

Unified AI Platform API Guide

API DOCUMENTATION

KAMAL RAJ

Contents

Overview	1
Authentication	1
Endpoints	1
Deploy Model	1
Get Model Status.....	2
List All Models	3
Delete Model.....	4
Error Codes	5
Best Practices.....	5
Additional Resources	6

Overview

The Unified AI Platform API allows you to interact with the platform's features programmatically. Use the API to deploy models, monitor their status, and retrieve performance metrics.

Note: Before using the API, ensure that you have a valid API key for secure access.

Authentication

Every request to the Unified AI Platform API requires authentication. Include your API key in the `X-API-Key` header for every request to ensure secure access.

Example Request:

```
GET /api/v1/models HTTP/1.1
Host: api.unified-ai-platform.com
X-API-Key: YOUR_API_KEY
```

cURL Example:

```
curl -H "X-API-Key: YOUR_API_KEY" https://api.unified-ai-
platform.com/api/v1/models
```

Python Example:

```
import requests
url = "https://api.unified-ai-platform.com/api/v1/models"
headers = {
    "X-API-Key": "YOUR_API_KEY"
}
response = requests.get(url, headers=headers)
print(response.json())
```

Endpoints

The Unified AI Platform API provides several endpoints to manage your models. This section describes each endpoint, including its usage, parameters, and examples.

Deploy Model

Use this endpoint to deploy a new model to the platform.

The `POST /api/v1/models/deploy` endpoint accepts the model's name and file to deploy it for real-time predictions.

- **HTTP Method:** POST
- **URL:** `/api/v1/models/deploy`

- **Headers:**

- Content-Type: application/json
- X-API-Key: YOUR_API_KEY

- **Body Parameters:**

- `model_name` (string, required): The name you want to assign to the model.
- `model_file` (file, required): The model file in `.h5` (TensorFlow) or `.pt` (PyTorch) format.

Request Example:

Use the following JSON to deploy a model:

```
{
  "model_name": "sentiment-analysis",
  "model_file": "sentiment_model.h5"
}
```

Response Example:

- **Success:** Returns a `201 Created` status code if the model is deployed successfully.

```
{
  "status": "success",
  "message": "Model deployed successfully.",
  "model_id": "12345"
}
```

- **Error:** Returns a `400 Bad Request` status code if there is an issue with the model file format.

```
{
  "status": "error",
  "message": "Invalid model file format."
}
```

Get Model Status

Use this endpoint to check the status of a specific model.

The `GET /api/v1/models/{model_id}/status` endpoint provides details such as the model's deployment status and performance metrics.

- **HTTP Method:** GET
- **URL:** `/api/v1/models/{model_id}/status`
- **Headers:**

- X-API-Key: YOUR_API_KEY

- **Path Parameters:**
 - `model_id` (string, required): The unique identifier of the model.

Response Example:

- **Success:** Returns a `200 OK` status code along with the model's status and performance details.

```
{
  "model_id": "12345",
  "status": "deployed",
  "performance": {
    "accuracy": 0.95,
    "latency": "200ms"
  }
}
```

- **Error:** Returns a `404 Not Found` status code if the model ID does not exist.

```
{
  "status": "error",
  "message": "Model not found."
}
```

List All Models

This endpoint allows you to retrieve a list of all models deployed under your account.

The `GET /api/v1/models` endpoint provides an overview of each model, including its name, status, and deployment date.

- **HTTP Method:** `GET`
- **URL:** `/api/v1/models`
- **Headers:**
 - `X-API-Key: YOUR_API_KEY`

Response Example:

- **Success:** Returns a `200 OK` status code along with a list of deployed models.

```
{
  "models": [
    {
      "model_id": "12345",
      "model_name": "sentiment-analysis",
      "status": "deployed",

```

```

    "deployed_at": "2024-01-01T12:00:00Z"
  },
  {
    "model_id": "67890",
    "model_name": "image-classification",
    "status": "training",
    "deployed_at": "2024-02-10T15:30:00Z"
  }
]

```

- **Error:** Returns a **401 Unauthorized** status code if the API key is invalid.

```

{
  "status": "error",
  "message": "Invalid API key."
}

```

Delete Model

Use this endpoint to delete a specific model from the platform.

The **DELETE /api/v1/models/{model_id}** endpoint removes the model and its associated data.

- **HTTP Method:** DELETE
- **URL:** /api/v1/models/{model_id}
- **Headers:**
 - X-API-Key: YOUR_API_KEY
- **Path Parameters:**
 - **model_id** (string, required): The unique identifier of the model to delete.

Response Example:

- **Success:** Returns a **204 No Content** status code, indicating the model was deleted successfully.
- **Error:** Returns a **404 Not Found** status code if the model ID does not exist.

```

{
  "status": "error",
  "message": "Model not found."
}

```

Error Codes

The API uses standard HTTP status codes to indicate the success or failure of a request. This section outlines the most common error codes you may encounter while using the API, along with possible causes and suggested solutions.

- **200 OK:** The request was successful, and the server returned the requested information. No further action is needed.
- **201 Created:** The resource was successfully created. For example, a new model deployment request has been completed, and the model is now available for use.
- **204 No Content:** The request was successful, but there is no content to return. Typically used for DELETE requests when a model is successfully deleted.
- **400 Bad Request:** The request contains incorrect or invalid parameters. This can occur if the request body is malformed, required fields are missing, or an invalid file format is provided, for example, uploading a non-supported model file type.
 - *Solution:* Verify that all parameters are correctly specified and use the appropriate file formats before retrying the request.
- **401 Unauthorized:** Authentication failed due to an invalid or missing API key. This error occurs when the API key is not included in the request header or if the key provided is incorrect.
 - *Solution:* Ensure that the correct API key is included in the `X-API-Key` header. Double-check the key for any typos or missing characters.
- **404 Not Found:** The requested resource was not found. This typically happens when an invalid or non-existent `model_id` is used in the request.
 - *Solution:* Verify that the `model_id` is correct and that the model exists in your account. Use the "List All Models" endpoint to retrieve valid model IDs.
- **429 Too Many Requests:** The rate limit for API requests has been exceeded. This error prevents overuse and protects the platform's resources.
 - *Solution:* Implement exponential backoff in your code to retry the request after a short delay. Ensure your code adheres to the rate limit of 100 requests per minute.
- **500 Internal Server Error:** An unexpected server error occurred. This is usually due to an issue on the server side, such as a temporary outage or an unhandled exception.
 - *Solution:* Retry the request after a few minutes. If the issue persists, contact support with the details of the request.

Best Practices

Follow these best practices to ensure a smooth and secure experience when using the Unified AI Platform API.

- **Handle errors:** Add error-handling logic to your code to manage issues like invalid input or server errors. Use the error codes and messages from the API to provide helpful feedback to users.
- **Secure your API key:** Store your API key in a safe location, such as an environment variable or a secure vault. Avoid exposing the key in client-side code or public repositories. For example, set an environment variable for the API key as follows:

```
export API_KEY='YOUR_API_KEY'
```

- **Respect rate limits:** The API has rate limits to prevent overuse. The API allows 100 requests per minute. If you receive a `429 Too Many Requests` error, implement exponential backoff in your code.
- **Use consistent naming:** When deploying models, use clear and consistent names to make them easier to find and manage.

Additional Resources

For more information and support, explore the following resources:

- **Unified AI Platform documentation:** Visit the [Unified AI Platform Documentation](#) for detailed guides on using other endpoints and platform features.
- **Support:** If you encounter any issues, visit the [Unified AI Platform Support](#) page or contact support at support@unified-ai-platform.com.