

Razor Pages with AI



גלעד מרקמן
פרויקט ב- Github



Semantic Kernel Tutorial

Created by [Gilad Markman](#)

Web Programming

Learn how to build AI-powered applications with Microsoft Semantic Kernel and ASP.NET Core Razor Pages

Tutorial Goals: Master Semantic Kernel integration with C# • Build interactive AI applications • Explore text, image, and audio generation



AI Chat

Interactive chat with GPT models powered by Semantic Kernel. Features web search integration with Tavily for real-time information.

✦ Features:

- GPT-4 powered conversations
- Real-time web search
- Chat history persistence
- RTL/LTR text direction support

Start Chatting



Image Generation

Create stunning AI-generated images using DALL-E 3. Transform your ideas into beautiful visual art with customizable settings.

✦ Features:

- DALL-E 3 powered generation
- Multiple size options
- HD quality settings
- Vivid & Natural styles

Generate Images



Voice Generation

Convert text to natural-sounding speech using OpenAI's TTS models. Choose from multiple voices and audio formats.

✦ Features:

- OpenAI TTS-1 model
- 6 different voice options
- Multiple audio formats
- Adjustable speech speed

Generate Speech



RAG Tutorial

Learn Retrieval Augmented Generation (RAG) with Semantic Kernel. Upload documents and chat with your data using AI embeddings and ChromaDB vector database.

✦ Features:

- Document upload & processing
- AI-powered embeddings
- ChromaDB vector database
- Semantic search
- Chat with your documents

Start RAG Chat



Summarization


Automatically summarize long texts, documents, and articles using AI. Extract key insights and main points efficiently.

✦ Features:

- AI-powered text summarization
- Multiple summary lengths
- Key points extraction
- Document processing

Summarize Documents

Chat AI

 Semantic Kernel Chat Image Generation Voice Generation Summarization RAG

Toggle RTL/LTR

Chat with GPT

.assistant: You are a helpful assistant. Use Tavily.Search when you need fresh info and cite links

? **user:** Hi, When the WWW2 started and why

assistant: World War II (WWII) started on September 1, 1939. The primary cause of the war was the aggressive expansionist policies of Nazi Germany, led by Adolf Hitler. Germany invaded Poland, prompting Britain and France to declare war on Germany. The conflict soon expanded as more countries got involved, resulting in a global war. The underlying reasons for the war included unresolved issues from World War I, economic instability, and the rise of totalitarian regimes in Germany, Italy, and Japan

user: מתי החלה מלחמת העולם השנייה ומדוע ?

:assistant

assistant [{"Name": "מלחמת העולם השנייה - אנציקלופדיה", "Value": "https://snopi.com/xenc/enc.aspx?id=82", "Link": "מלחמת העולם השנייה. החלה ב-3 בספטמבר 1939, כאשר הכריזו בריטניה הגדולה וצרפת מלחמה על גרמניה הנאצית, שפלשה לפולין יומיים לפני כן. היא הסתיימה ב-15 באוגוסט"}, {"Name": "מלחמת העולם השנייה - ויקיפדיה", "Value": "https://he.wikipedia.org/wiki/%D7%9E%D7%9C%D7%97%D7%9E%D7%AA_%D7%94%D7%A2%D7%95%D7%9C%D7%9D_%D7%94%D7%A9%D7%A0%D7%99", "Link": "%D7%99%D7%94", "Value": "בזירה האירופית החלה המלחמה ב-1 בספטמבר 1939 עם פלישת גרמניה הנאצית לפולין, והסתיימה ב-8 במאי 1945 עם חתימת גרמניה הנאצית על חוזה הכניעה הסופי. בזירה האסייתית"}]

assistant: מלחמת העולם השנייה החלה ב-1 בספטמבר 1939, כאשר גרמניה הנאצית פלשה לפולין. בעקבות זאת, בריטניה וצרפת הכריזו מלחמה על גרמניה ב-3 בספטמבר 1939. המלחמה החלה בגלל המדיניות האגרסיבית של גרמניה תחת הנהגתו של אדולף היטלר, כמו גם מחלוקות לא פתורות מהמלחמה הראשונה, משברים כלכליים ועליית משטרים טוטליטריים

Send

...Type your message

Image Generator

AI Image Generator

Image generated successfully! Prompt: "תמונה של רובוט משחק עם כלב על הדשא מתחת לעץ ירוק. השמיים כחולים והשמש זורחת. עננים רכים בשמיים וקצת ציפורים. הציור צריך להיות בסגנון ריאלי

What image do you want to generate?

תמונה של רובוט משחק עם כלב על הדשא מתחת לעץ ירוק. השמיים כחולים והשמש זורחת. עננים רכים בשמיים וקצת ציפורים. הציור צריך להיות בסגנון ריאלי

Size


1024 x 1024 (Square)

Quality

HD (High Definition)

Style

Vivid


 Generate Image

Generated Image:



Prompt: "תמונה של רובוט משחק עם כלב על הדשא מתחת לעץ ירוק. השמיים כחולים והשמש זורחת. עננים רכים בשמיים וקצת ציפורים. הציור צריך להיות בסגנון ריאלי

Voice Generator

 **AI Voice Generator**

Audio generated successfully! Prompt: "מה שימך. קוראים לי גלעד. שלום, קוראים לי גלעד. מה שימך?"

What text do you want to convert to voice?

מה שימך. קוראים לי גלעד. שלום, קוראים לי גלעד. מה שימך?

Voice


Format

Speed

Nova

MP3

1

 **Generate Voice**

Generated Audio:

▶ 0:02 / 0:02

🔊 ⋮

Prompt: "מה שימך. קוראים לי גלעד. שלום, קוראים לי גלעד. מה שימך?"

Summarization

AI PDF Summarizer

PDF summarized successfully! Original file: Lucas A. Meyer - A Quick Tour of the Semantic Kernel.pdf

Select PDF file to summarize

Choose File

No file chosen

Upload a PDF document for AI-powered summarization.

Summary Language

English

Pages per Chunk

5

Group Size

5

How many pages to process together

Number of summaries to merge at once


Summarize PDF

Summary Result:

Download Summary


- Microsoft Semantic Kernel (SK) is a thin, open-source SDK designed to facilitate application integration with AI services, initially developed to power Microsoft Copilots like Microsoft 365 and Bing.
- Originally written in C#, SK has been extended to Python and Java, with Python examples available on GitHub.
- Running SK code requires an Azure subscription with OpenAI API access or a direct OpenAI subscription; Azure offers a free trial but OpenAI services on Azure are paid.
- SK is designed for enterprise or large-scale consumer applications, providing more customization and control than alternatives like LangChain, though it requires more coding effort.
- Language support varies: Python supports both LangChain and SK; JavaScript supports LangChain but not officially SK (community TypeScript API exists); Java and .NET support SK but not LangChain.
- The Semantic Kernel acts as a lightweight object to which AI-related functions and services are attached.
- Connectors link SK to multiple AI services simultaneously, allowing complex workflows using different models; for example, an Azure subscription can deploy both "gpt35" (GPT-3.5 Turbo) and "gpt4" (GPT-4) models loaded into the same kernel.
- Semantic functions are LLM-powered functions performing specific tasks using prompt templates with placeholders replaced by input parameters; e.g., a semantic function generating knock-knock jokes.
- Native functions are regular Python functions performing tasks without LLMs, such as image classification using the timm library; these are decorated with `@sk_function` and imported into the kernel as native plugins.`
- Plugins are collections of functions and a key strength of SK; they come in two types: semantic plugins (collections of semantic functions) and native plugins

RAG

 Semantic Kernel Chat Image Generation Voice Generation Summarization RAG

Document Management


Choose File No file chosen

 Load
PDF

 Clear Documents

Select a PDF file to upload and index for RAG chat.

AI Chat (General Mode)

 Clear Chat

```
import semantic_kernel as sk
```

```
kernel = sk.Kernel()
```

Connectors


Connectors are the way you connect to AI services. You can connect multiple services to the same kernel, which allows you to perform a complex task using different services for each step.

Based on this context, I will now provide an answer. If the answer cannot be found in the provided context, I will indicate that the information is not available in the documents.

 **AI:**

Based on the provided context, the Semantic Kernel is a framework or tool (from Microsoft) that allows creating plugins, which are collections of functions imported into the kernel. It supports semantic plugins, which are collections of semantic functions, each organized in its own directory. The Semantic Kernel uses connectors to connect to AI services, allowing the use of multiple AI services within the same kernel to perform complex tasks.

Ask me anything...

 Send

