

HttpGemini Plugin - Official Documentation

Author: R3D HILLS

Last Updated: October 1, 2025

1. Overview

HttpGemini is a lightweight and powerful Unreal Engine plugin designed to seamlessly integrate Google's Gemini API directly into your projects using Blueprints. It empowers developers to implement runtime conversational AI features in games and applications without writing a single line of C++ or relying on third-party libraries.

2. Key Features

- **100% Blueprint Ready:** Access all functionalities through a single, easy-to-use Blueprint node.
- **Runtime Communication:** Send messages and receive AI-generated responses from the Gemini API live.
- **Easy Configuration:** Set your API Key and choose the Gemini Model directly from the Project Settings.
- **Asynchronous by Design:** API requests are non-blocking, ensuring your game's performance remains smooth without any freezes or frame drops.
- **Robust Error Handling:** Get clear feedback on connection issues, invalid API keys, or other potential problems.
- **No External Dependencies:** The plugin is self-contained and uses Unreal Engine's built-in HTTP and JSON modules.

3. Technical Specifications

- **Supported Engine Versions:** 5.5, 5.6+
- **Supported Platforms:** Windows (64-bit), Android

4. Installation & Configuration

Follow these two simple steps to get the plugin up and running.

Step 1: Enable the Plugin

1. In the Unreal Editor, navigate to Edit > Plugins.
2. Search for HttpGemini and ensure the **Enabled** checkbox is ticked.
3. Restart the editor when prompted.

Step 2: Configure Settings

1. Navigate to Edit > Project Settings.
2. On the left panel, scroll down to the **Plugins** section and click on **HttpGemini**.
3. Fill in the following two fields:
 - **Gemini API Key:** Paste your unique API key obtained from the Google AI Studio.
 - **Gemini Model:** Select a model from the dropdown. Gemini 1.5 Flash is recommended for its speed and cost-effectiveness.

**

5. How to Use: The "Send Chat Message (Gemini)" Node

The core of the plugin is the Send Chat Message (Gemini) Blueprint node. It can be used in any Blueprint graph (e.g., an Actor or a Widget Blueprint).

Inputs:

- **Message (String):** The text message or prompt you want to send to the Gemini API.

Outputs:

- **OnResponseReceived (Event Dispatcher):** This is an asynchronous event that fires as soon as a response is received from the server. Your game logic will not wait for it.
- **ResponseText (String):** This event's output parameter contains the AI-generated text response.

6. Blueprint Example

1. In any Blueprint graph, right-click and add the **Send Chat Message (Gemini)** node.
2. Connect a string variable or text to the **Message** input pin.
3. Drag a wire from the **OnResponseReceived** output pin and create a **Custom Event**.
4. Name it something descriptive, like HandleGeminiResponse.
5. This new event will now execute whenever a response is available. The **ResponseText** pin from this event holds the reply from Gemini.
6. You can use a Print String node to display it or store it in a variable for your UI.

**

7. Important Requirements

- **Valid API Key:** You must have a valid API Key from the Google AI platform to use this plugin.
- **Internet Connection:** The device running your project must have an active internet

connection at runtime to communicate with the API.

8. Troubleshooting

Error: API Key not set

- **Cause:** The API Key field in the Project Settings is empty.
- **Solution:** Navigate to Project Settings > Plugins > HttpGemini and enter your valid API key.

Error: HTTP request failed (often with a response code like 400, 403, or 500)

- **Cause:** There was a problem connecting to Google's servers. This can be due to a lack of internet connectivity, a firewall blocking the request, or an invalid/disabled API key.
- **Solution:** Check your internet connection. Ensure your API key is correct and active. A 4xx error code usually indicates a problem with your request or key, while a 5xx error indicates a server-side issue with Google's API.

Error: Could not parse response

- **Cause:** The API returned a response, but its structure was not what the plugin expected. This can sometimes happen if the API is updated.
- **Solution:** Please check for plugin updates or contact support.

9. Support

For bug reports, feature requests, or technical support, please contact us:

- **Website:** <https://r3dhills.com>
- **Email:** info@r3dhills.com