Step-by-Step Guide: Using Semantic Kernel with Hugging Face Inference

This document explains each step of the provided C# program that uses Semantic Kernel with Hugging Face for chat completion. The flow covers configuration, kernel setup, chat history management, interactive user input, and streaming AI responses.

## Configuration Setup

Loads settings from appsettings.json, environment variables, and user secrets using IConfiguration.

## Extract Settings

Reads values like model ID, endpoint, and API key from the configuration.

## Kernel Builder

Creates a Kernel builder instance which will manage AI services.

## Register Hugging Face Connector

Adds the Hugging Face Chat Completion service to the kernel using model ID, endpoint, and API key.

## Build the Kernel

Finalizes the kernel and locks in all registered services.

## Create Chat History

Initializes a conversation history with a system message that defines the assistant's behavior.

## Get Chat Service

Fetches the chat completion service from the kernel.

## Define Execution Settings

Configures generation settings such as Temperature (creativity) and MaxTokens (output length).

## Optional Reducer for History

Adds a reducer to trim or summarize chat history to avoid exceeding token limits.

## Print Service Attributes

Displays metadata about the configured Hugging Face service.

## Interactive Loop

Runs a loop where the user can type prompts until they exit by submitting an empty line.

## Add User Message to History

Stores the user input in the conversation memory.

## Stream AI Response

Sends the chat history to Hugging Face and streams back the AI’s reply in real-time.

## Save Assistant Reply to History

Adds the AI's generated response to the conversation memory.

## Token Usage

For Hugging Face, detailed token usage reporting is not available (unlike Azure/OpenAI).

## Apply History Reducer

Trims or summarizes chat history when it exceeds a threshold, keeping context manageable.