GPT@JRC API Developers Manual

V0.1

Introduction

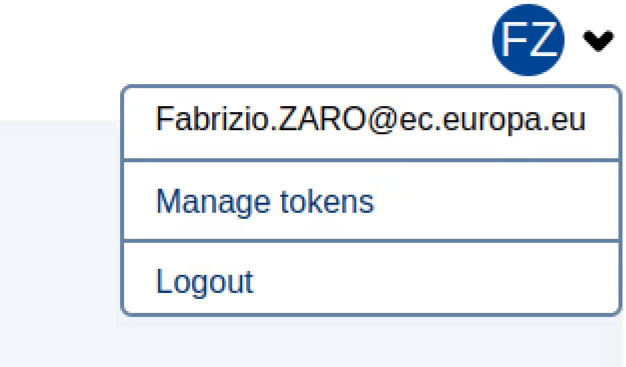
The GPT@JRC API provides programmatic access to all featured LLMs and it is compliant with OpenAI specifications.

Accessing the API requires a token that may be generated from the website.

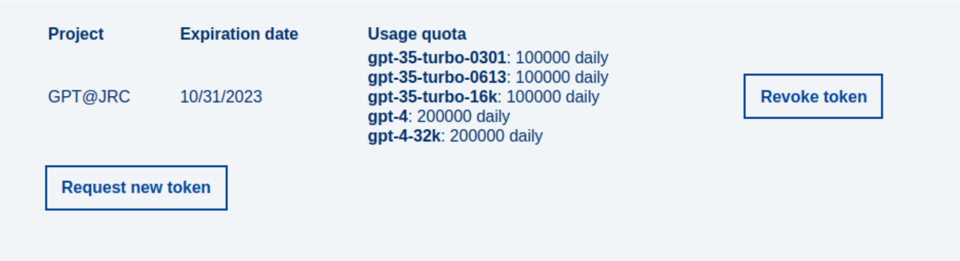
The API usage may not exceed a fixed daily quota that defaults to 200.000 tokens but can be adjusted, depending on specific project needs. Please note that the daily quota is set per project, not per individual user.

How to create a token

1. After logging into the JRC@GPT platform, click on your user icon

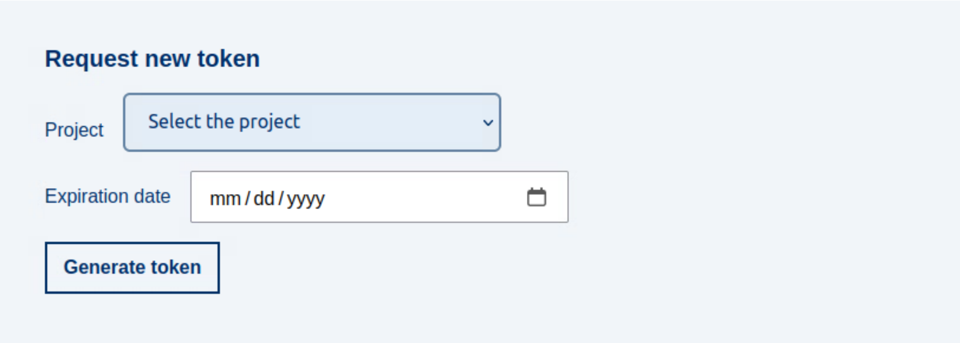


2. Then click "Manage tokens"



Project	Expiration date	Usage quota
GPT@JRC	10/31/2023	**gpt-35-turbo-0301**: 100000 daily **gpt-35-turbo-0613**: 100000 daily **gpt-35-turbo-16k**: 100000 daily **gpt-4**: 200000 daily **gpt-4-32k**: 200000 daily

3. Then click ”Request new token”

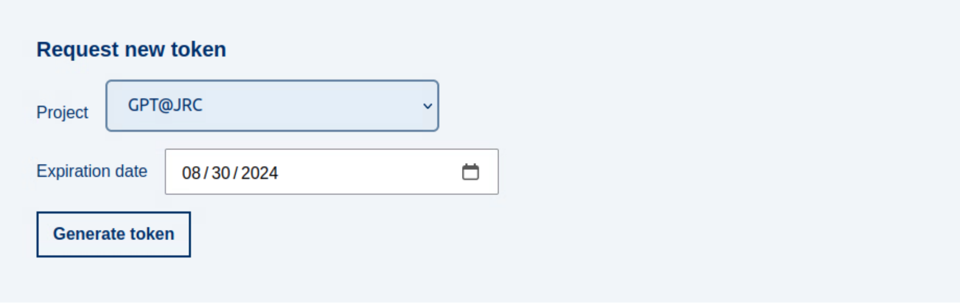


Request new token

Project: Select the project

Expiration date: mm / dd / yyyy

4. Fill the form, specifying the project for which the new token will be, and the expiration date (max 1 year), then click “Generate token”

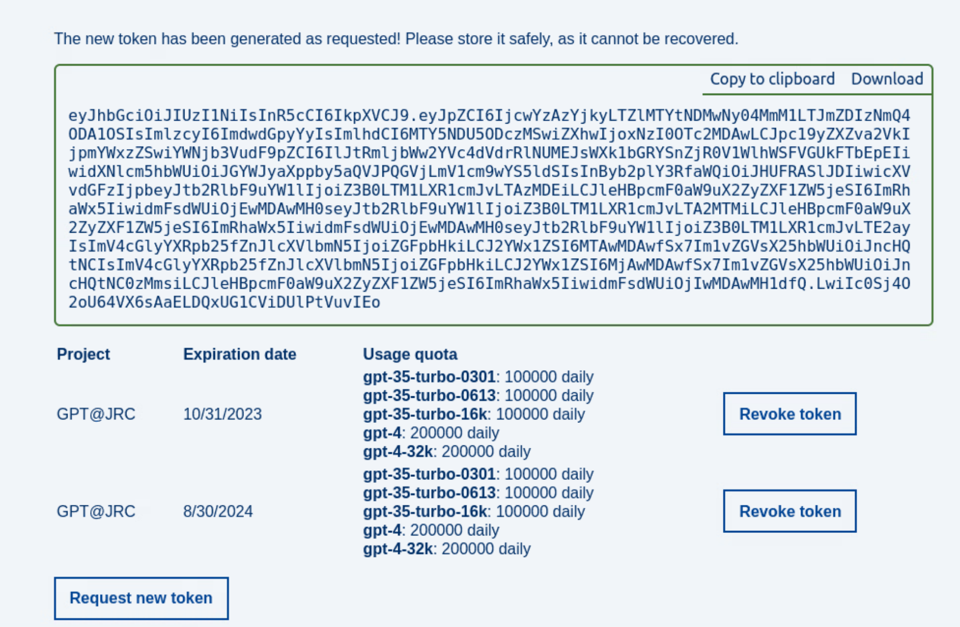


Request new token

Project: GPT@JRC

Expiration date: 08 / 30 / 2024

5. The new token is generated! Please remember to save the token, as it cannot be retrieved



The new token has been generated as requested! Please store it safely, as it cannot be recovered.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjcwYzAzYjkyLTZlMTYtNDMwNy04MmM1LTJmZDIzNmQ4
ODA1OSIsImlzcyI6ImdwdGpyYyIsImlhdCI6MTY5NDU5ODczMSwiZXhwIjoxNzI0OTc2MDAwLCJpc19yZXZva2VkI
jpmYWxzZSwiYWNjb3VudF9pZCI6IlJtRmljbWw2YVc4dVdrRlNUMEJsWXk1bGRYSnZjR0V1WlhWSFVGUkFTbEpEIi
widXNlcm5hbWUiOiJGYWJyaXppby5aQVJPQGVjLmV1cm9wYS5ldSIsInByb2plY3RfaWQiOiJHUFRASlJDIiwicXV
vdGFzIjpbeyJtb2RlbF9uYW1lIjoiZ3B0LTM1LXR1cmJvLTAzMDEiLCJleHBpcmF0aW9uX2ZyZXF1ZW5jeSI6ImRh
aWx5IiwidmFsdWUiOjEwMDAwMH0seyJtb2RlbF9uYW1lIjoiZ3B0LTM1LXR1cmJvLTA2MTMiLCJleHBpcmF0aW9uX
2ZyZXF1ZW5jeSI6ImRhaWx5IiwidmFsdWUiOjEwMDAwMH0seyJtb2RlbF9uYW1lIjoiZ3B0LTM1LXR1cmJvLTE2ay
IsImV4cGlyYXRpb25fZnJlcXVlbmN5IjoiZGFpbHkiLCJ2YWx1ZSI6MTAwMDAwfSx7Im1vZGVsX25hbWUiOiJncHQ
tNCIsImV4cGlyYXRpb25fZnJlcXVlbmN5IjoiZGFpbHkiLCJ2YWx1ZSI6MjAwMDAwfSx7Im1vZGVsX25hbWUiOiJn
cHQtNC0zMmsiLCJleHBpcmF0aW9uX2ZyZXF1ZW5jeSI6ImRhaWx5IiwidmFsdWUiOjIwMDAwMH1dfQ.LwiIc0Sj4O
2oU64VX6sAaELDQxUG1CViDUlPtVuvIEo
```

Project	Expiration date	Usage quota	
GPT@JRC	10/31/2023	**gpt-35-turbo-0301**: 100000 daily **gpt-35-turbo-0613**: 100000 daily **gpt-35-turbo-16k**: 100000 daily **gpt-4**: 200000 daily **gpt-4-32k**: 200000 daily	Revoke token
GPT@JRC	8/30/2024	**gpt-35-turbo-0301**: 100000 daily **gpt-35-turbo-0613**: 100000 daily **gpt-35-turbo-16k**: 100000 daily **gpt-4**: 200000 daily **gpt-4-32k**: 200000 daily	Revoke token

You will have the possibility to copy it to the clipboard, to save it how you prefer, or to download it as a *.jwt file

CURL example

```
curl -s \
  -H "Authorization: Bearer [TOKEN]" \
  -H "Content-Type: application/json" \
  -d '{"model": "gpt-35-turbo-0613", "max_tokens": 1024, "stream": false,
"messages": [{"role": "user", "content": "Hello!"}]}'
  https://api-gpt.jrc.ec.europa.eu/v1/chat/completions
```

Python Example

```
import openai

TOKEN = "your token"

openai.api_base = "https://api-gpt.jrc.ec.europa.eu/v1"
openai.api_key = TOKEN

chat_completion = openai.ChatCompletion.create(
    headers = {"Authorization": f"Bearer {TOKEN}"},
    model = "gpt-35-turbo-0613",
    stream = False,
    messages = [{"role": "user", "content": "Hello!"}]
)

# print the chat completion
print(chat_completion.choices[0].message.content)
```

NodeJS Example

```
import { Configuration, OpenAIApi } from "openai";

const prompt = "YOUR PROMPT HERE";
const token = "YOUR TOKEN HERE";
const model = "gpt-35-turbo-0613"; // Or any other available model

const configuration = new Configuration({
  apiKey: token,
  basePath: "https://api-gpt.jrc.ec.europa.eu/v1"
});
const openai = new OpenAIApi(configuration);

openai.createChatCompletion({
    model: model,
    stream: false,
    messages: [
        {
            role: 'user',
            content: prompt
        }
    ]
})
.then(value => {
    // Print the chat completion answers
    console.log(value.data.choices.map(c => c.message?.content ||
"").join("\n"));
})
```