Getting Started

You can find latest documentation online at https://docs.driftingmoose.com/.

If you have any questions - join our Discord to contact support.

Table of contents

Configure AI Providers

OpenAI

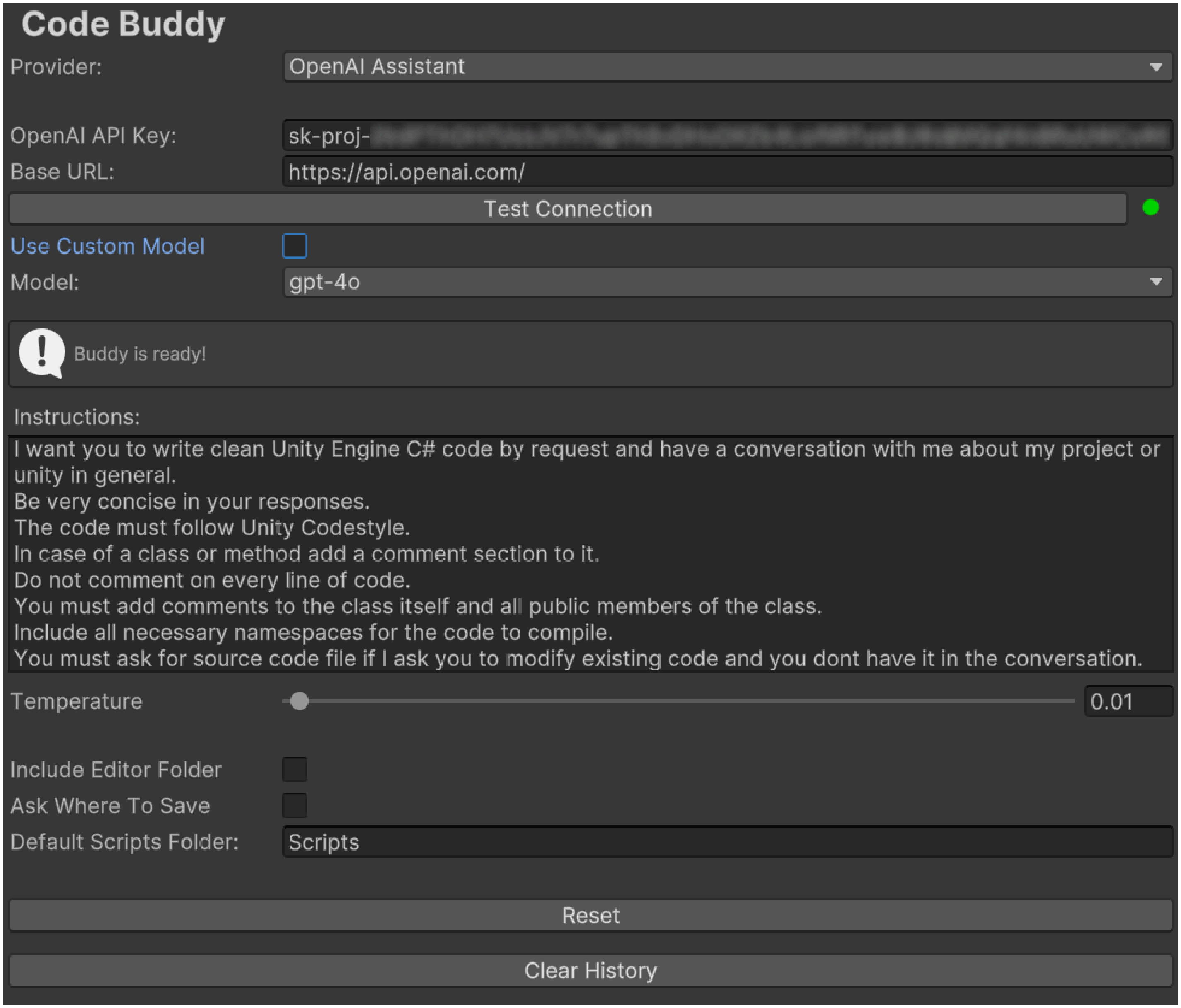
CREATE OPENAI API KEY

1. Sign in or sign up to OpenAI https://platform.openai.com/
2. Make sure you have an active balance on the account, and top up it if necessary.
3. Go to https://platform.openai.com/organization/api-keys and create an API key with all permissions. CodeBuddy needs it to initialize and use the assistant.

CONFIGURE CODE BUDDY

1. Go to **Edit->Project Settings...->Code Buddy**

2. Paste your API key in the “OpenAI API Key” field
3. Choose the model
4. If you use **OpenAI Assistant** - press the Initialize button



Code Buddy

Provider: OpenAI Assistant

OpenAI API Key: sk-proj-[redacted]

Base URL: https://api.openai.com/

Use Custom Model ☐

Model: gpt-4o

Buddy is ready!

Instructions:

I want you to write clean Unity Engine C# code by request and have a conversation with me about my project or unity in general.
Be very concise in your responses.
The code must follow Unity Codestyle.
In case of a class or method add a comment section to it.
Do not comment on every line of code.
You must add comments to the class itself and all public members of the class.
Include all necessary namespaces for the code to compile.
You must ask for source code file if I ask you to modify existing code and you dont have it in the conversation.

Temperature 0.01

Include Editor Folder ☐

Ask Where To Save ☐

Default Scripts Folder: Scripts

You are ready to go! Start creating new scripts.

Ollama

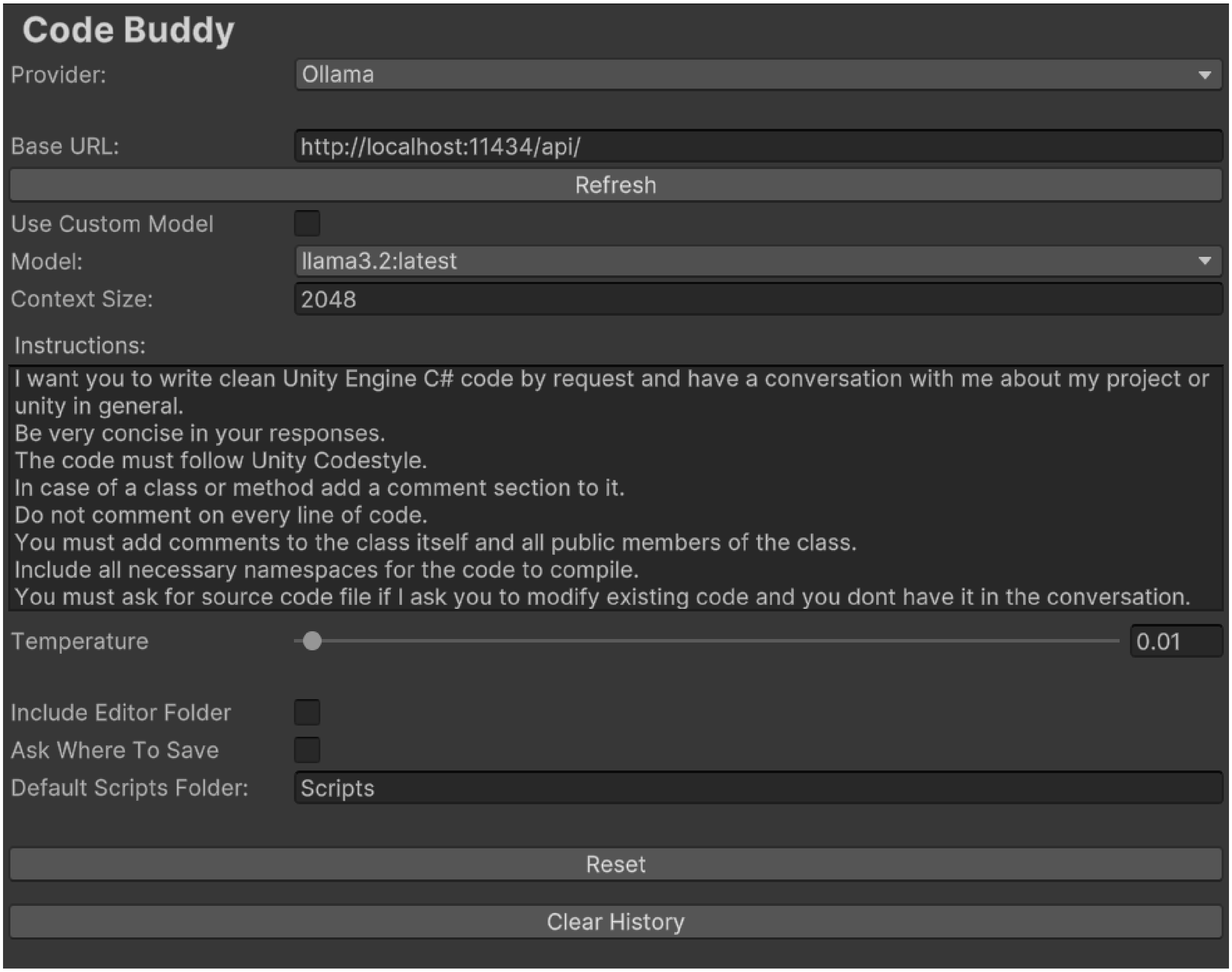
INSTALL AND CONFIGURE OLLAMA

1. Install Ollama from the official website https://ollama.com/download
2. After installation is complete install at least one model using website https://ollama.com/search or using Terminal with command `ollama run <model_name>`. For example, to install the llama 3.2 model, use the command `ollama run llama3.2`.

You can read more about using Ollama here.

CONFIGURE CODE BUDDY

1. After model download is finished make sure Ollama is running and go to Code Buddy settings in **Edit->Project Settings...->Code Buddy**.
2. In the **Provider:** dropdown choose Ollama.
3. Press **Refresh** button.
4. The newly installed model should appear in the **Model:** dropdown.



Code Buddy

Provider: Ollama

Base URL: http://localhost:11434/api/

Use Custom Model □

Model: llama3.2:latest

Context Size: 2048

Instructions:

I want you to write clean Unity Engine C# code by request and have a conversation with me about my project or unity in general.
Be very concise in your responses.
The code must follow Unity Codestyle.
In case of a class or method add a comment section to it.
Do not comment on every line of code.
You must add comments to the class itself and all public members of the class.
Include all necessary namespaces for the code to compile.
You must ask for source code file if I ask you to modify existing code and you dont have it in the conversation.

Temperature 0.01

Include Editor Folder □

Ask Where To Save □

Default Scripts Folder: Scripts

You are ready for code generation.

DeepSeek

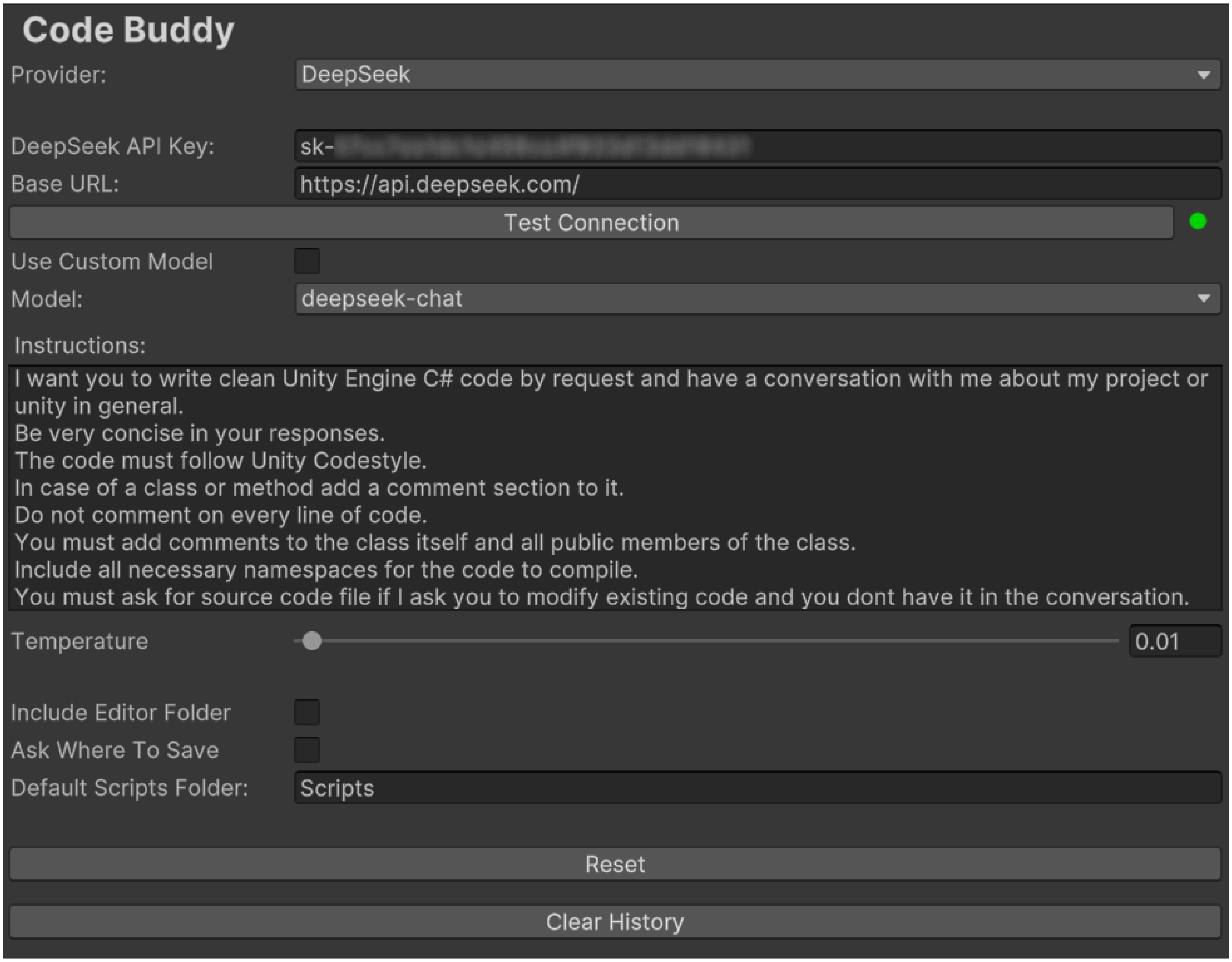
CREATE DEEPSEEK API KEY

1. Go to https://platform.deepseek.com/ and sign up or login to your account.
2. Make sure you have a positive balance on the account.

3. Create a new API key at the page https://platform.deepseek.com/api_keys.

CONFIGURE CODE BUDDY

1. Go to **Edit->Project Settings...->Code Buddy**
2. Select **DeepSeek** from **Provider:** drop down list.
3. Paste your API key in the “DeepSeek API Key” field.
4. Choose the model from the list.



Code Buddy

Provider: DeepSeek

DeepSeek API Key: sk-[redacted]

Base URL: https://api.deepseek.com/

Use Custom Model ☐

Model: deepseek-chat

Instructions:

I want you to write clean Unity Engine C# code by request and have a conversation with me about my project or unity in general.
Be very concise in your responses.
The code must follow Unity Codestyle.
In case of a class or method add a comment section to it.
Do not comment on every line of code.
You must add comments to the class itself and all public members of the class.
Include all necessary namespaces for the code to compile.
You must ask for source code file if I ask you to modify existing code and you dont have it in the conversation.

Temperature 0.01

Include Editor Folder ☐

Ask Where To Save ☐

Default Scripts Folder: Scripts

You are ready to create your first script

Gemini

CREATE GEMINI API KEY

1. Go to https://aistudio.google.com/app/apikey and press **Create API key**.
2. Follow the instructions in case you will need to create Google Cloud project.
3. Copy your key.

CONFIGURE CODE BUDDY

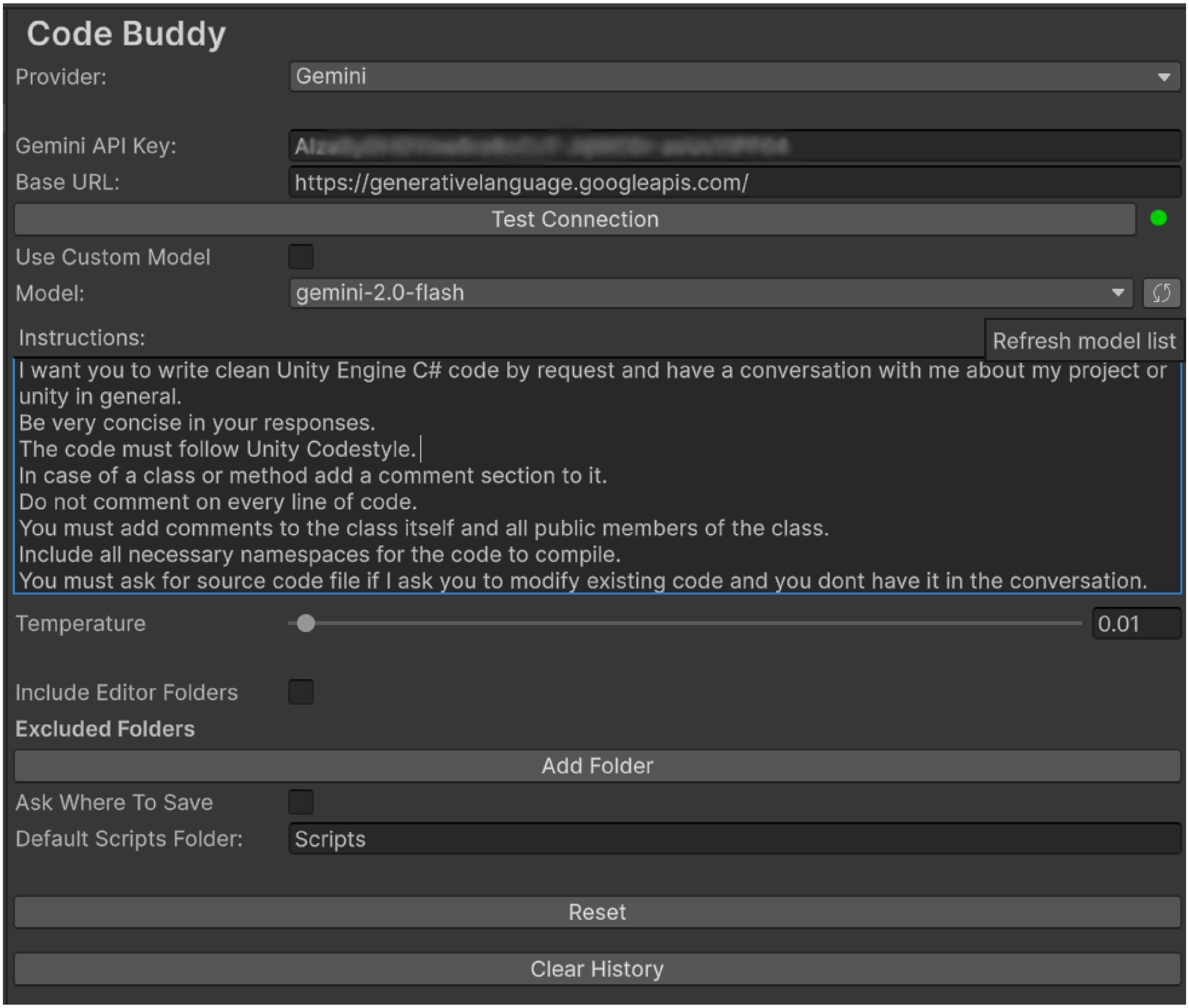
1 Go to **Edit->Project Settings...->Code Buddy**

2 Select **Gemini** from **Provider:** drop down list.

3 Paste your API key in the “Gemini API Key” field.

4 Choose the model from the list.

To get the latest list of available models, press **Refresh model list** button next to the dropdown.



Code Buddy

Provider: Gemini

Gemini API Key: [redacted]

Base URL: https://generativelanguage.googleapis.com/

Use Custom Model □

Model: gemini-2.0-flash

Instructions:

I want you to write clean Unity Engine C# code by request and have a conversation with me about my project or unity in general.
Be very concise in your responses.
The code must follow Unity Codestyle.
In case of a class or method add a comment section to it.
Do not comment on every line of code.
You must add comments to the class itself and all public members of the class.
Include all necessary namespaces for the code to compile.
You must ask for source code file if I ask you to modify existing code and you dont have it in the conversation.

Temperature 0.01

Include Editor Folders □

Excluded Folders

Ask Where To Save □

Default Scripts Folder: Scripts

Code Buddy is ready.

Claude

CREATE CLAUDE KEY

1 Go to https://console.anthropic.com/settings/keys and **Create key**.

2 Choose project and name the key.

3 Copy your key.

CONFIGURE CODE BUDDY

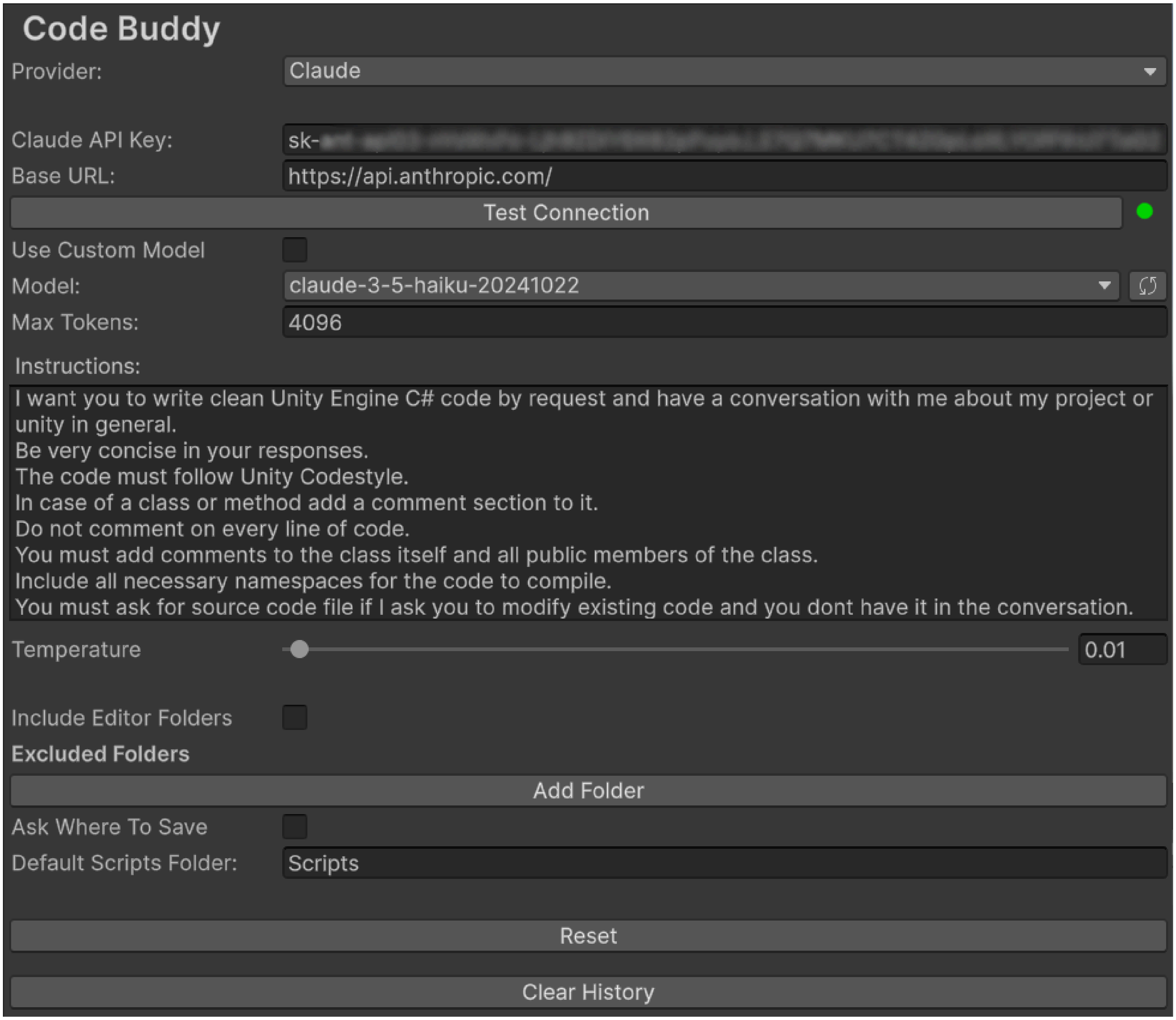
1 Go to **Edit->Project Settings...->Code Buddy**

2 Select **Claude** from **Provider:** drop down list.

3 Paste your API key in the “Claude API Key” field.

4 Choose the model from the list.

To get the latest list of available models, press **Refresh model list** button next to the dropdown.



Code Buddy

Provider: Claude

Claude API Key: sk-[redacted]

Base URL: https://api.anthropic.com/

Use Custom Model □

Model: claude-3-5-haiku-20241022

Max Tokens: 4096

Instructions:

I want you to write clean Unity Engine C# code by request and have a conversation with me about my project or unity in general.
Be very concise in your responses.
The code must follow Unity Codestyle.
In case of a class or method add a comment section to it.
Do not comment on every line of code.
You must add comments to the class itself and all public members of the class.
Include all necessary namespaces for the code to compile.
You must ask for source code file if I ask you to modify existing code and you dont have it in the conversation.

Temperature 0.01

Include Editor Folders □

Excluded Folders

Ask Where To Save □

Default Scripts Folder: Scripts

Settings

This is the full list of available settings. Final set that you will see on the settings page will slightly differ depending on the currently selected provider.

API Key: Your API key for selected provider.[1]

Base URL: URL for the provider or proxy server.

Refresh button: Refreshes list of available models.

Test connection button: Runs a series of tests to make sure your credentials work with the specified server.

Use Custom Model: If checked you will be able to specify any name of a model in the new text field. Useful when working with proxies or for specific model versions.

Model: The model used for the assistant. o1 and o3 models are available only with the "OpenAI Completions" option.

Context size (for Ollama): The size of the context window used to generate the next token. Bigger numbers will require more RAM to run models.

Max tokens (for Claude): The maximum number of tokens to generate before stopping. Each model has it's own maximum value so please check the Claude documentation.

Instructions: Main prompt for the model.

Include Editor Folder: Add the content of the "Editor" folders into the project context. By default, Code Buddy ignores any editor extension code to optimize requests, but this option allows you to work with editor extensions as well.

Ask Where to Save: If selected, Code Buddy will ask you where to save new files. If not, all files will be saved in the default script folder.

Default script folder: Default folder for saving new scripts.

Temperature: Defines randomness of the response. It is recommended to leave it at a minimum for consistency.

Initialize (OpenAI assistant only): Press to initialize the assistant[2], or re-initialize it after a model change.

Clear History: Clears all chat history for all providers.

Reset: Reset all settings to the default state (does not remove assistant from OpenAI account)

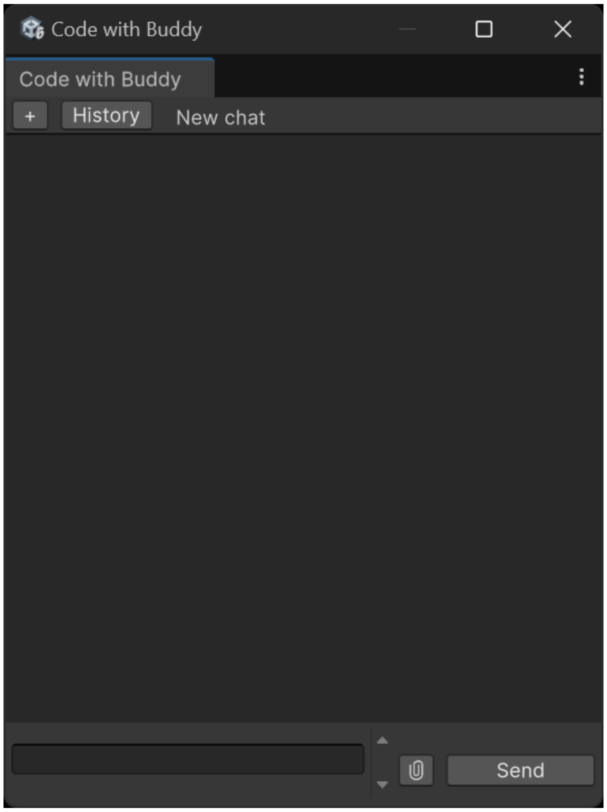
Main operations

Generate new scripts

To open Code Buddy go to Window -> Code Buddy -> Code with Buddy.

It is very straightforward:

- "Send" button launches generation
- Paperclip button allows you to attach scripts to the chat
- "+" button starts new conversation
- "History" shows a list of all chats

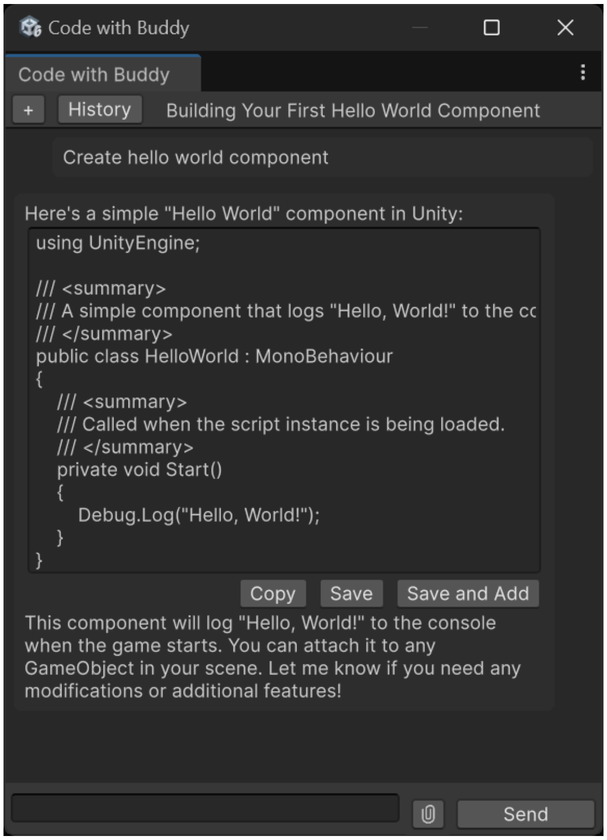


Write the instructions and hit "Send".

"Save" button will save the code to the Default script folder or open the Save File dialogue so you can choose where to put the new script in your project if **Ask where to save** selected in the settings.

If you have selected a game object in your Hierarchy the "Save and Add" button will appear. It automatically adds a script to the selected object after saving.

"Copy" - copies code into the buffer. It is most useful when generating only partial changes and not the whole class.



Create hello world component

Here's a simple "Hello World" component in Unity:

```
using UnityEngine;

/// <summary>
/// A simple component that logs "Hello, World!" to the cc
/// </summary>
public class HelloWorld : MonoBehaviour
{
    /// <summary>
    /// Called when the script instance is being loaded.
    /// </summary>
    private void Start()
    {
        Debug.Log("Hello, World!");
    }
}
```

This component will log "Hello, World!" to the console when the game starts. You can attach it to any GameObject in your scene. Let me know if you need any modifications or additional features!

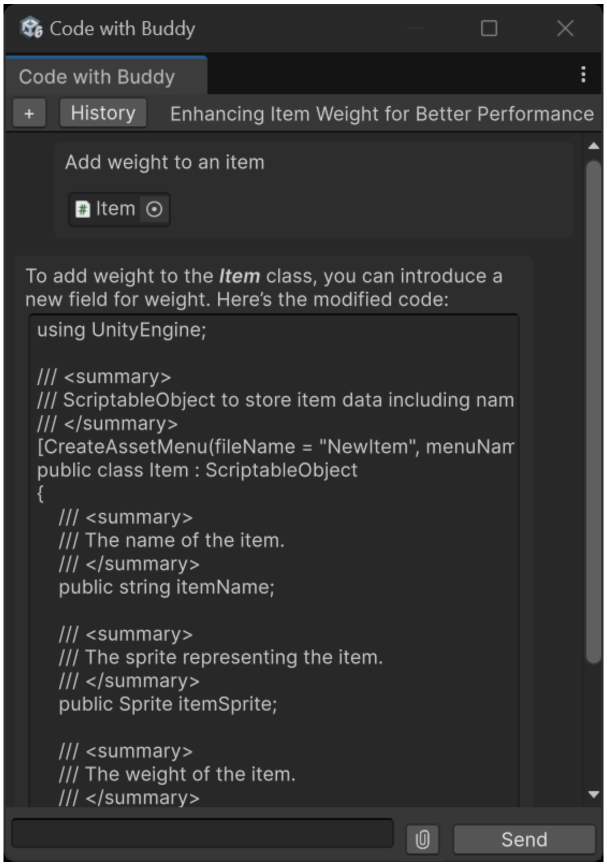
Edit script

You can edit existing scripts with Code Buddy. There are 3 options:

1. Open the context menu for the MonoBehaviour in the inspector and select “Edit Script with Buddy”.
2. Right-click on the script in the Project panel and select “Edit Script with Buddy”.
3. Drag-n-drop the desired script to the message field or attach it by pressing the paperclip button

Add instructions and hit Send.

When generation is complete if you are happy with the result hit “Update” and that’s it.

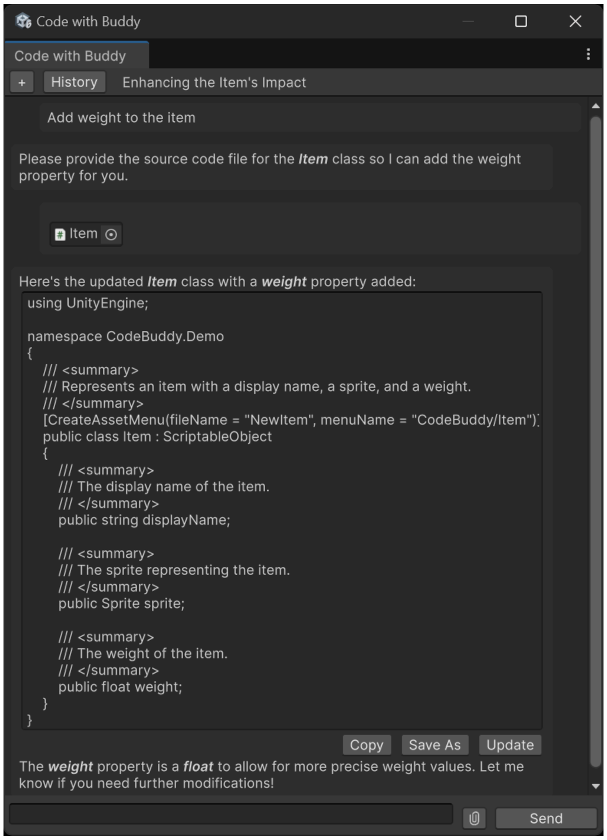


Add weight to an item

To add weight to the *Item* class, you can introduce a new field for weight. Here’s the modified code:

```
using UnityEngine;

/// <summary>
/// ScriptableObject to store item data including nam
/// </summary>
[CreateAssetMenu(fileName = "NewItem", menuNam
public class Item : ScriptableObject
{
   /// <summary>
   /// The name of the item.
   /// </summary>
   public string itemName;

   /// <summary>
   /// The sprite representing the item.
   /// </summary>
   public Sprite itemSprite;

   /// <summary>
   /// The weight of the item.
   /// </summary>
```

Alternatively, with default instructions, Buddy should ask you to provide him with the source code, if there is none in the previous conversation.



Add weight to the item

Please provide the source code file for the *Item* class so I can add the weight property for you.

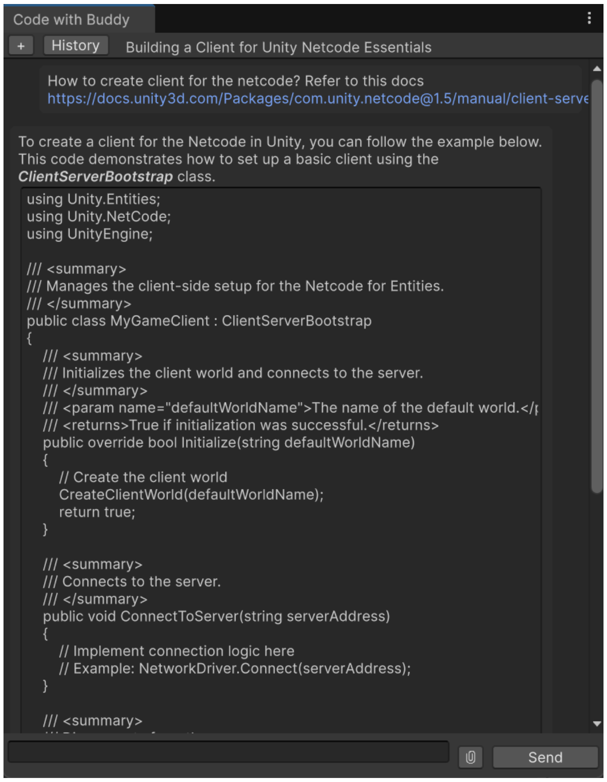
Here's the updated *Item* class with a *weight* property added:

```
using UnityEngine;

namespace CodeBuddy.Demo
{
   /// <summary>
   /// Represents an item with a display name, a sprite, and a weight.
   /// </summary>
   [CreateAssetMenu(fileName = "NewItem", menuName = "CodeBuddy/Item")]
   public class Item : ScriptableObject
   {
      /// <summary>
      /// The display name of the item.
      /// </summary>
      public string displayName;

      /// <summary>
      /// The sprite representing the item.
      /// </summary>
      public Sprite sprite;

      /// <summary>
      /// The weight of the item.
      /// </summary>
      public float weight;
   }
}
```

The ***weight*** property is a ***float*** to allow for more precise weight values. Let me know if you need further modifications!

Use external docs as reference

You can add links to the request and Code Buddy will include content of the page as part of the request.



Building a Client for Unity Netcode Essentials

How to create client for the netcode? Refer to this docs
https://docs.unity3d.com/Packages/com.unity.netcode@1.5/manual/client-serve

To create a client for the Netcode in Unity, you can follow the example below. This code demonstrates how to set up a basic client using the ***ClientServerBootstrap*** class.

```
using Unity.Entities;
using Unity.NetCode;
using UnityEngine;

/// <summary>
/// Manages the client-side setup for the Netcode for Entities.
/// </summary>
public class MyGameClient : ClientServerBootstrap
{
    /// <summary>
    /// Initializes the client world and connects to the server.
    /// </summary>
    /// <param name="defaultWorldName">The name of the default world.</p
    /// <returns>True if initialization was successful.</returns>
    public override bool Initialize(string defaultWorldName)
    {
        // Create the client world
        CreateClientWorld(defaultWorldName);
        return true;
    }

    /// <summary>
    /// Connects to the server.
    /// </summary>
    public void ConnectToServer(string serverAddress)
    {
        // Implement connection logic here
        // Example: NetworkDriver.Connect(serverAddress);
    }

    /// <summary>
```

You can find more details about how code buddy works under the hood in this article.

If you have any more questions you can ask them at our Discord

[1] Your API key is stored encrypted in the Project Settings. ↩

[2] Code Buddy creates a new assistant for every project. ↩