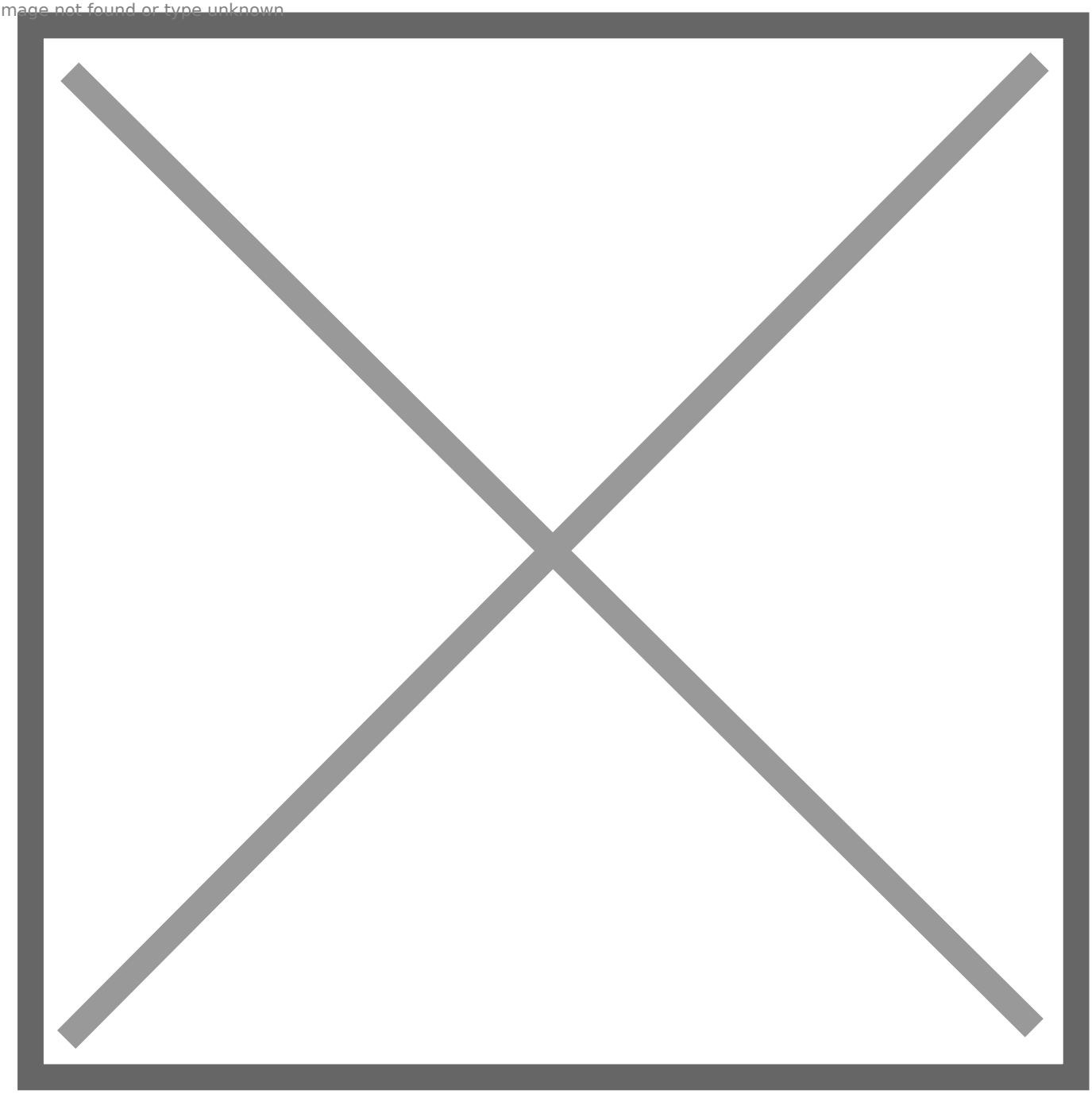
Obtaining advisor id

FeelGPT AnalyzeFile API endpoint requires an advisor-id as part of the request. This document explains how to get obtain an advisor-id

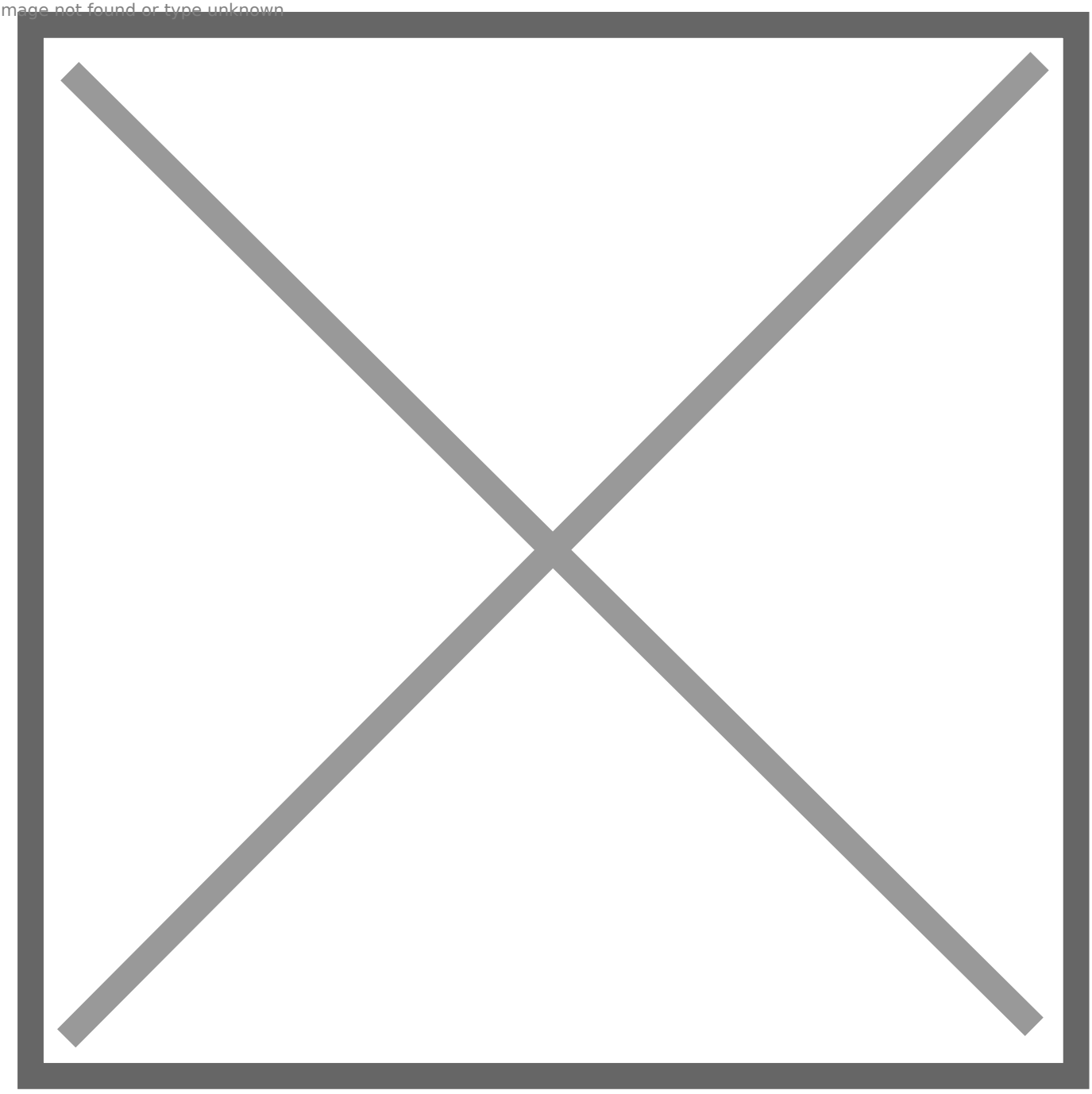
1. On FeelGPT, click "Let's Start" button on your prefered advisor

Image not found or type unknown



2. The advisor-id it located at the top-right of the screen

Image not found or type unknown



3. Copy the advisor-id to the clipboard by clicking the "copy" icon.

FeelGPT Get Advisors List

advisors is an HTTP GET enpoint to retrieve a list of all available advisors.

A call to **advisors** endpoint requires basic authentication. Please refer to Analyze Now Authentication

Here is a sample Javascript code to fetch the advisors list

```
const fetchAdvisors = async () => {

  const baseUrl = 'https://api.emlo.cloud/apigateway-service/feelgpt/v1';
  const apiKey = '[YOUR API KEY]';
  const apiPassword = '[YOUR API PASSWORD]';
  const url = `${baseUrl}/advisors`;

  try {
    const response = await fetch(url, {
      method: 'GET', // HTTP method
      headers: {
        'Authorization': 'Basic ' + btoa(`${apiKey}:${apiPassword}`), // Authorization header
        'Content-Type': 'application/json' // Content type header
      }
    });

    if (!response.ok) {
      throw new Error('Network response was not ok ' + response.statusText);
    }

    const data = await response.json(); // Parsing the JSON response
    console.log(data); // Logging the response data
  } catch (error) {
    console.error('There has been a problem with your fetch operation:', error);
  }
};
```

analyze is an HTTP POST enpoint to start an asynchronus process to analyze an audio file.

The analysis process status reported though a webhook calls from FeelGPT analyzer.

A call to analyze endpoint requires basic authentication. Please refer to Analyze Now Authentication

It is recommended to encrypt the callback payload data by passing an "encryptionKey" string value on the request. Read more

Learn how to obtain the advisor-id for your prefered advisor Here

Parameters

Param Name	Is Mandatory	Comments
audioLanguge	yes	The spoken language in the audio file
file	yes	a file to analyze

analysisLanguage	yes	The language FeelGPT will use for the analysis report
statusCallbackUrl	yes	A webhook URL for status calls from FeelGPT analysis engine
sendPdf	no	I "true", send the analysis results in PDF format on analysis completion. The file on the callback is based64 encoded
encryptionKey	no	Encryption key to encode the "payload" field on webhook callback

See NodeJS sampke code

Install required libraries

```
npm install axios form-data
```

```
const axios = require('axios');
const FormData = require('form-data');
const fs = require('fs');
const path = require('path');

const analyze = async () => {
  const baseUrl = 'https://api.emlo.cloud/apigateway-service/feelgpt/v1';
  const apiKey = '[API KEY]';
  const apiPassword = '[API PASSWORD]';
  const advisorId = '[ADVISOR ID]'; // the advisorId of the advisor you want to use for analysis
  const url = `${baseUrl}/advisors/${advisorId}/analyze`;
  const webHookCallbackURL = "[WEBHOOK URL]"; // a callback URL to get asynchronous status calls
  const encryptionKey = "qDtvUr7gU8*njigft9"; // an encryption key to encrypt the "payload" section in the status callbacks

  const form = new FormData();
  form.append('audioLanguage', 'en');
  form.append('file', fs.createReadStream(path.join(__dirname, 'path_to_your_audio_file.wav')));
  form.append('analysisLanguage', 'en');
  form.append('statusCallbackUrl', webHookCallbackURL);
  form.append('sendPdf', 'true');

  const authString = Buffer.from(`${apiKey}:${apiPassword}`).toString('base64');

  try {
    const response = await axios.post(url, form, {
      headers: {
        ...form.getHeaders(),
        'Authorization': `Basic ${authString}`
      }
    });

    console.log(response.data);
  } catch (error) {
    console.error('Error during the analyze request:', error.response ? error.response.data : error.message);
  }
};

analyze();
```

Explanation

1. **Importing Libraries:**
    1. `axios` for making HTTP requests.
    2. `form-data` for handling form data, especially for file uploads
    3. `fs` for file system operations
    4. `path` for handling file paths.
2. **Creating the Form Data:**
    1. A new instance of `FormData` is created.
    2. Required fields are appended to the form, including the audio file using `fs.createReadStream()` to read the file from the disk.
3. **Making the Request:**
    1. The `axios.post()` method sends a POST request to the specified URL.
    2. Basic authentication is used via the `auth` option.
    3. `form.getHeaders()` is used to set the appropriate headers for the form data.
4. **Handling the Response:**
    1. The response is logged to the console.
    2. Any errors are caught and logged, with detailed error information if available
    3. Replace `'path_to_your_audio_file.wav'` with the actual path to your audio file. This code will send a POST request to the "analyze" endpoint with the required form data and handle the response accordingly.

Response Structure

Upon request reception, FeelGPT validate the request parameters. For a valid request FeelGPT will return a "reportId" identifier to be used when recieving asynchronous status updates.

For invalid parameter the response will return an error code and message which indicates the invalid param.

Sample response for a valid request


```
{
    "reportId": "573b1949-6f6a-4176-bfa5-25c50b116922"
}
```


Sample response for a request with an invalid parameter


```
{
    "errorCode": "invalid_parameter",
    "error": "body.statusCallbackUrl must be a valid URL"
}
```


Once a valid request accepped on FeelGPT, it starts sending status update to the URL provided on "statusCallbackUrl" parameter.

Sample status callback data


```
{
  "requestId": "4df2a898-2617-4324-8dd5-45de15e5bb51",
  "application": "feelgpt",
  "eventDate": "2024-06-18T06:00:49.154Z",
▼ ▼ "payload": {
    "reportId": "573b1949-6f6a-4176-bfa5-25c50b116922",
    "status": "queued"
  },
  "encrypted": false
}
```


application: always "feelgpt".

eventDate: Time of the event in GMT timezone

payload: contain the actual event data

payload/reportId: The reportId that was provided on the response for the analysis request

payload/status: The current analysis status

encrypted: true of "encryptionKey" parameter sent on the analysis request

Avaialble Status

queued - The analysis request was successfully accepted, and queud for analysis

transcripting - The audio is now on transcription

analyzing - FeelGPT analyze the audio for emotions

completed - The report is ready. the "result" data contains the analysis data

pdfReady - If a PDF report was requested on the request, the payload for this status contains a PDF file in Base64 encoding

---

Revision #1
Created 12 March 2025 13:36:34 by mauricio olarte
Updated 12 March 2025 14:05:40 by mauricio olarte