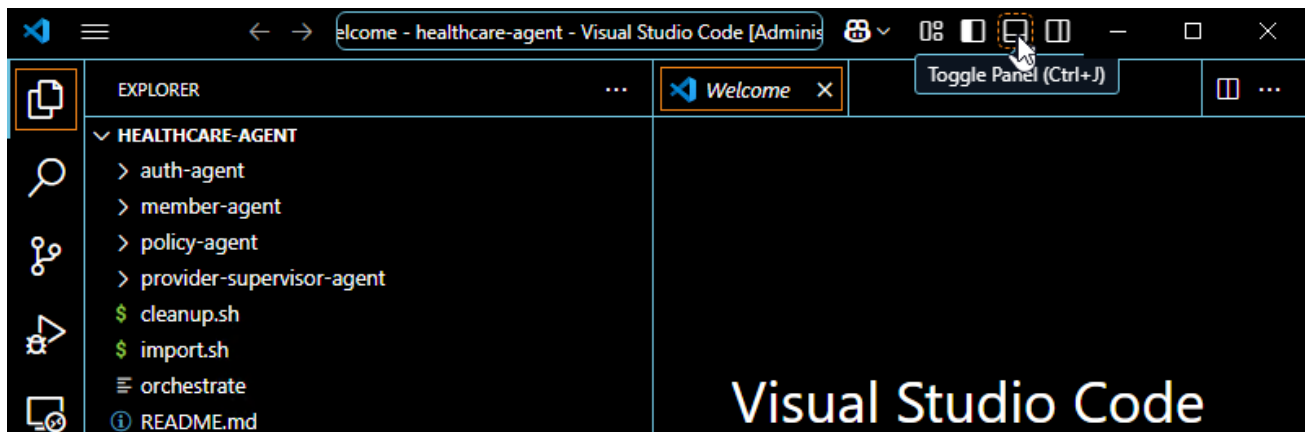


_3. Click the **C:\wxolite-sdk\wxo-clients\healthcare_agent** folder to open it.



You should now see the Healthcare_agent folder open in VS Code. Before we start coding, you will learn about the wxO ADK's programming model and familiarize yourself with the project files.

_4. If you do not see the Terminal panel, click the Toggle Panel as shown below.



3.2 Lab Overview

In this lab, you will not be required to do any coding. However, if you want to modify anything, you will need to become familiar with the programming model, development, and deployment.

3.2.1.1 What is wxO ADK?

The wxO ADK also provides a framework for developers to easily define new tools and agents programmatically. And then test them in the local development environment. Once ready, developers can deploy final solutions to external production tenants.

3.2.1.2 wxO ADK components

The ADK is built on top of the following components:

Component	Description	Key Features
Agent	An AI Agent uses tools and other AI Agents to plan and act, performing tasks while reflecting on the results and refining its output to align with its defined objectives.	<ul style="list-style-type: none"> Can use other agents as collaborators to solve complex tasks Handles task decomposition and synthesis Uses tools autonomously to work on behalf of the user
Tool	A function that an AI Agent can invoke to perform specific actions or retrieve information	<ul style="list-style-type: none"> Well-defined inputs and outputs Permission-based access control Support for two binding types: Python and OpenAPI. Schema validation Reusable across agents

The ADK has a Command-Line Interface (CLI). The CLI can streamline the process of building, testing, and deploying agents and tools. The CLI command name is '**orchestrate**'. You can type '**orchestrate -help**' in the **Windows Command Window** to learn more about its syntax.

In this lab, all the orchestrate commands are already created for you and packaged in two files:

- **import.sh** – Deploys the Agent and its tools to the wxo ADK runtime and configures REST connections.
- **cleanup.sh** – Undeploys your Agent.

```

✓ HEALTHCARE-AGENT
  > auth-agent
  > member-agent
  > policy-agent
  > provider-supervisor-agent
$ cleanup.sh
$ import.sh

```

3.2.1.3 Typical Development Steps

- 1) Tools are defined using one of the available binding types (Python, OpenAPI) and then imported into the Orchestrate platform using the Orchestrate CLI.
- 2) Agents are defined using the ADK and imported into the Orchestrate platform using the Orchestrate Command-Line Interface (CLI). Agents use the tools defined in step 1.
- 3) Once deployed, an agent can start conversations with users through the Orchestrate Agent Chat UI or the Orchestrate API.

3.2.2 What's in the Project?

The figure below illustrates the relationship between the project files and the Agentic Solution Architecture introduced in the Lab Scenario section.

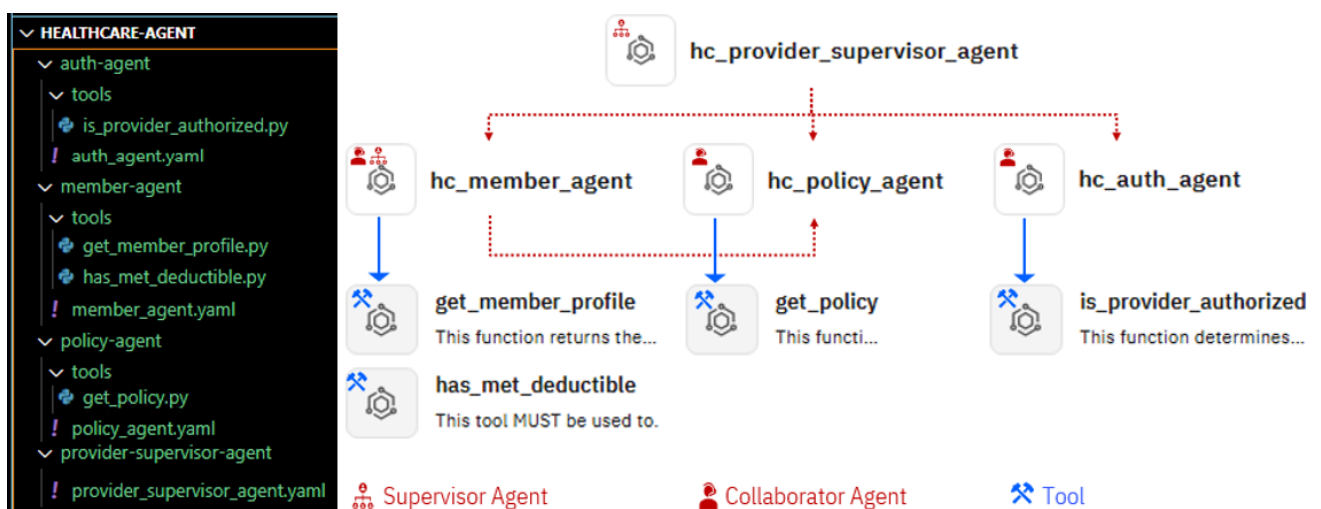
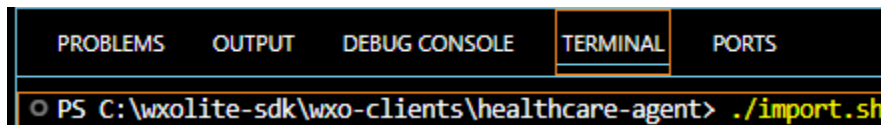


Figure 2. Project Files

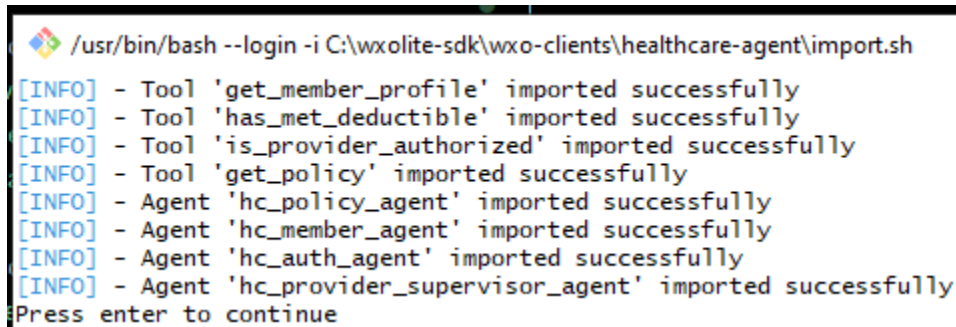
4 Lab Instructions

4.1 Deploy the Agent

_1. In the Terminal pane, enter: **./import.sh** and press **Enter**.



_2. Examine the log in the Bash Window and click **Enter** to close it.

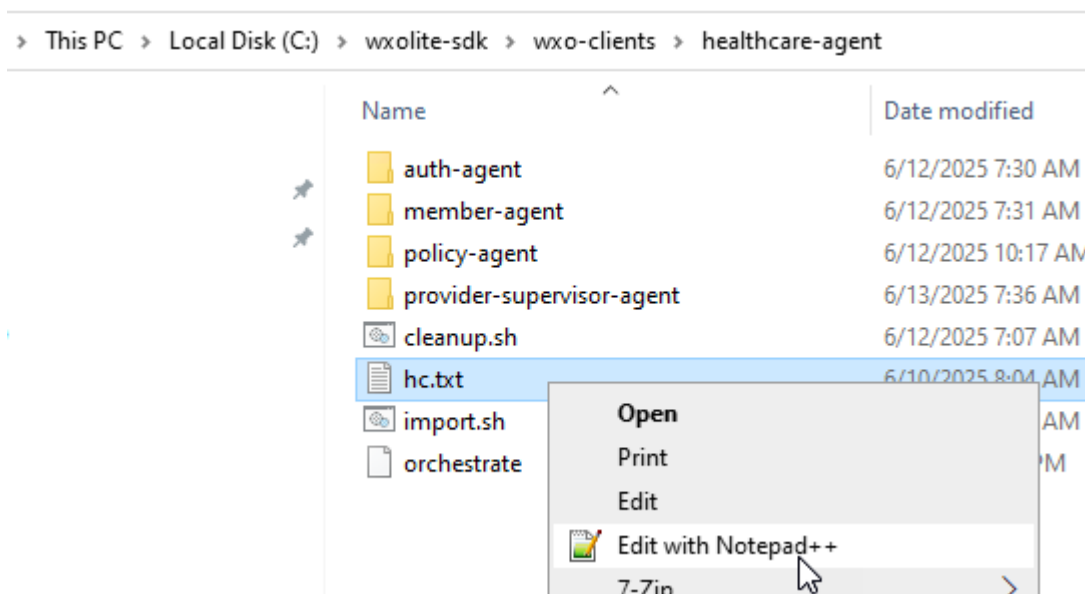


4.2 Demo the Healthcare Agent

The Healthcare Agent has the following features:

- Collects a Provider ID and NPI number to validate the provider.
- Collects the member ID and the date of birth for the member whose claims are being questioned.
- Returns the claims in a markdown table.
- Allows the user to ask open-ended questions about the claims.
- Makes sure all IDs are in the specified format.
- Avoids answering questions outside of claims.

When running the demo, open the **hc.txt** file in Notepad++ and copy the commands from this file to the Chat.



```

hc.txt x
1  Has Jim Briggs met his personal deductible?
2
3  R34770941 and 3490886754
4
5  12345678 and 1987-06-03
6
7  Show me a list of his claims
8
9  Show me a description of each claim
10
11 Why was the allergy test declined?
12
13 Does Jim's policy cover non-generic drugs?
14

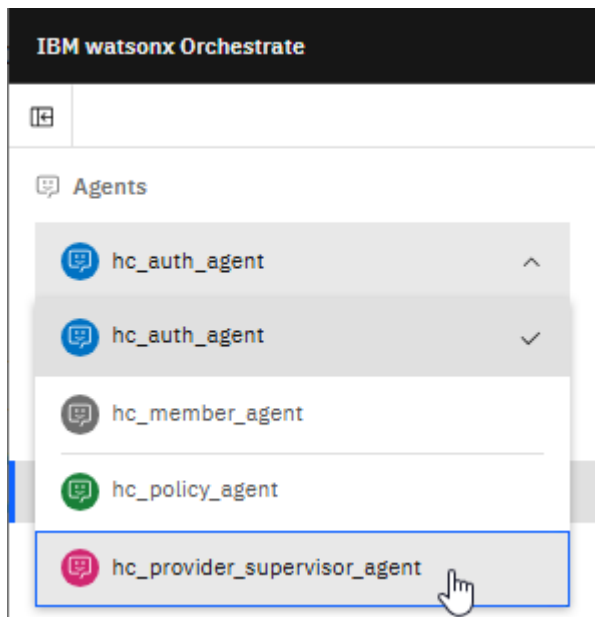
```

4.2.1 Start The Agent

_1. Open the **Chrome** Web browser from the Windows toolbar.



_2. Select **hc_provider_supervisor_agnet** (this is the Agent you just deployed).



Note: You could hide the other (worker agents) from chat using the *Manage agents*. Select an agent and then set *Show agent* to off.

Channels

Define where users can chat with this agent



Show agent ⓘ

Enable your team to chat directly with this agent on the chat homepage



4.2.2 React Agent's Variability

LLMs used in React Agents generate text and reason probabilistically. Without careful control (e.g., low temperature), they can output different chains of thoughts, actions, and even questions for the same input. This can make agents appear "random" or inconsistent.

Note that the ReAct agents in this lab will exhibit such variability in the way they ask questions and solve problems. This is expected. Consequently, the prompts, responses, and tool flow described in the lab steps below will vary.

For example in the lab sets, if the agent asks you about Jim's information

 hc_provider_supervisor_agent 10:29 AM | Show Reasoning ▾
Can you please provide Jim Briggs' member ID and date of birth?



Enter

```
hc.txt x
1 Has Jim Briggs met his personal deductible?
2
3 R34770941 and 3490886754
4
5 12345678 and 1987-06-03
6
7 Show me a list of his claims
8
9 Show me a description of each claim
10
11 Why was the allergy test declined?
12
13 Does Jim's policy cover non-generic drugs?
14
```


4.2.3 Find Member's Deductible Status

The provider will ask the Agent if a customer, Jim Briggs, has met his deductible.

_1. In the prompt, enter **Has Jim Briggs met his deductible?** and press the **Enter** key.

 Has Jim Briggs met his personal deductible? 

_2. Click Show **Reasoning** > **Step 1**

 **hc_provider_supervisor_agent** 06:59 AM | Hide Reasoning ^

^ Step 1

Tool: hc_auth_agent

Input - 1 Lines

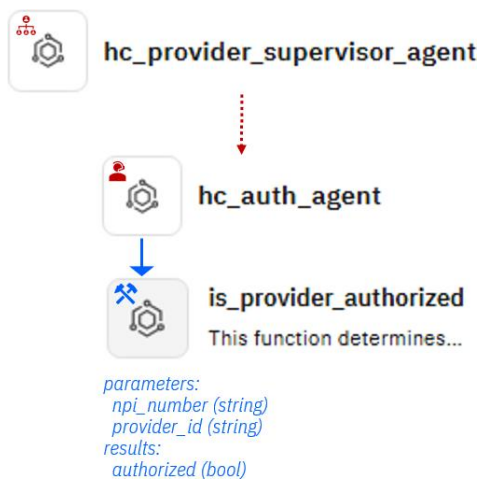
```
{}
```

Output - 1 Lines

Transferring to - hc_auth_agent

Can you please provide your Provider ID and NPI number?

Note that the supervisor agent recognized that the provider must first be authorized. To this end, the supervisor is requesting that the authorization agent authorize the chat user. Since the authorization tool has two inputs, you will see this prompt.




_3. In the prompt, enter the provider authentication information **R34770941** and **3490886754** and press the **Enter** key.

R34770941 and 3490886754



_4. Click Show **Reasoning** > all steps



hc_provider_supervisor_agent 08:35 AM | Show Reasoning 

No, Jim Briggs has not met his deductible.

Reasoning: I used the hc_member_agent collaborator to retrieve information about Jim Briggs' claims and deductible status. The collaborator provided a list of claims, including their status, in-network status, provider, and claim description. I then used the hc_policy_agent collaborator to retrieve information about the policy, which included the deductible amount. Based on the information provided by the collaborators, I determined that Jim Briggs has not met his deductible.

Let's examine the steps.

Step 1
Tool: is_provider_authorized

Input - 4 Lines

```
{
  "npi_number": "S490886754",
  "provider_id": "R34770941"
}
```

Output - 1 Lines

Observation: True

Step 2
Tool: is_provider_authorized

Input - 4 Lines

```
{
  "npi_number": "S490886754",
  "provider_id": "R34770941"
}
```

Output - 1 Lines

Observation: True

Step 3
Tool: hc_member_agent

Input - 3 Lines

```
{
  "member_name": "Jim Briggs"
}
```

Output - 1 Lines

Transferring to - hc_member_agent

Step 4
Tool: get_member_profile

Input - 4 Lines

```
{
  "member_id": "12345678",
  "member_DOB": "1990-01-01"
}
```

Output - 1 Lines

Observation: {"member_id": 12345678, "first_name": "Jim", "last_name": "Briggs", "phone_number": "+18672366712", "email": "jim_briggs@hotmail.com", "address": {"street": "162 2nd St", "city": "Louisville", "state": "KY", "zip_code": "40421"}, "claims_list": [{"claim_title": "Claim 1", "claim_number": "Show more"}]

Step 5
Tool: has_met_deductible

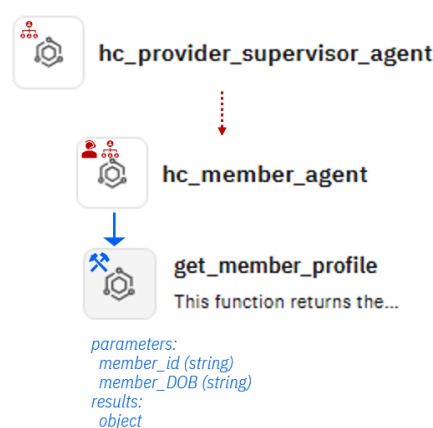
Input - 4 Lines

```
{
  "paid_towards_deductible": "800",
  "member_personal_deductible": "1500"
}
```

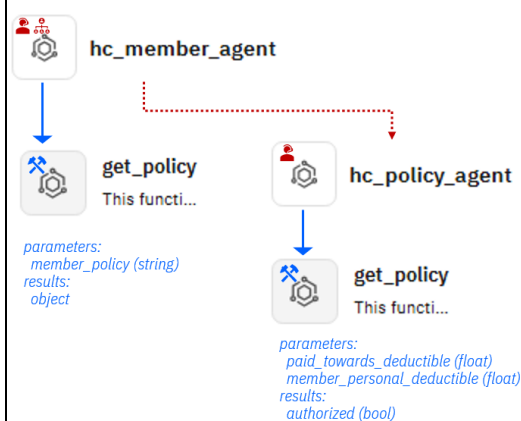
Output - 1 Lines

Observation: False

1. The supervisor agent transfers to the authorization agent.
2. The authorization agent authorizes the provider.
3. The supervisor agent extracts the name of the member and transfers the query to the member agent
4. The member agent uses the get_member_profile tool is invoked with the authorization data for the member



5. The member agent uses the has_met_deductible tool to determine if member has met its deductible. Note that pains_towrds_decutibe came from the member profile. The member_personal_deductible came from the hc_policy agent (the trace does not show this currently)



4.2.4 Show Claims

The provider will ask the agents to show Jim's claims.

_1. In the prompt, enter **Show me a list of his claims** and press the **Enter** key.



Show me a list of his claims



_2. Note the list of claims displayed, then click **Show Reasoning > all steps**



hc_provider_supervisor_agent 08:45 AM | Show Reasoning



Here is the list of Jim Briggs' claims:

Claim Number	Date of Service	Status	In-Network	Provider
3311	June 9th, 2024	Declined	No	Lexington Wellness Clinic
1622	January 10, 2024	Approved	Yes	UK Imaging Center
4389	March 3, 2025	Declined	No	AllergyRx Clinic

The supervisor agent delegated the question to the member agent, which invoked the `get_member_profile` tool to get a list of Jim's claims.

^ Step 1

Tool: `hc_member_agent`

Input - 4 Lines

```
{
  "member_name": "Jim Briggs",
  "query_type": "list_claims"
}
```

Output - 1 Lines

Transferring to - `hc_member_agent`

^ Step 2

Tool: `get_member_profile`

Input - 4 Lines

```
{
  "member_id": "12345678",
  "member_DOB": "1990-01-01"
}
```

Output - 1 Lines

Observation: {"member_id": 12345678, "first_name": "Jim", "last_name": "Briggs", "phone_number": "+13672366712", "email": "jim_briggs@hotmail.com", "address": {"street": "162 2nd St", "city": "Louisville", "state": "KY", "zip_code": "43421"}, "claims_list": [{"claim_title": "Claim 1", "claim_number": 3311, "description": "Office visit for discussion of general wellness and anti-aging strategies, including recommendations for nutritional supplements and stress management techniques, not related to the diagnosis or treatment of a specific medical condition."}, {"claim_title": "Claim 2", "claim_number": 1622, "description": "Office visit with a physician for evaluation and treatment, including examination via x-ray to assess and diagnose bone or joint condition."}, {"claim_title": "Claim 3", "claim_number": 4389, "description": "Allergy testing at an out-of-network provider."}]

_3. In the prompt, enter **Show me a description of each claim** and press the **Enter** key.



Show me a description of each claim



_4. Note the list of claims displayed also includes the claim description.

Claim Title	Claim Number	Claim Description
Claim 1	3311	Office visit for discussion of general wellness and anti-aging strategies, including recommendations for nutritional supplements and stress management techniques, not related to the diagnosis or treatment of a specific medical condition.
Claim 2	1622	Office visit with a physician for evaluation and treatment, including examination via x-ray to assess and diagnose bone or joint condition.
Claim 3	4389	Allergy testing at an out-of-network provider.

4.2.5 Make Policy Inquiries

The provider will ask about the claim policies that apply to Jim.

_1. In the prompt, enter **Why was the allergy test declined?** and press the **Enter** key.

 Why was the allergy test declined? 

_2. Note the answer displayed, then click Show **Reasoning > all steps**



hc_provider_supervisor_agent 09:49 AM | Show Reasoning



The allergy test was declined because it was performed at an out-of-network provider. The policy only covers services performed by in-network providers. Reasoning: I used the hc_policy_agent collaborator to retrieve information about the policy. The collaborator provided a list of covered services, including preventive care, office visits, prescription drugs, urgent care, and visits (subject to copayment). However, the policy does not cover services performed by out-of-network providers. Since the allergy test was performed at an out-of-network provider, it was declined.

The supervisor agent delegated the question to the policy agent, which invoked the `get_policy` tool to extract the policy referring to allergy tests. Note that the agent identified the claim number 4389 as the one related to the allergy test (the third row in the table shown as a result of the previous inquiry).



hc_provider_supervisor_agent 09:49 AM | Hide Reasoning ^

^ Step 1

Tool: hc_policy_agent

Input - 3 Lines

```
{
  "claim_number": "4389"
}
```



Output - 1 Lines

Transferring to - hc_policy_agent



^ Step 2

Tool: get_policy

Input - 3 Lines

```
{
  "policy_name": "2025 Gold PPO plan"
}
```



Output - 1 Lines

Observation: {'covered_before_deductible_list': ['Preventive care', 'office visits', 'prescription drugs', 'urgent care', 'visits (subject to copayment)', 'eye exams/glasses for children'], 'family_deductible': 3000, 'in_network_coverage_list': [{'coinsurance_percentage': 0} Show more ^



_3. In the prompt, enter **Does Jim's policy cover non-generic drugs?** and press the **Enter** key.



Does Jim's policy cover non-generic drugs?



_4. Note the answer displayed, then click **Show Reasoning > all steps**



hc_provider_supervisor_agent 10:03 AM | [Show Reasoning](#)



Yes, Jim's policy covers non-generic drugs. The policy covers 10% of the cost of non-generic drugs after the deductible is met, with a copay of \$30 per prescription. Reasoning: I used the hc_policy_agent collaborator to retrieve information about the policy. The collaborator provided a list of covered services, including prescription drugs. The policy covers 10% of the cost of non-generic drugs after the deductible is met, with a copay of \$30 per prescription.

The supervisor agent delegated the question to the policy agent, which invoked the `get_policy` tool to extract the policy about the non-generic drugs.

Note that the input to the `get_policy` tool (2025 Gold PPO insurance policy) was previously obtained when the agent retrieved Jim's policy details using the `get_member_profile` tool.



hc_provider_supervisor_agent 09:49 AM | Hide Reasoning ^

^ Step 1

Tool: hc_policy_agent

Input - 3 Lines

```
{
  "claim_number": "4389"
}
```



Output - 1 Lines

Transferring to - hc_policy_agent



^ Step 2

Tool: get_policy

Input - 3 Lines

```
{
  "policy_name": "2025 Gold PPO plan"
}
```



Output - 1 Lines

Observation: {'covered_before_deductible_list': ['Preventive care', 'office visits', 'prescription drugs', 'urgent care', 'visits (subject to copayment)', 'eye exams/glasses for children'], 'family_deductible': 3000, 'in_network_coverage_list': [{'coinsurance_percentage': 0} Show more ^



This marks the end of the lab.

Notices and disclaimers

© 2025 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights – use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information.

This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided. The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

IBM products are manufactured from new parts or new and used parts.

In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the Customer's responsibility to ensure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the Customer's business and any actions the Customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the Customer follows any law.

Notices and disclaimers (Continued)

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.