USAi API guidance

**Note:** Within USAi, users will be able to see partner-specific API instructions. Please fill in the yellow box below, and email us at [partnerships@usai.gov](mailto:partnerships@usai.gov) so we can incorporate this agency-specific guidance into your tenant implementation.

# Getting started

## Introduction

USAi API provides programmatic access to AI services for government users and approved partners. It currently supports Chat Completions (model inference) and Embeddings (used for RAG and other applications).

Our API has resource-oriented URLs, accepts JSON request bodies, returns JSON responses, and uses HTTP response codes to indicate API errors, authentication messaging, and verbs.

## Models and endpoints

USAi AI API is organized around [REST](http://en.wikipedia.org/wiki/Representational_State_Transfer) and provides access to the large language models (LLMs):

**Google AI models**

* [Gemini 2.5 Flash](https://ai.google.dev/gemini-api/docs/models#gemini-2.5-flash)
* [Gemini 2.5 Pro](https://ai.google.dev/gemini-api/docs/models#gemini-2.5-pro)

**Anthropic models**

* [Claude Haiku 3.5](https://docs.claude.com/en/docs/overview)
* [Claude Sonnet 3.7](https://docs.claude.com/en/docs/overview)
* [Claude Sonnet 4](https://docs.claude.com/en/docs/overview)
* [Claude Opus 4](https://docs.claude.com/en/docs/overview)

**Meta models**

* [Llama 3.2 11B](https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_2/)
* [Llama 4 Maverick](https://www.llama.com/docs/model-cards-and-prompt-formats/llama4/)

The interface is modeled after the [OpenAI Chat Completion API](https://platform.openai.com/docs/guides/text?api-mode=responses). We will continue to add more models over time; we may also remove models if they are found to not meet our standards. You will be notified if a model is removed.

Each model we draw on has unique capabilities. Models may also respond differently to the same API request. Below is a summary of each endpoint’s functionality:

* Chat Completions: The ability to send prompts and receive a response from the LLM models.
* Embedding: Converts input into numeric vectors representing semantic meaning. The vectors are used for tasks like retrieving information from documents, document analysis, and building Retrieval Augmented Generation (RAG) systems.
* Models: A list of models and IDs used to specify LLM models in requests.

### Content types

The models USAi API draws on support multiple types of inputs; text, image, and file content types. However, each model has different input capabilities: for example, Claude Sonnet supports Optical Character Recognition (OCR) and recognizes image-only PDFs, while Claude Haiku and LLaMa currently do not. For additional detail about how we handle these content types, reference [OpenAI’s API documentation](https://platform.openai.com/docs/api-reference/chat/create).

## Authentication

All API requests require an API key.

| Insert agency-specific guidance here. For example:  To request an API key at GSA, please [navigate to ServiceNow [Link], open a Jira ticket [Link], or email XXX@agency.gov. |
| --- |

API keys will rotate every 24 hours, and you will have the ability to revoke, rotate, or delete an API key.

After you’re granted access, API keys will be available by clicking on API in the sidebar, and navigating to “API key management.”

## Limitations

### Rate limitations

We currently have a rate limit of 3 calls / second / API key. If you hit a rate limit, you should expect a 429 error code. If you need additional capacity, please contact us at support@usai.gov. We can work with our infrastructure providers to secure higher model limits.

### Feature limitations

Because the underlying platforms have different capabilities and interfaces, not all features of the Chat Completions API are available at this time. Current known limitations include:

* Audio
* Structured output

### Guardrail limitations

We do not have guardrails beyond those provided by our model providers; this means that an API user has full control over what inputs and outputs are allowed when using the API. We expect API users to interact with the API ethically and deliberately, and add their own guardrails and system prompts as needed. We recommend implementing system prompts when using the API in situations where you do not know what inputs the model will receive.

System prompts should address your needs and concerns. Some of the system prompts in USAi Chat include:

* You are a helpful assistant that works for a government agency.
* You help users with general knowledge, problem-solving, coding, and interactive tasks.
* You maintain a friendly, helpful, professional, and empathetic tone at all times.
* You want to understand the user's intent, and apply your knowledge and background to formulate the most helpful response possible.
* Redirect conversations that veer into inappropriate, illegal, or explicit territory.
* You're not an expert in government policies, security, safety, health, procurement, contracts, or law. Provide general guidance only and advise users to reference appropriate material.
* Prioritize historical accuracy, scientific inquiry, and objectivity in all responses.
* Break down complex questions and walk users through solutions step-by-step.
* Use real-world analogies to simplify complex concepts.
* When the user's request is unclear, ask for more details to help refine your response.
* Ask users for feedback on the answer that can help you respond more accurately.
* Never knowingly make false statements or deceive users.
* Avoid generating explicit, hateful, dangerous, or illegal content.
* Protect privacy and do not share personal information about individuals.
* Redirect users' requests around potentially controversial or polarizing topics quickly.
* You do not prefer or recommend specific political views, groups, religions, companies, products, or enterprise.

## Code demo

To see an example of how to implement features using Python, visit our HTML example notebook. Another option is to [download the Jupyter notebook].

# Endpoints

API endpoints will be available when you login.

The API endpoint base URL is reachable at api.[agency].[usai.gov](http://usai.gov).

USAi API is built using the OpenAI standard; the endpoints echo OpenAI’s format and structure. For the most up to date, detailed information, visit theSwagger documentation at api.[agency].usai.gov/docs.

## 1. Models

GET /api/v1/models

To reach models the documented endpoint is <base url>/api/v1/models, which will allow you to retrieve available model information.

#### Example Request

curl -X 'GET' \  
 '<base url>/api/v1/models' \  
 -H 'accept: application/json' \  
 -H 'Authorization: Bearer <Your API Key>'

#### Example Response

[  
 {  
 "name": "Claude 3.7 Sonnet",  
 "id": "claude\_3\_7\_sonnet",  
 "capability": "chat"  
 },  
 {  
 "name": "Claude 3 Haiku",  
 "id": "claude\_3\_haiku",  
 "capability": "chat"  
 },  
 {  
 "name": "Llama 3.2 11B",  
 "id": "llama3211b",  
 "capability": "chat"  
 },  
 {  
 "name": "Cohere English Embeddings",  
 "id": "cohere\_english\_v3",  
 "capability": "embedding"  
 }

]

## 2. Chat Completions

POST /api/v1/chat/completions

To reach chat completions the documented endpoint is <base url>/api/v1/chat/completions, which will allow you to retrieve chat completion information.

#### Request Body

* model: The model ID (e.g: gemini-2.0-flash, claude\_3\_haiku)
* messages: An array of of message items consisting of
  + User message, Image Content, Document Content, Assistant Message
* max\_tokens: Maximum response length (optional)
* temperature: Response creativity (0.0-2.0, optional)
* top\_p: An alternative to sampling with temperature, called nucleus sampling (optional)
* stream: a boolean indicating whether to send partials responses as available (optional)

#### Example Request

curl -X 'POST' \  
 '<base url>/api/v1/chat/completions' \  
 -H 'accept: application/json' \  
 -H 'Authorization: Bearer <Your API Key>' \  
 -H 'Content-Type: application/json' \

-d '{  
 "messages": [  
 {  
 "content": "You speak only pirate",  
 "role": "system"  
 },  
 {  
 "content": "Hello!",  
 "role": "user"  
 }  
 ],  
 "model": "gemini-2.5-flash"  
 }'

#### **Example Response** {

"object": "chat.completion",  
 "created": 1748517745,  
 "model": "gemini-2.5-flash",  
 "choices": [  
 {  
 "index": 0,  
 "message": {  
 "role": "assistant",  
 "content": "Ahoy there, matey! What brings ye to me waters?\n"  
 },  
 "finish\_reason": "stop"  
 }  
 ],  
 "usage": {  
 "prompt\_tokens": 14,  
 "completion\_tokens": 15,  
 "total\_tokens": 29  
 }  
}

## 3. Embeddings

POST /api/v1/embeddings

To reach embeddings the documented endpoint is <base url>/api/v1/embeddings, which will allow you to retrieve embeddings information.

#### Request Body

* model: The model ID (e.g: cohere\_english\_v3)
* input: Input text to embed, encoded as a string or array of strings. Each input must not exceed the max input tokens for the model.
* dimensions: The number of dimensions the resulting output embeddings should have. Only supported in some models (Optional)
* input\_type: (Note: this is not part of the OpenAI specification but is useful on some models). Specify the kind of input to allow the model to optimize for specific uses. Options are: "search\_document", "search\_query", "classification", "clustering", "semantic\_similarity" (Optional)

#### Example Request

curl -X 'POST' \  
 '<base url>/api/v1/embeddings' \  
 -H 'accept: application/json' \  
 -H 'Authorization: Bearer <Your API Key>\  
 -H 'Content-Type: application/json' \  
 -d '{  
 "encodingFormat": "float",  
 "input": "A mighty woman with a torch, whose flame / Is the imprisoned lightning",  
 "input\_type": "search\_document",  
 "model": "cohere\_english\_v3"  
}

#### Example Response

{  
 "object": "list",  
 "data": [  
 {  
 "object": "embedding",  
 "embedding": [  
 0.06933594,  
 -0.030883789,  
 0.054351807,  
 -0.0018196106,  
 …  
 0.013938904  
 ],  
 "index": 0  
 }  
 ],  
 "model": "cohere.embed-english-v3",  
 "usage": {  
 "promptTokens": 12,  
 "totalTokens": 12  
 }  
}

# Support

## Support

You may encounter issues while working with the USAi API, as it is under active development. We may not be able to fix all issues immediately, but will add them to our backlog for future releases. If you have any questions, hit rate limits, or otherwise need assistance, please contact us at [support@usai.gov](mailto:support@usai.gov).

—-------------------------------------------------------------------------------------------------