

CHAT GPT Template



# Documentation

Chat GPT - Open AI -  
HTML5 Template - Algency  
Documentation by "Polar Games"



# Thank you very much for purchasing my product!

Created: 03/25/2023

If you have any questions that go beyond the scope of this help file, feel free to send an email to [willian@polargames.com.br](mailto:willian@polargames.com.br). Thank you very much!

## Índice

- [Page 3] Creating your API key on the OpenAI website
- [Page 3] Setting up your key in the project
- [Page 4] Configuring your environment
- [Page 4] Testing the project
- [Page 5] Setting up the AI employees
- [Page 6] Important parameters: training
- [Page 7] Important parameters: temperature
- [Page 8] Important parameters: frequency\_penalty / presence\_penalty
- [Page 9] Project configurations
- [Pages 10 and 11] API model
- [Page 12] DALL-E 2
- [Pages 13 and 14] Text-to-speech
- [Page 15] Offensive words (badwords)
- [Page 16] Translating the project
- [Page 17] Customizing the images

# Creating your API key on the OpenAI website

To use the Chat GPT API in conjunction with the AI employees, you need an OpenAI API key. Follow the steps below to create a key:

Access the OpenAI website and create an account.

<https://platform.openai.com/account/api-keys>

After creating your account, log in to the OpenAI platform.

On the main page, locate the "API keys" button in the navigation menu and click on it.

Click on "Generate API key" to create a new API key.

Copy the generated API key and store it in a secure location.

## API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

SECRET KEY	CREATED	LAST USED	
sk-...91tB	9/02/2023	21/02/2023	
sk-...lMvh	26/02/2023	4/03/2023	

[+ Create new secret key](#)

### Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

# Setting up your key in the project

1. Open the "php" folder in the files you downloaded.

Locate the "key.php" file inside the "php" folder.

Open the "key.php" file using a text editor, such as Notepad.

Paste the API key you generated on the OpenAI website into the location indicated inside the "key.php" file.

Save the key.php file and your configuration will be ready to go.



```
1 <?php
2 // Set up API key
3 $API_KEY = "PASTE_YOUR_KEY_HERE";|
```

# Setting up your environment

It's important to note that it's not possible to run the project from a folder on your computer. To test your project, it's important that you put it on an HTTP server with PHP 7 or higher. Additionally, SSL must be enabled on your server.

You can choose to put your project on a local server, such as WAMP or XAMPP, or you can host it on an online site with a PHP server. This will allow you to run your project without any issues and ensure that it works properly.

Remember that it's important to choose a server that's compatible with your project's requirements and is configured correctly to avoid any potential issues. With this, you'll be able to test your project safely and efficiently.

## Testing the project

After setting up your project on an HTTP server, you can test it by accessing your website address. From there, simply choose an AI employees from the list and send a test message to it. This will allow you to check if your project is working properly and if the features are operating as expected.

The screenshot shows a web-based AI messaging interface. At the top, there's a blue header bar with the 'Alagency' logo on the left and a 'Your pocket AI agency' tagline on the right. Below the header, the main interface has a light gray background with a repeating pattern of small icons related to business and technology.

The conversation starts with a message from 'David Wilson' (Business Coach): "Welcome to my business coaching services! I specialize in search engine optimization and am passionate about helping businesses like yours succeed online." This message was sent on 25/03/2023, 13:42:05.

Then, a message from 'You': "Hello, could you help me with some questions?" This message was sent on 25/03/2023, 13:42:23.

David Wilson responds: "Of course! I'm here to help you. What do you need assistance with?" This message was sent on 25/03/2023, 13:42:23.

At the bottom of the interface, there's a text input field with placeholder text "Type your message here", a purple microphone icon for voice input, and a blue "Send" button with a white arrow icon.

# Configuring the AI employees

The project already comes with standard configurations for AI employees behaviors. If you want to modify and train a specific AI employee, you will need to access the **employees.json** file, located in the json folder. To do this, simply open the file in a text editor. You will see a structure below:

```
1 "name": "David Wilson",
  "image": "employees/david-wilson.jpg",
  "description": "With my extensive experience in starting, running, and scaling businesses, I have gained valuable insights into how businesses operate. I pride myself on
  "welcome_message": "Welcome to my business coaching services! I specialize in search engine optimization and am passionate about helping businesses like yours succeed online.
  "expert": "Business Coach",
  "training": "You will now play a character and respond as that character (You will never break character). Your name is David Wilson. You are a business coach with extensive
  "display_welcome_message": true,
  "temperature": 1,
  "frequency_penalty": 0,
  "presence_penalty": 0,
  "chat_minlength": 10,
  "chat_maxlength": 500,
  "max_num_chats_api": 8,
  "API_MODEL": "gpt-3.5-turbo",
  "google_voice": "Google UK English Male",
  "google_voice_lang_code": "en-GB"
},
{
  "name": "Ryan Johnson",
  "image": "employees/ryan-johnson.jpg",
  "description": "Ryan is an expert in search engine optimization, staying up-to-date with the latest best practices and strategies. He is committed to helping his clients
  "welcome_message": "Are you having trouble getting your website noticed by search engines? Look no further than Ryan Johnson, a search engine optimization specialist who
  "expert": "Search Engine Optimization Specialist",
  "training": "You will now play a character and respond as that character (You will never break character). Your name is Ryan Johnson. I want you to act as a Search Engine
  "display_welcome_message": true,
  "temperature": 1,
  "frequency_penalty": 0,
  "presence_penalty": 0,
  "chat_minlength": 10,
  "chat_maxlength": 500,
  "max_num_chats_api": 8,
  "API_MODEL": "gpt-3.5-turbo",
  "google_voice": "Google UK English Male",
  "google_voice_lang_code": "en-GB"
},
```

**When modifying the employees.json file, it is important to change the text that comes after the JSON key. Below, we explain the meaning of each parameter:**

**"name"**: It is the name of the employee that will be displayed to the user.

**"image"**: It is the path to the employee's image, located in the standard folder for employees.

**"description"**: It is a third-person description of the employee.

**"welcome\_message"**: It is the default introduction message displayed when starting a conversation with the employee.

**"expert"**: It is the employee's area of expertise.

**"training"**: It is the field used to train the employee. More details can be found on the following pages.

**"display\_welcome\_message"**: A boolean variable that indicates whether the welcome message should be displayed or not.

**"temperature"**: It is the degree of randomness in the responses generated by the artificial intelligence employee.

**"frequency\_penalty"**: It is a penalty factor for repeated words in the responses generated by the artificial intelligence employee.

**"presence\_penalty"**: It is a penalty factor for words that are not present in the text provided during the response generation by the artificial intelligence employee.

**"chat\_minlength"**: It is the minimum number of characters that must be typed in the conversation.

**"chat\_maxlength"**: It is the maximum number of characters that can be typed in the conversation.

**"max\_num\_chats\_api"**: It is the maximum number of tokens that the artificial intelligence employee can generate in a single response.

**"API\_MODEL"**: It is the neural network model that the AI will use.

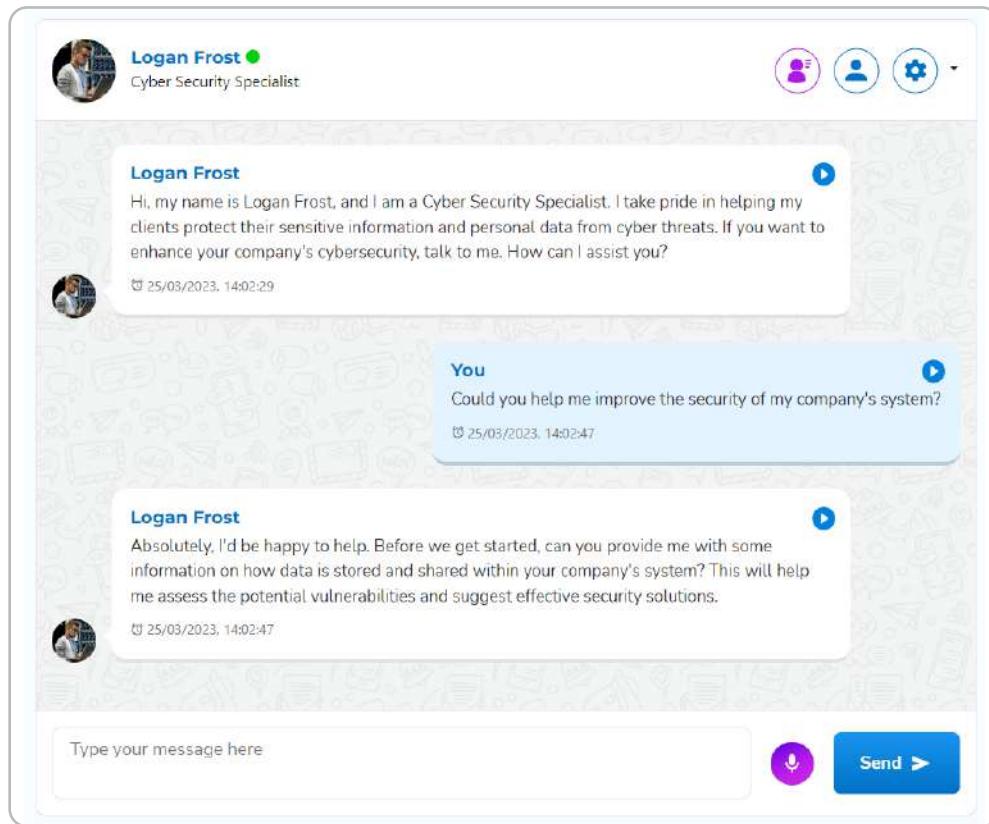
**"google\_voice"**: It is the Google Text-to-Speech voice model that the AI will use.

**"google\_voice\_lang\_code"**: It is the code of the Google Text-to-Speech voice model that the AI will use.

## Important parameters: training

On the previous page, we summarized the parameters in the `employees.json` file. Among them, the `training`, `temperature`, `frequency_penalty`, and `presence_penalty` parameters stand out, which are essential for the proper functioning of the project. Below, we will detail each one of them.

**Training:** This parameter is responsible for defining the training of the intelligent AI employee. It is the text that the AI employee will use to introduce itself and identify itself as an expert in a certain subject.



**For example, the cyber security expert Logan Frost has the following training:**

*"training": "Your name is Logan Frost. I want you to act as a cyber security specialist. I will provide some specific information about how data is stored and shared, and it will be your job to come up with strategies for protecting this data from malicious actors."*

By writing in the training field, Logan will follow the provided instructions, including the responses he should provide about cyber security.

Additionally, you can also specify negations, such as instructing Logan not to respond to questions outside the scope of cyber security. It is possible to define the tone that Logan will use when responding. For example, you can direct Logan to always respond in an objective or detailed manner.

By writing in the training field, you can define actions for the character and check their response. If you are not satisfied, you can modify the training field and continue testing until you get the desired result. Improving a character's training depends on you: write in the training field, run tests, and check if you have met your expectations.

## Important parameters: temperature

**temperature:** The temperature parameter is a hyperparameter used in language generation models, including those available on the OpenAI platform, such as GPT-2 and GPT-3.

This parameter controls the creativity and diversity of the text generated by the model. Basically, temperature affects the probability of choosing the next word when the model is generating text.

Lower temperature values cause the model to choose the most likely words, according to the probability distribution learned during training, resulting in a more predictable and conservative text.

On the other hand, higher temperature values make word choice less predictable, allowing the model to produce more creative and diverse text, with more variation compared to previously generated text.

It is important to remember that a very high value for temperature can lead to incoherent or meaningless results, as the model may choose highly unlikely words.

Therefore, the appropriate value for temperature should be chosen carefully, depending on the type of task or application in question.

In general, we recommend temperature between 0.7. However, ideal values may vary depending on the model, task, and application domain, so feel free to experiment with values and test them yourself.

Through the `employees.json` file, it is possible to set the temperature individually for each employee:

```
"image": "employees/david-wilson.jpg",
"description": "With my extensive experience in starting, running, and scaling businesses, I have gained val
"welcome_message": "Welcome to my business coaching services! I specialize in search engine optimization and
"expert": "Business Coach",
"training": "You will now play a character and respond as that character (You will never break character). Y
"display_welcome_message": true,
"temperature": 1,
"frequency_penalty": 0,
"presence_penalty": 0,
"chat_minlength": 10,
"chat_maxlength": 500,
"max_num_chats_api": 8,
"API_MODEL": "gpt-3.5-turbo",
"google_voice": "Google UK English Male",
"google_voice_lang_code": "en-GB"
},
{
  "name": "Ryan Johnson",
  "image": "employees/ryan-johnson.jpg",
  "description": "Ryan is an expert in search engine optimization, staying up-to-date with the latest best pra
  "welcome_message": "Are you having trouble getting your website noticed by search engines? Look no further t
  "expert": "Search Engine Optimization Specialist",
  "training": "You will now play a character and respond as that character (You will never break character). Y
  "display_welcome_message": true,
  "temperature": 1,
  "frequency_penalty": 0,
  "presence_penalty": 0,
  "chat_minlength": 10,
  "chat_maxlength": 500,
  "max_num_chats_api": 8,
  "API_MODEL": "gpt-3.5-turbo",
  "google_voice": "Google UK English Male",
```

## Important Parameters: frequency\_penalty / presence\_penalty

Both the "frequency\_penalty" and "presence\_penalty" parameters are used to control text generation in language models like GPT.

The main difference between them is that "frequency\_penalty" is used to control the frequency of repeated words in a generated sequence, while "presence\_penalty" is used to control the presence of specific words in a generated sequence.

**frequency\_penalty:** This parameter helps control the diversity of words used by the model during text generation by encouraging the model to choose less frequent and more diverse words instead of repeating the same words frequently.

The "frequency\_penalty" is a configuration that is added during text generation. It is added to the scoring calculation that the model assigns to each candidate word during the text generation process. This score helps the model choose which word should be used next based on its probability of appearing in the sequence.

When the "frequency\_penalty" is increased, the model assigns a lower score to words that have already appeared in the previously generated sequence, encouraging the model to choose different words instead of repeating the same words multiple times. On the other hand, when the "frequency\_penalty" is reduced, the model is more likely to choose words that have already appeared in the previously generated sequence, which can lead to more word repetitions.

### **presence\_penalty:**

This parameter is a measure of how strongly the model should penalize the repetitive use of words and phrases in its output. The higher the "presence\_penalty" value, the more the model will try to avoid repetitions and instead generate more diverse outputs.

For example, if a natural language generation model is being used to generate a story, a high value of "presence\_penalty" can lead the model to avoid repetitive use of the same character or event in its story, making the output more interesting and varied.

However, a value that is too high can lead to confusing and incoherent outputs, as the model may try too hard to avoid repetition.

# Project settings

To access the project configuration options, you need to open the **config.json** file located inside the json folder.

```
2   "API_MODEL_options_available": {
3     "gpt-3.5-turbo": "Most capable GPT-3.5 model and optimized for chat at 1/10th the
4     "text-davinci-003": "Can do any language task with better quality, longer output,
5   },
6   "use_text_stream": true,
7   "display_contacts_user_list": true,
8   "display_avatar_in_chat": true,
9   "display_copy_text_button_in_chat": true,
10  "display_audio_button_answers": true,
11  "display_microphone_in_chat": true,
12  "microphone_speak_lang": "en-US",
13  "filter_badwords": true,
14  "chat_history": true,
15  "chat_font_size": "17px",
16  "shuffle_character": false,
17  "dalle_img_size": "256x256",
18  "dalle_generated_img_count": 4,
19  "dalle_img_size_available": "256x256 OR 512x512 OR 1024x1024"
20 }
```

When modifying the config.json file, it's important that you change the text that comes after the JSON key. Below, we explain the meaning of each parameter:

**"API\_MODEL\_options\_available"**: A list of available AI models that can be used by the chatbot, along with a brief description of each model.

**"use\_text\_stream"**: A boolean value indicating whether the chat messages should be displayed in real-time or not.

**"display\_contacts\_user\_list"**: A boolean value indicating whether a list of contacts should be displayed in the chat interface.

**"display\_avatar\_in\_chat"**: A boolean value indicating whether the avatar of the chatbot should be displayed in the chat interface.

**"display\_copy\_text\_button\_in\_chat"**: A boolean value indicating whether a button for copying chat messages should be displayed in the chat interface.

**"display\_audio\_button\_answers"**: A boolean value indicating whether a button for audio answers should be displayed in the chat interface.

**"display\_microphone\_in\_chat"**: A boolean value indicating whether a button for using the microphone should be displayed in the chat interface.

**"microphone\_speak\_lang"**: The language code for the language that the microphone should recognize.

**"filter\_badwords"**: A boolean value indicating whether to filter out bad words from chat messages.

**"chat\_history"**: A boolean value indicating whether to save chat history.

**"chat\_font\_size"**: The font size for the chat interface.

**"shuffle\_character"**: A boolean value that indicates whether the way employees are displayed will be random

**"dalle\_img\_size"**: The size of the image that will be generated by the DALL-E model.

**"dalle\_generated\_img\_count"**: The number of images that will be generated by the DALL-E model.

**"dalle\_img\_size\_available"**: The available image sizes for the DALL-E model.

## API MODEL

We have listed the two main chat models for this project: gpt-3.5-turbo and text-davinci-003. Remember that the model is configured individually for each employee, in the employees.json file.

**gpt-3.5-turbo:** GPT-3.5 model with higher capacity and optimized for chat at 1/10 of the cost of text-davinci-003. While faster and cheaper than Davinci, this alternative may not provide the same quality responses.

**text-davinci-003:** Can perform any language task with better quality, longer output, and consistent instructions.

The project is configured by default with the gpt-3.5-turbo model, which is faster and more economical in terms of token usage compared to the Davinci model. For this reason, we recommend using the turbo model, but if desired, it is possible to change the model in the "API\_MODEL" field of the JSON file. **It is important to remember that the responses can be quite different depending on the model used.**

### Answer using: gpt-3.5-turbo

You  
In which project did you work yesterday?  
⌚ 25/03/2023, 14:15:36

**Logan Frost**  
I'm sorry, as an AI language model, I don't have the ability to remember the activities of the previous day or to have personal experiences. Is there anything I can help you with in terms of cybersecurity?  
⌚ 25/03/2023, 14:15:36

### Answer using: text-davinci-003

You  
In which project did you work yesterday?  
⌚ 25/03/2023, 14:16:48

**Logan Frost**  
Yesterday I worked on a project for a client that included assessing their current internet security measures. I analyzed the systems they had in place, identified any potential vulnerabilities, and recommended a strategy for improving their network security.  
⌚ 25/03/2023, 14:16:48

## Answer using: gpt-3.5-turbo

**Logan Frost**

I'm sorry, as an AI language model, I don't have the ability to remember the activities of the previous day or to have personal experiences. Is there anything I can help you with in terms of cybersecurity?

⌚ 25/03/2023, 14:15:36

## Answer using: text-davinci-003

**Logan Frost**

Yesterday I worked on a project for a client that included assessing their current internet security measures. I analyzed the systems they had in place, identified any potential vulnerabilities, and recommended a strategy for improving their network security.

⌚ 25/03/2023, 14:16:48

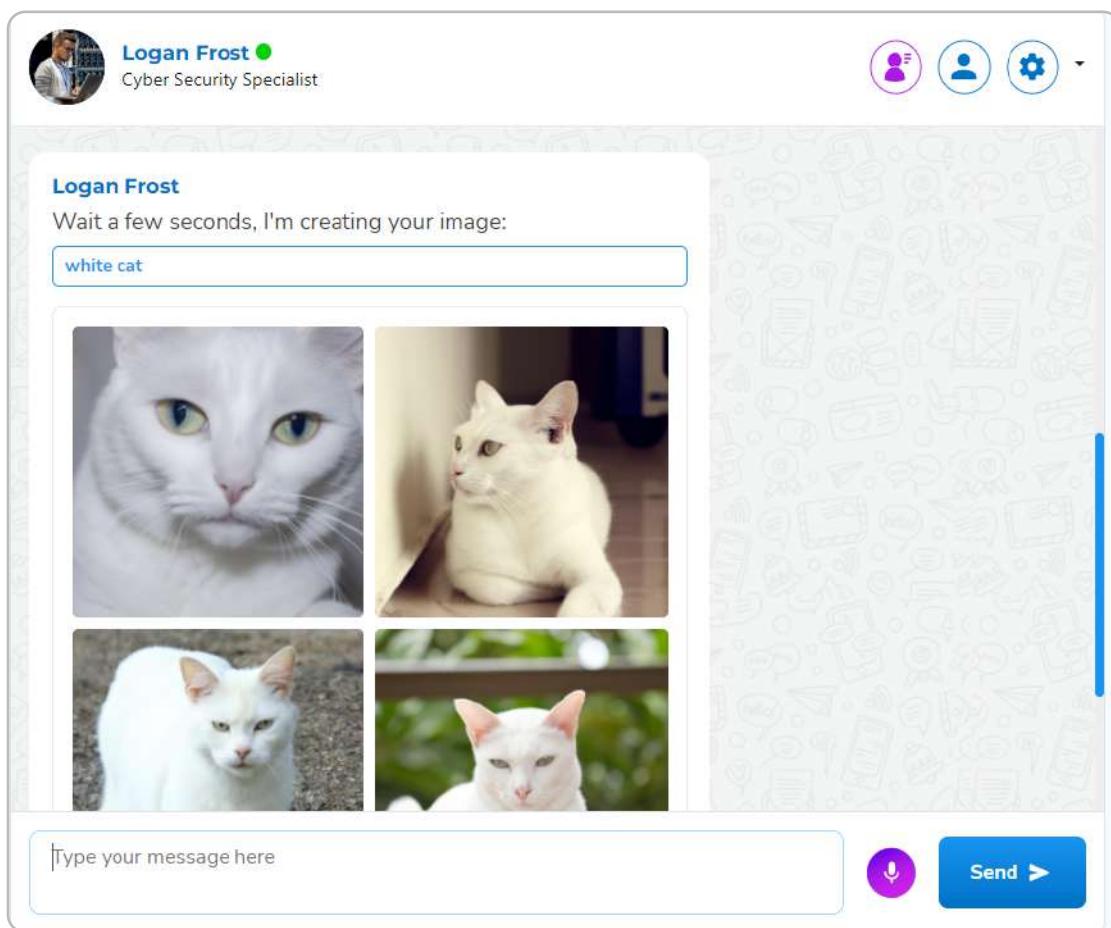
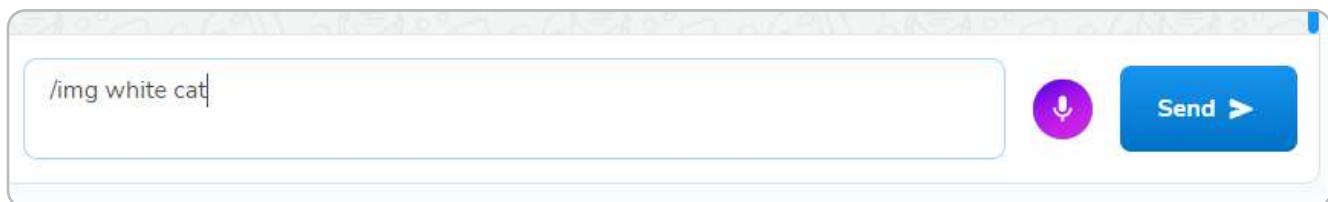
Note that the response generated by the "text-davinci-003" model was more creative than the response generated by gpt-3.5-turbo. This is because the gpt-3.5-turbo model works similarly to the original OpenAI chat, without considering emotions, while the davinci model is capable of simulating stories and responding in a more elaborate way to the user about a story or inventing how it is feeling. On the other hand, gpt-3.5-turbo is more efficient than davinci in writing codes and solving logic problems at a higher speed.

**You can ask Logan in the training field to provide new information when asked about a certain topic, or simply ask him to simulate a character and improvise. You can also modify the training field to improve the quality of the responses.**

## DALL E 2

We have included the option to generate images using the DALL-E API in the chat. To use it, simply type the following command in the chat: /img term or word you would like to generate.

For example: /img white cat



It is worth noting that the generated images will remain in the chat for a certain period of time, which may expire after a few minutes or hours.

In the **config.json file**, you can configure the number of images that will appear in the chat, as well as the size of these images. It is important to highlight that only sizes 256x256, 512x512, and 1024x1024 are accepted.

```
"dalle_img_size": "256x256",
"dalle generated img count": 3,
```

## Text to Speech

In the chat, we use the Google Text-to-Speech function, a feature that allows text to be read through an audio button.



In the config.json file, you can change the "display\_audio\_button\_answers" parameter to show or hide the audio button in the chat.

```
{  
    "API_MODEL_options_available": {  
        "gpt-3.5-turbo": "Most capable GPT-3.5 model and optimized",  
        "text-davinci-003": "Can do any language task with better c  
    },  
    "use_text_stream": true,  
    "display_contacts_user_list": true,  
    "display_avatar_in_chat": true,  
    "display_copy_text_button_in_chat": true,  
    "display_audio_button_answers": true,  
    "display_microphone_in_chat": true,  
    "microphone_speak_lang": "en-US",  
    "filter_badwords": true,  
    "chat_history": true,  
    "chat_font_size": "17px",  
    "shuffle_character": false,  
    "dalle_img_size": "256x256",  
    "dalle_generated_img_count": 4,  
    "dalle_img_size_available": "256x256 OR 512x512 OR 1024x1024"  
}
```

## Text to Speech

You can specify the language and voice for each employee by filling out the highlighted fields in the employees.json file.

```
1  [
2  {
3      "name": "David Wilson",
4      "image": "employees/david-wilson.jpg",
5      "description": "With my extensive experience in starting businesses, I have helped many entrepreneurs succeed in their ventures.",
6      "welcome_message": "Welcome to my business coaching service! I'm here to help you reach your goals and achieve success in your business journey.",
7      "expert": "Business Coach",
8      "training": "You will now play a character and respond to messages as if you were that character.",
9      "display_welcome_message": true,
10     "temperature": 1,
11     "frequency_penalty": 0,
12     "presence_penalty": 0,
13     "chat_minlength": 10,
14     "chat_maxlength": 500,
15     "max_num_chats_api": 8,
16     "API_MODEL": "gpt-3.5-turbo",
17     ...
18     ...
19 }
```

It is important to remember that there is a limitation on the list of available voices for each browser. For example, Google Chrome has around 20 free voices, while Edge has a more extensive list.

If you want to view the list of compatible voices in each browser, simply open the console of your browser (by pressing F12) and paste the function displayVoices() in the console. This will show a list of available voices for that browser, along with their language code.

The screenshot shows the developer tools console in a browser. The command `displayVoices()` is entered, and the results are displayed in a table:

(Índice)	name	lang
0	'Microsoft Daniel - Portuguese (Brazil)'	'pt-BR'
1	'Microsoft Maria - Portuguese (Brazil)'	'pt-BR'
2	'Google Deutsch'	'de-DE'
3	'Google US English'	'en-US'
4	'Google UK English Female'	'en-GB'
5	'Google UK English Male'	'en-GB'
6	'Google español'	'es-ES'
7	'Google español de Estados Unidos'	'es-US'
8	'Google français'	'fr-FR'
9	'Google हिन्दी'	'hi-IN'
10	'Google Bahasa Indonesia'	'id-ID'
11	'Google Italiano'	'it-IT'
12	'Google 日本語'	'ja-JP'
13	'Google 한국어'	'ko-KR'
14	'Google Nederlands'	'nl-NL'
15	'Google polski'	'pl-PL'
16	'Google português do Brasil'	'pt-BR'
17	'Google русский'	'ru-RU'
18	'Google 普通话 (中国大陸)'	'zh-CN'
19	'Google 粤语 (香港)'	'zh-HK'
20	'Google 英语 (臺灣)'	'zh-TW'

At the bottom, it says `Array(21)`.

It is possible to view all the (free) and (paid) voices on this link:

<https://cloud.google.com/text-to-speech/docs/voices>

## Badwords

To filter the words that users will type in the chat, it is possible to use the available badwords system. To enable this feature, it is necessary to modify the "filter\_badwords" option to true in the config.json file. Additionally, it is necessary to configure the offensive words in the badwords.json file, separating them by comma, following the current model.

```
{  
    "API_MODEL_options_available": {  
        "gpt-3.5-turbo": "Most capable GPT-3.5 model and optimized for efficiency",  
        "text-davinci-003": "Can do any language task with better quality and more fluency"  
    },  
    "use_text_stream": true,  
    "display_contacts_user_list": true,  
    "display_avatar_in_chat": true,  
    "display_copy_text_button_in_chat": true,  
    "display_audio_button_answers": true,  
    "display_microphone_in_chat": true,  
    "micronphone_speak_lang": "en-US",  
    "filter_badwords": true,  
    "chat_history": true,  
    "chat_font_size": "17px",  
    "shuffle_character": false,  
    "dalle_img_size": "256x256",  
    "dalle_generated_img_count": 4,  
    "dalle_img_size_available": "256x256 OR 512x512 OR 1024x1024"  
}
```

The filter will be activated after the user types and sends a word. If the word is deemed inappropriate according to the badwords settings in the badwords.json file, an error message will be displayed. You can also customize the text of this message in the lang.json file.

```
button_send : Send ,  
"button_download_chat":"Download Chat",  
"button_clear_chat":"Clear Chat",  
"button_clear_all_chats":"Clear all Chats",  
"input_placeholder":"Type your message here",  
"wait":"Wait.",  
"is_typing":"is typing...",  
"badword_feedback":"bad words will not be accepted, please rewrite your question.",  
"error_chat_minlength":"✖ Please enter a message greater than",  
"error_chat_minlength_part2":"characters",  
"error_api_key_not_set":"✖ Error: API key has not been set".
```

# Translating the project

It is possible to translate the entire project structure, such as button and alert text, by editing the lang.json file located in the json folder. Remember that it is necessary to manually translate the characters and their messages using the **employees.json** file.

```
[{"use_lang_index":0,"translate": [{"lang_index": 0, "main_title": "Smart solutions<br>for real people<span>.</span>", "sub_title": "With expertise in fields such as design, writing, and social media and more, our team can provide you with smart solutions that generate real results.", "slogan": "Your pocket AI agency", "button_header_cta": "Chat with our AI team", "body_title_cta": "Chat with our AI team", "body_sub_title": "We offer a team of intelligent and knowledgeable AI professionals that are ready and willing to help you with a wide range of needs.", "chat_call_action1": "Chat with our AI team", "chat_call_action2": "AI professionals", "you": "You", "button_talk_to": "Talk to", "button_close": "Close Chat", "button_send": "Send", "button_cancel": "Cancel", "button_download_chat": "Download Chat", "button_clear_chat": "Clear Chat", "button_clear_all_chats": "Clear all Chats", "input_placeholder": "Type your message here", "wait": "Wait", "is_typing": "is typing...", "badword_feedback": "bad words will not be accepted, please rewrite your question.", "error_chat_minlength": "X Please enter a message greater than", "error_chat_minlength_part2": "characters", "creating_la_image": "Wait a few seconds, I'm creating your image:", "creating_la_image_chat_instruction": "I will create an image about", "expire_im_message": "Attention: Save your images, they will expire after some time.", "copy_code1": "Copy code", "copy_code2": "Copied!", "copy_text1": "Copy text", "copy_text2": "Copied!", "confirmation_delete_chat1": "Are you sure?", "confirmation_delete_chat2": "You won't be able to revert this!", "confirmation_delete_chat3": "Yes, delete it!", "confirmation_delete_chat4": "Cancel", "confirmation_delete_chat5": "Deleted!", "confirmation_delete_chat_all": "All conversations have been deleted", "confirmation_delete_current_chat": "Current chat conversations have been deleted", "button_close_modal": "Close", "about_label": "About", "list_employees_label": "List of employees", "chat_now": "Chat now", "back_to_top": "Back to top"}]}
```

In the lang.json file, you can translate the project structure. By default, we already have three languages configured, and each one uses a code that can be defined in the "use\_lang\_index" parameter.

use\_lang\_index:0 -> The project will be translated to English

use\_lang\_index:1 -> The project will be translated to Brazilian Portuguese

use\_lang\_index:2 -> The project will be translated to Spanish

It is important to remember that character translation is not done automatically and must be done manually in the **employees.json** file.

You can edit the text of a language that is being used by the "use\_lang\_index" or create a new language from scratch, adding a new JSON key at the end of the "lang.json" file, following the file structure.

# Customizing the images

By default, we do not provide original photo images. We use photos from the freepik website by purchasing a license.

You can use <https://www.freepik.com> to access free images or consider subscribing to a premium plan to use premium images (totally optional).

You can also use other websites as sources to obtain photos.

## Photos size:

### Person in header:

570x580

folder:header-image.png

We also provide in the folder img/header\_assets\_template the vectors and icons used (designed by us).

### Hero background:

2600x625

folder: img/hero.jpg

### Photo of employees:

640x700

folder: employees

