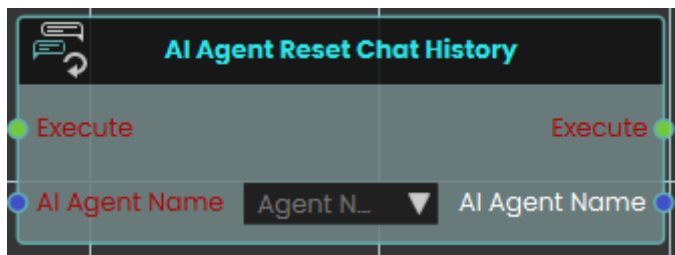


AI Agent

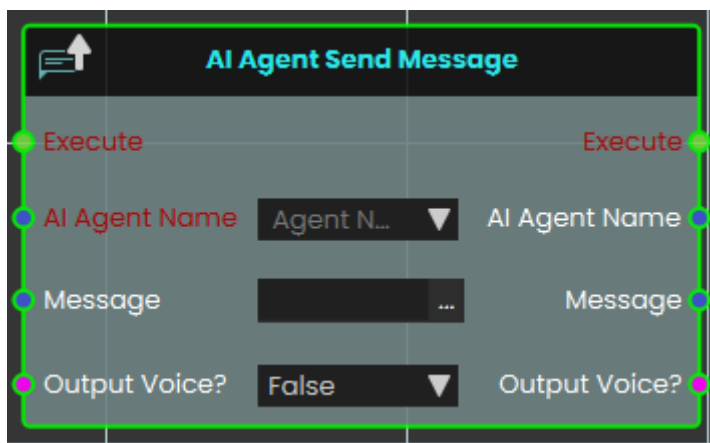
AI Agent Reset Chat History



The **AI Agent Reset Chat History** node

enables the system to clear and reset the ongoing conversation history for a specific AI Agent. It is typically used when you want the AI to start a completely fresh interaction or switch to a new scenario without being influenced by the context of any prior messages.

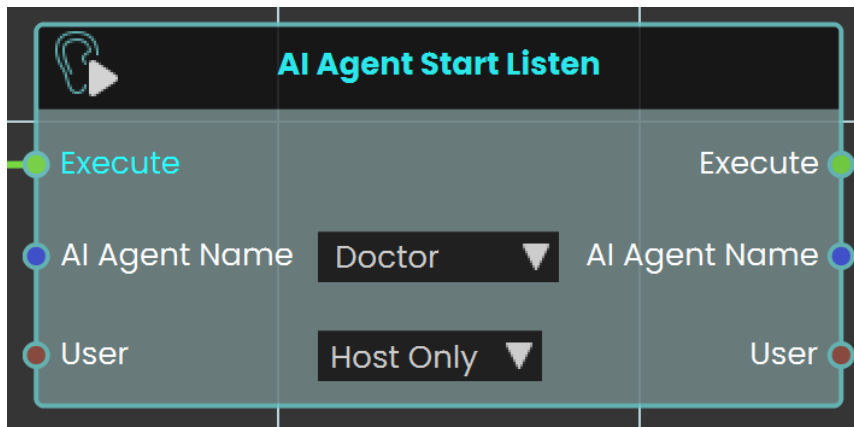
AI Agent Send Message



The **AI Agent Send Message** node enables the system to send a specific text message to an assigned AI Agent for processing. The **Output Voice?** boolean parameter that dictates how the AI agent will deliver its response:

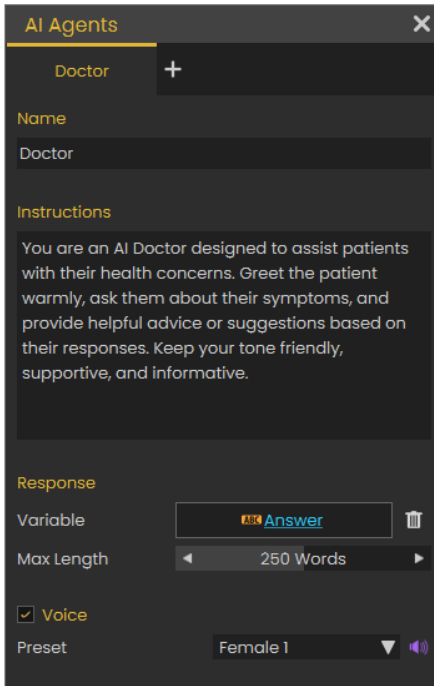
- **False:** The AI agent will respond with text.
- **True:** The AI agent will process the text message and respond with generated voice audio.

AI Agent Start Listen

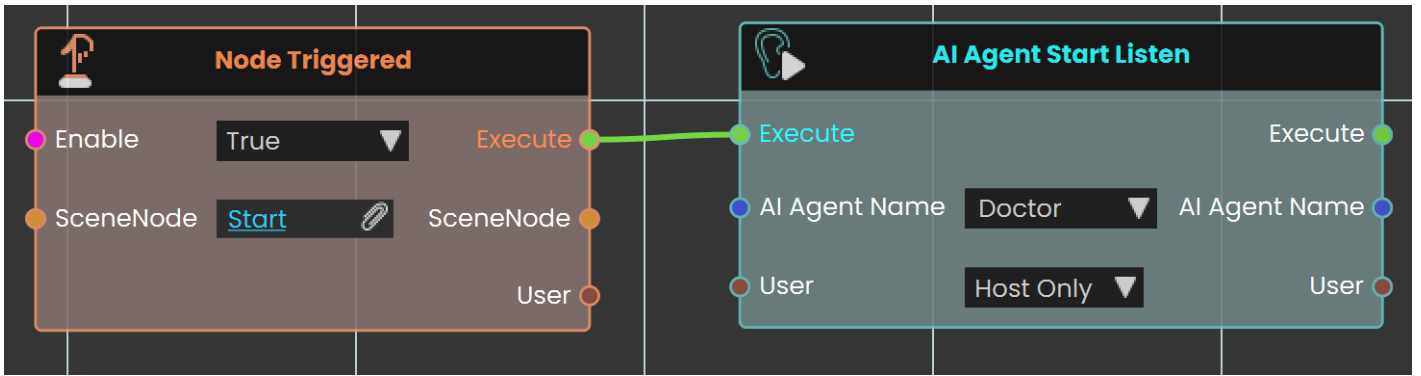


The **AI Agent Start Listen node** is used to make the AI start listening to the user. When activated, this node enables the AI to process and listen to spoken input from the user, allowing for interactive conversations and commands within the VR environment.

Example



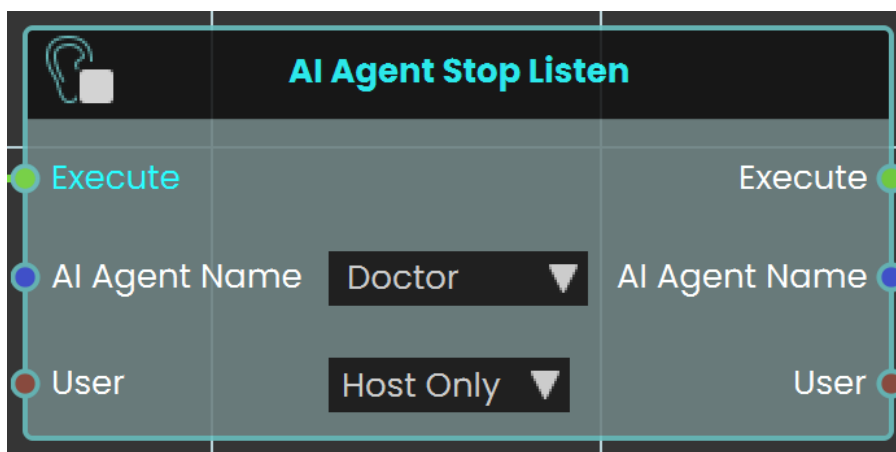
In this example, an AI Doctor is set up in the AI Agents window. This window can be accessed by clicking the Interaction icon in the viewport menu, then select the AI Agents, and then add the AI name with the desired instructions in the Instructions field. A variable is created to store the AI's responses.



The **AI Agent Start Listen node** is used to make the AI Agent named Doctor start listening to the user when the object named "Start" is triggered. This setup enables the AI to process and listen to user input as soon as the trigger event is activated.

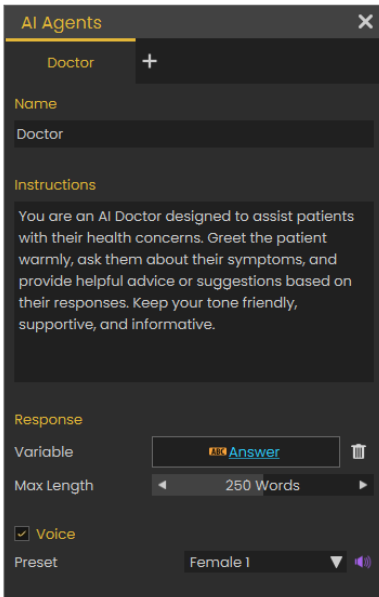
<https://www.youtube.com/embed/TdABZrHXY2g>

AI Agent Stop Listen

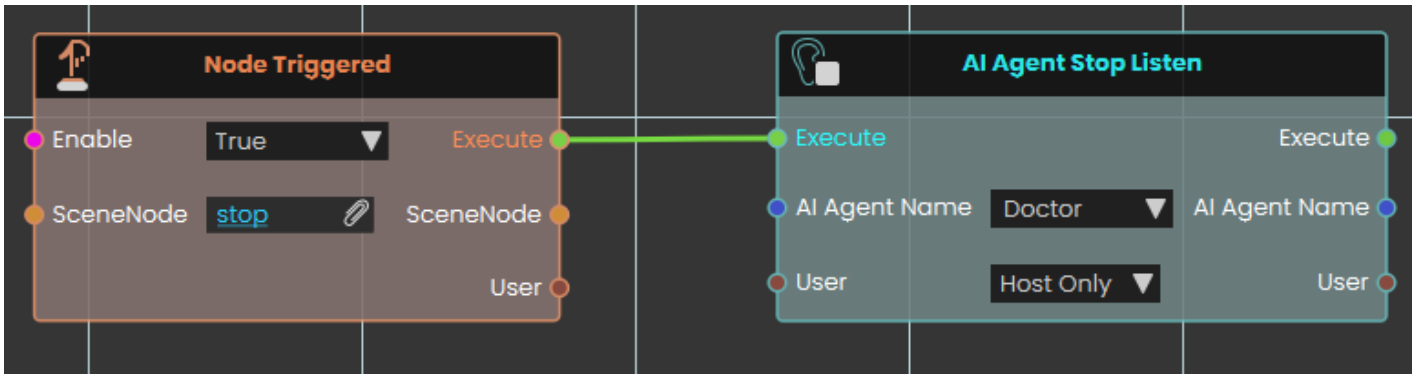


The **AI Agent Stop Listen node** is used to make the AI stop listening to the user. When activated, this node halts the AI's ability to process further user input, allowing the AI to respond based on the information gathered up to that point.

Example

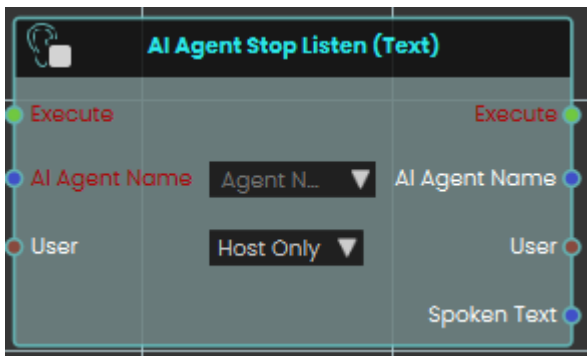


In this example, an AI Doctor is set up in AI Agents window. This window can be accessed by clicking the Interaction icon in the viewport menu, then select the AI Agents, and then add the AI name with the desired instructions in the Instructions field. A variable is created to store the AI's responses.



The **AI Agent Stop Listen** node is used to stop the AI Agent from listening to the user when the "Stop" trigger event occurs. This allows the AI agent named Doctor to respond based on the input received before listening was stopped.

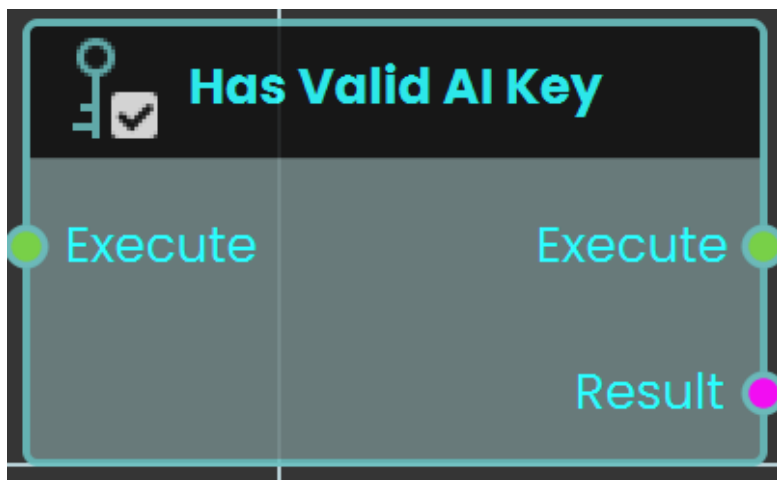
AI Agent Stop Listen (Text)



The **AI Agent Stop Listen (Text)** node is used to make the AI stop listening to the user. When activated, this node halts the AI's ability to process further user input, allowing the AI to respond based on the information gathered up to that point.

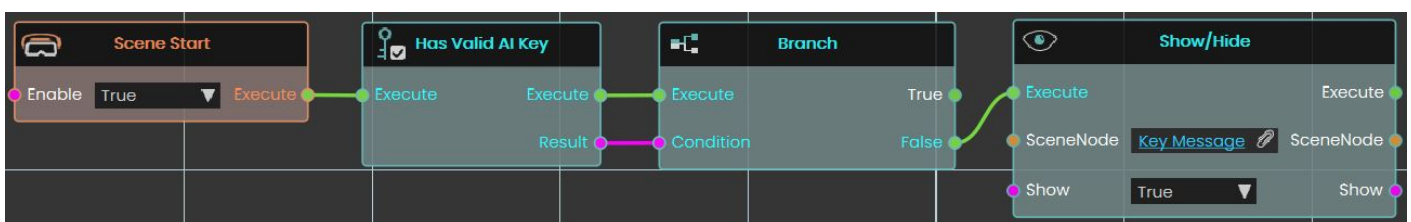
This node has an extra output "Spoken Text", when the node is activated, it outputs the input audio in text format.

🔑 Has Valid AI Key



The **Has Valid AI Key** node checks whether a valid AI key is available for AI-related features in Sim Lab Composer. This node ensures that AI functionalities can operate properly by verifying the presence of an active and valid AI key.

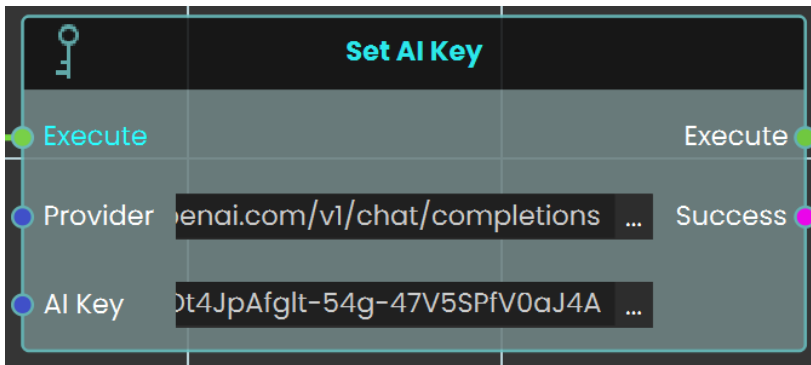
Example



In this example, the **Has Valid AI Key** node is used at the start of the scene to check if the user has an activated AI key. The node is connected to a **Branch** node, where the **False** output (indicating no valid AI key) triggers a message in front of the user, informing them that the AI key must be activated.

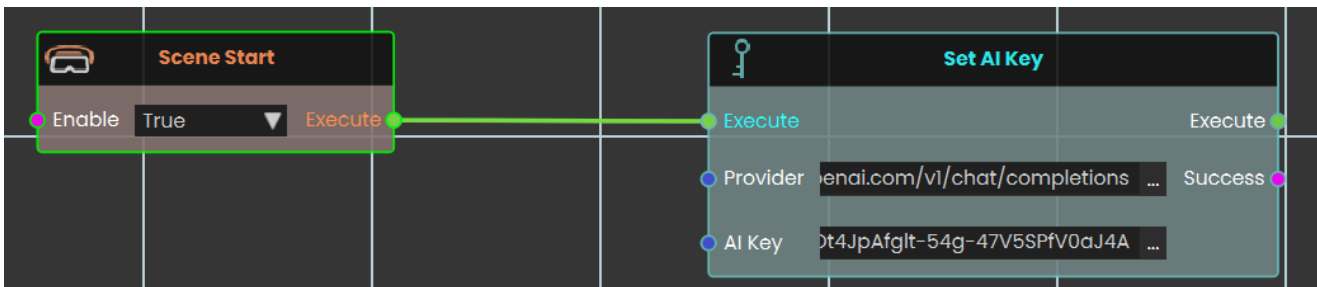
<https://www.youtube.com/embed/YEMeCS8QcFY>

Set AI Key



The **Set AI Key** node allows you to assign an AI API key to a VR experience directly from the Training Builder. This lets the experience use AI features without requiring the user to manually activate an API key in the viewer. You can select the provider (OpenAI, Gemini, or OpenRouter) and embed the corresponding API key into the experience.

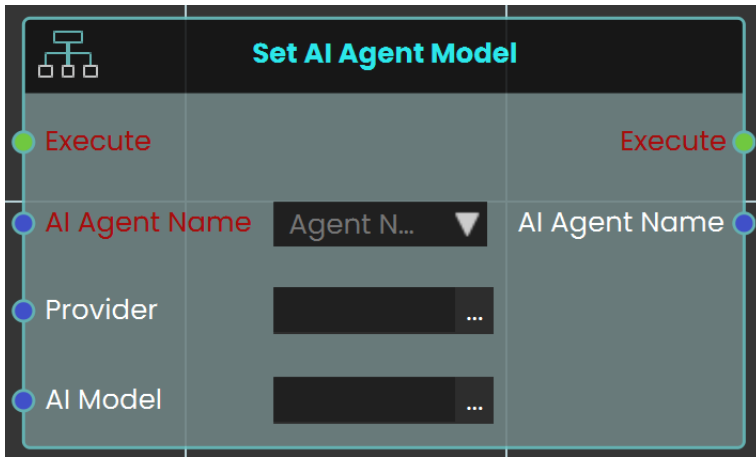
Example



In this example, the **Set AI Key** node is connected to a **Scene Start** event to initialize the AI provider as soon as the experience begins. After adding the node and selecting the provider, the API key is pasted into the node's field. The key is then activated when the scene starts, allowing the AI agent to respond without the user needing their own key. If the embedded key is not active or reaches its usage limit, the experience automatically switches to the user's API key in the viewer if one is available.

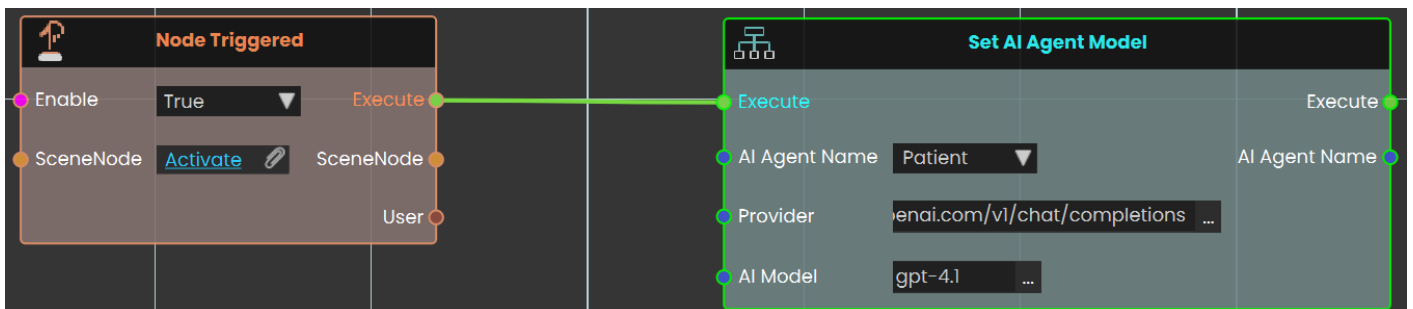
Check this [tutorial](#) for more about this node.

Set AI Agent Model



The **Set AI Agent Model** node is used to assign a specific AI provider and model to an AI agent in your VR experience. This ensures the AI agent will operate using the selected provider and model for any interactions that occur in the scene. You can choose the AI agent you have created, specify the provider (OpenAI, Gemini, or OpenRouter), and define the model name.

Example



In this example, the **Set AI Agent Model** node is triggered by pressing a button in the VR scene. When the button is clicked, the node assigns the “Patient” AI agent to the OpenAI provider and sets the model to GPT-4. This setup allows the AI agent to function with the defined provider and model as soon as the event is triggered.

Check this **tutorial** for more about this node.

<https://www.youtube.com/embed/qzqNtXBTGY4>

Using AI Providers and API Keys in SimLab Composer

SimLab Composer now supports **OpenAI, Google Gemini, and OpenRouter** for integrating AI into your VR experiences. With the new **Set AI Key** and **Set AI Agent Model** nodes, you can connect your experience to these providers .

Providers and Models:

1 OpenAI

- **Provider URL:** `https://api.openai.com/v1/chat/completions`
- **Models:**
 - gpt-4.1
 - gpt-4
 - gpt-3.5-turbo

2 Google Gemini (OpenAI-compatible endpoint)

- **Provider URL:**

`https://generativelanguage.googleapis.com/v1beta/openai/chat/completions`

- **Models:**

- gemini-2.0-flash

3 **OpenRouter**

- **Provider URL:** `https://openrouter.ai/api/v1/chat/completions`

- **Models:**

- openrouter/cypher-alpha:free
- nvidia/llama-3.3-nemotron-super-49b-v1:free

- **OpenAI keys** work with **Chat, Text-to-Speech, and Speech-to-Text**

- **Gemini and OpenRouter keys** work with **Text only.**

■ ■

■

■

Revision #52

Created 2 September 2024 13:22:09

Updated 9 March 2026 12:56:06 by Ahmad Qasim