

Open Arena API User Guide

Contents

- Introduction
- Swagger Documentation
- Authentication
- Making Requests
- Endpoints
- Inference / Prediction using a Chain
- Using the Streaming Endpoint for Inference
- Conclusion

Welcome to the Open Arena API User Guide. Our objective is to provide you with a clear and concise manual to facilitate your interaction with our API, enabling you to leverage the power of Large Language Models (LLMs) seamlessly.

Introduction

Open Arena offers a robust API that serves as a conduit to our advanced Large Language Models (LLMs). This guide is crafted to assist you in navigating through the API's functionalities, ensuring that you can integrate and utilize the LLMs effectively in your applications.

Swagger Documentation

The Open Arena API is documented using Swagger, which provides a comprehensive overview of the available endpoints, request parameters, and response structures.

You can access the Swagger documentation by clicking on the following link: [Open Arena API Swagger Documentation](#).

Authentication

Obtaining Your ESSO Token

To ensure secure interaction with the Open Arena API, an ESSO token is required for authenticating your requests. Follow the step-by-step guide below to acquire your authentication token.

Step 1: Access the AI Platform

Navigate to the Open Arena AI Platform by clicking on the following link: [Open Arena AI Platform](#). You will be prompted to enter your login credentials. Please proceed to log in.

Step 2: Open Developer Tools

Once logged in, you will need to access the Developer Tools in your browser to inspect network traffic:

- **For Windows Users:** Press `Ctrl+Shift+I`.

[Skip to main content](#)

Alternatively, you can right-click on the webpage and select 'Inspect Element' from the context menu.

Step 3: Inspect Network Activity

With the Developer Tools open, navigate to the 'Network' tab. This section captures all network activity between your browser and the server.

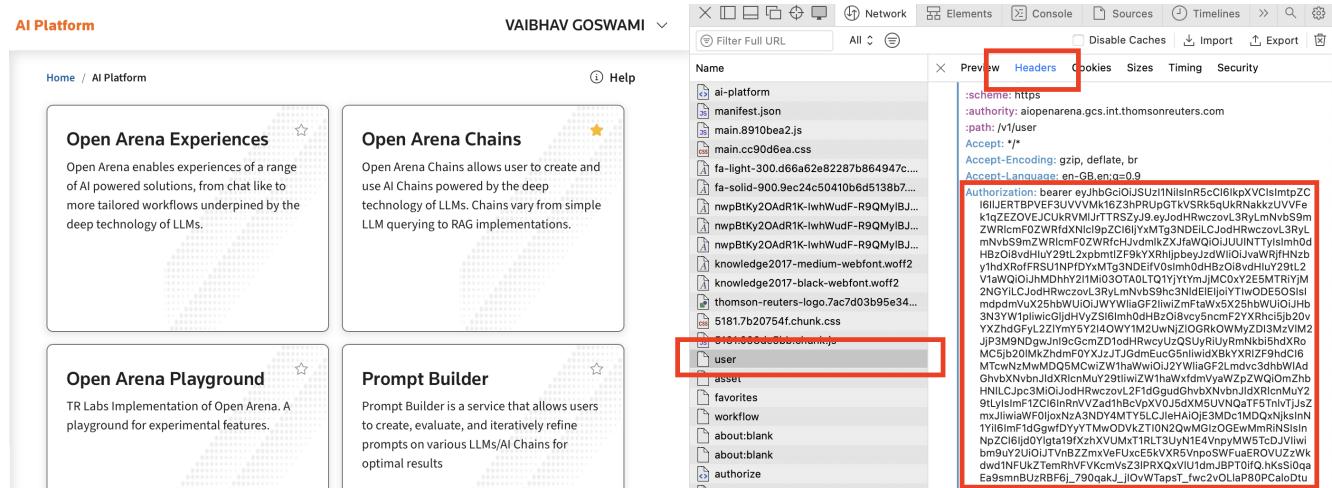
Step 4: Trigger a Network Request

If no network activity is displayed, you may need to initiate a network request. This can be done by simply refreshing the webpage. Look for a request in the list named 'user'.

Step 5: Locate Your ESSO Token

Click on the 'user' request to view its details. Within the details pane, switch to the 'Headers' tab. Scroll down to the 'Request Headers' section, and you will find the authorization token listed there.

Important Security Note: Your ESSO token is a critical piece of information akin to a password. Protect it diligently and avoid sharing it with any unauthorized parties.



Making Requests

Base URL

The base URL for the Open Arena API is as follows:

<https://aiopenarena.gcs.int.thomsonreuters.com/>

Request Headers

When making requests to the Open Arena API, you will need to include the ESSO token in the request headers. The token should be passed in the `Authorization` header as shown below:

Authorization: Bearer <your_essos_token>

Endpoints

The Open Arena API offers several endpoints, each catering to specific service capabilities. Below are the available endpoints:

User

- **Retrieve User Data:**
 - `GET /v1/user` - Fetches details associated with a specific user's Asset ID.
- **Create New User:**
 - `POST /v1/user` - This endpoint is designed to create a new user in the system. When invoking this endpoint, you will need to provide a JSON payload that includes the user's asset ID, organization ID, and department ID. Ensure that your JSON payload is properly structured as shown in the example below:

```
{  
  "asset_id": "<your_asset_id>",  
  "organisation_id": "<your_organisation_id>",  
  "department_id": "<your_department_id>"  
}
```

Asset Information

- **List Assets:**
 - `GET /v1/asset` - Provides a list of all available asset IDs.

Workflow Operations

- **List Workflows:**
 - `GET /v2/workflow` - Retrieves a list of workflows with optional query parameters:
 - `show_all`: When set to `true`, returns all workflows.
 - `created_by_me`: When set to `true` and `show_all` is `false`, returns workflows created by the user.
- **Workflow Details:**
 - `GET /v1/workflow{/workflow_id}` - Obtains detailed information for a specific workflow identified by its unique ID, which can be found using the `/v2/workflow` endpoint or within the UI.

Remember to replace placeholder tokens and IDs with actual values when making your requests. Each endpoint serves a distinct purpose and requires appropriate handling as per your application's needs.

Inference / Prediction using a Chain

POST /v1/inference

This endpoint allows you to perform inference using a chain. Below are the different ways you can use this endpoint:

- **Infer using Chain:**

```
{  
  "workflow_id": "80f448d2-fd59-440f-ba24-ebc3014e1fdf",  
  "query": "hi",  
  "is_persistence_allowed": false  
}
```

The `workflow_id` is the actual workflow ID that you want to use for inference for **Open AI GPT 4 Turbo** chain. The `query` is the input text for which you want to generate the inference. The `is_persistence_allowed` flag is used to indicate whether the results should be persisted or not into the users account. **Details:**

`workflow_id`: Identifier for the workflow. `query`: Text input for inference. `is_persistence_allowed`: Flag to allow result persistence.

Response:

- The response includes the inference result. Look for the 'answer' field in the response JSON.

Infer using Chain with System Prompt:

- `POST /v1/inference` - This endpoint is used to perform inference using a chain with a system prompt. The request body should contain a JSON payload with the following structure:

```
{  
  "workflow_id": "80f448d2-fd59-440f-ba24-ebc3014e1fdf",  
  "query": "hi",  
  "is_persistence_allowed": false,  
  "modelparams": {  
    "openai_gpt-4-turbo": {  
      "system_prompt": "You are an experienced Software Developer. Respond in a professional manner."  
    }  
  }  
}
```

The `workflow_id` is the actual workflow ID that you want to use for inference for **Open AI GPT 4 Turbo** chain. The `query` is the input text for which you want to generate the inference. The `is_persistence_allowed` flag is used to indicate whether the results should be persisted or not into the users account. The `modelparams` section is used to provide the system prompt for the **Open AI GPT 4 Turbo** chain.

- **Infer using Chain with Context:**

- `POST /v1/inference` - This endpoint is used to perform inference using a chain with context. The request body should contain a JSON payload with the following structure:

```
{  
  "workflow_id": "80f448d2-fd59-440f-ba24-ebc3014e1fdf",  
  "query": "hi",  
  "is_persistence_allowed": false,  
  "context": "This is the context for the query."  
}
```

The `workflow_id` is the actual workflow ID that you want to use for inference for **Open AI GPT 4 Turbo** chain. The `query` is the input text for which you want to generate the inference. The `is_persistence_allowed` flag is used to indicate whether the results should be persisted or not into the users account. The `context` section is used to provide the context for the **Open AI GPT 4 Turbo** chain.

- **Infer using Chain with Model Parameters:**

- `POST /v1/inference` - This endpoint is used to perform inference using a chain with model parameters. The request body should contain a JSON payload with the following structure: **Model Parameters:**

Every model has its own set of parameters that can be used to control the inference process.

The `modelparams` section is used to provide these parameters for the **Open AI GPT 4 Turbo** chain. The set of parameters that can be used for the **Open AI GPT 4 Turbo** chain are:

- `temperature`: Controls randomness. Lowering results in less randomness.
- `top_p`: Controls diversity. Lowering results in less diversity.
- `frequency_penalty`: Controls repetition. Lowering results in less repetition.

- `max_tokens`: Controls the maximum number of tokens to generate.
- `presence_penalty`: Controls the presence of entities. Lowering results in less presence of entities.

```
{
  "workflow_id": "80f448d2-fd59-440f-ba24-ebc3014e1fdf",
  "query": "hi",
  "is_persistence_allowed": false,
  "modelparams": {
    "openai_gpt-4-turbo": {
      "temperature": "0.7",
      "top_p": "0.9",
      "frequency_penalty": "0",
      "system_prompt": "You are a helpful, respectful and honest assistant.",
      "max_tokens": "800",
      "presence_penalty": "0"
    }
  }
}
```

To get the list of all available LLMs and their model parameters, use the `GET /v1/components` endpoint with the `task_id` query parameter set to `query`.

Tasks and Components

- **List Tasks:**
 - `GET /v1/tasks` - Retrieves a list of all available tasks.
- **List Components:**
 - `GET /v1/components` - Retrieves a list of all available components for a task.

Using the Streaming Endpoint for Inference

- **Infer using Streaming Endpoint:**

```
wscat -c wss://wymocw0zke.execute-api.us-east-1.amazonaws.com/prod?Authorization=<your_esso_token>
```

Example command

```
'wscat -c  
wss://wymocw0zke.execute-api.us-east-1.amazonaws.com/prod?Authorization=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVC....'
```

This command is used to connect to the streaming endpoint for inference. The request body should contain a JSON payload with the following structure:

```
{
  "action": "SendMessage",
  "workflow_id": "80f448d2-fd59-440f-ba24-ebc3014e1fdf",
  "query": "hi",
  "is_persistence_allowed": false
}
```

Please note that the `Authorization` query parameter in the URL should contain your same ESSO token without the `Bearer` keyword.

Conclusion

This concludes the Open Arena API User Guide. We hope that this guide has provided you with a comprehensive understanding of the API's capabilities and how to effectively utilize them. Should you have any further queries or require assistance, please feel free to reach out to our support team. We are committed to ensuring that your experience with Open Arena is seamless and

productive.

Connect with us at [Support Channel](#) for any assistance or queries.

< Previous

[Open Arena UI User Guide](#)

Next >

[Prompt Builder](#)