## 7.3. Листинги

|  |
| --- |
| Тестовый код с <https://github.com/gpt4free/gpt4free.github.io/blob/main/docs/async_client.md> |
| Вывод сразу полностью готового ответа  import asyncio  from g4f.client import AsyncClient  async def main():  client = AsyncClient()    response = await client.chat.completions.create(  model="gpt-4o-mini",  messages=[  {  "role": "user",  "content": "Say this is a test"  }  ],  web\_search = False  )    print(response.choices[0].message.content)  asyncio.run(main()) |
| Вывод потоковый  import asyncio  from g4f.client import AsyncClient  async def main():  client = AsyncClient()  stream = client.chat.completions.stream(  model="gpt-4",  messages=[  {  "role": "user",  "content": "Say this is a test"  }  ],  web\_search = False  )  async for chunk in stream:  if chunk.choices and chunk.choices[0].delta.content:  print(chunk.choices[0].delta.content, end="")  asyncio.run(main()) |
| Используем системную роль для указания ИИ в какой роли он выступает  import asyncio  from g4f.client import AsyncClient  async def main():  client = AsyncClient()  stream = client.chat.completions.stream(  model="gpt-4",  messages=[  {"role": "system", "content": "You are a helpful assistant."},  {"role": "user", "content": "What day is it today?"}  ],  web\_search = False  )  async for chunk in stream:  if chunk.choices and chunk.choices[0].delta.content:  print(chunk.choices[0].delta.content, end="")  asyncio.run(main()) |

|  |
| --- |
| Регистрация через аккаунт гугла на <https://opencagedata.com/> и получение ключа |
| <https://opencagedata.com/demo>    Открываем JSON и ищем в нем координаты города    Копируем JSON и просматриваем его на специальном сайте <https://jsononline.net/ru/json-beautifier> Переключаемся на Tree в правом окне и ищем параметр geometry там будут широта и долгота    <https://opencagedata.com/tutorials/geocode-in-python> Пример проекта |
| Копируем код  from opencage.geocoder import OpenCageGeocode  key = '97c595bec990457d975c12c16a4ec4a7'  geocoder = OpenCageGeocode(key)  query = u'Bosutska ulica 10, Trnje, Zagreb, Croatia'  # no need to URI encode query, module does that for you  results = geocoder.geocode(query)  print(u'%f;%f;%s;%s' % (results[0]['geometry']['lat'],  results[0]['geometry']['lng'],  results[0]['components']['country\_code'],  results[0]['annotations']['timezone']['name']))  # 45.797095;15.982453;hr;Europe/Belgrade |
| Устанавливаем библиотеку pip install opencage |
| Переделаем код  from opencage.geocoder import OpenCageGeocode  def get\_coordinates(city, key):  geocoder = OpenCageGeocode(key)  query = city  results = geocoder.geocode(query)  if results:  return results[0]['geometry']['lat'], results[0]['geometry']['lng']  else:  return "Город не найден"  # Пример использования  key = '97c595bec990457d975c12c16a4ec4a7'  city = 'Москва'  coordinates = get\_coordinates(city, key)  print(f"Координаты города {city}: {coordinates}") |
| Добавляем обработку исключений  from opencage.geocoder import OpenCageGeocode  def get\_coordinates(city, key):  """ Получает координаты города, используя библиотеку OpenCage. """  try:  geocoder = OpenCageGeocode(key)  results = geocoder.geocode(city)  if results:  # Возвращает первый результат  return results[0]['geometry']['lat'], results[0]['geometry']['lng']  else:  return "Город не найден"  except Exception as e:  return f"Общая ошибка: {e}"  # Пример использования  key = '97c595bec990457d975c12c16a4ec4a7'  city = 'Москва'  coordinates = get\_coordinates(city, key)  print(f"Координаты города {city}: {coordinates}") |
| Округляем координаты  from opencage.geocoder import OpenCageGeocode  def get\_coordinates(city, key):  """ Получает координаты города, используя библиотеку OpenCage. """  try:  geocoder = OpenCageGeocode(key)  results = geocoder.geocode(city)  if results:  lat = round(results[0]['geometry']['lat'], 2)  lng = round(results[0]['geometry']['lng'], 2)  return lat, lng  else:  return "Город не найден"  except Exception as e:  return f"Общая ошибка: {e}"  # Пример использования  key = '97c595bec990457d975c12c16a4ec4a7'  city = 'Москва'  coordinates = get\_coordinates(city, key)  print(f"Координаты города {city}: {coordinates}") |
| Добавляем при выводе слова Широта и Долгота  def get\_coordinates(city, key):  """ Получает координаты города, используя библиотеку OpenCage. """  try:  geocoder = OpenCageGeocode(key)  results = geocoder.geocode(city)  if results:  lat = round(results[0]['geometry']['lat'], 2)  lng = round(results[0]['geometry']['lng'], 2)  return f"Широта: {lat}, Долгота: {lng}"  else:  return "Город не найден"  except Exception as e:  return f"Общая ошибка: {e}" |
| Добавляем в проект интерфейс  from tkinter import \*  from opencage.geocoder import OpenCageGeocode  def get\_coordinates(city, key):  try:  geocoder = OpenCageGeocode(key)  results = geocoder.geocode(city)  if results:  lat = round(results[0]['geometry']['lat'], 2)  lng = round(results[0]['geometry']['lng'], 2)  return f"Широта: {lat}, Долгота: {lng}"  else:  return "Город не найден"  except Exception as e:  return f"Ошибка: {e}"  def show\_coordinates(event=None):  city = entry.get()  coordinates = get\_coordinates(city, key)  label.config(text=coordinates)  # Интерфейс  window = Tk()  window.title("Поиск координат города")  key = '97c595bec990457d975c12c16a4ec4a7'  # Элементы интерфейса  entry = Entry()  entry.pack()  entry.bind("<Return>", show\_coordinates) # Привязка события нажатия Enter  button = Button(text="Поиск", command=show\_coordinates)  button.pack()  label = Label(text="Введите город и нажмите Поиск")  label.pack()  # Запуск приложения  window.mainloop() |
| <https://jsononline.net/ru/json-beautifier> Переключаемся на Tree в правом окне и ищем параметр 'country' там будет название страны и рядом название региона в параметре 'state'  from tkinter import \*  from opencage.geocoder import OpenCageGeocode  def get\_coordinates(city, key):  try:  geocoder = OpenCageGeocode(key)  #results = geocoder.geocode(city)  results = geocoder.geocode(city, language='ru')  if results:  lat = round(results[0]['geometry']['lat'], 2)  lng = round(results[0]['geometry']['lng'], 2)  country = results[0]['components']['country']  region = results[0]['components']['state']  return f"Широта: {lat}, Долгота: {lng}\nСтрана: {country}\nРегион: {region}"  else:  return "Город не найден"  except Exception as e:  return f"Ошибка: {e}"  def show\_coordinates(event=None):  city = entry.get()  coordinates = get\_coordinates(city, key)  label.config(text=coordinates)  # Интерфейс  window = Tk()  window.title("Поиск координат города")  key = '97c595bec990457d975c12c16a4ec4a7'  # Элементы интерфейса  entry = Entry()  entry.pack()  entry.bind("<Return>", show\_coordinates)  button = Button(text="Поиск", command=show\_coordinates)  button.pack()  label = Label(text="Введите город и нажмите Поиск")  label.pack()  # Запуск приложения  window.mainloop() |
| Добавляем проверку на наличие параметра state  def get\_coordinates(city, key):  try:  geocoder = OpenCageGeocode(key)  results = geocoder.geocode(city, language='ru')  if results:  lat = round(results[0]['geometry']['lat'], 2)  lon = round(results[0]['geometry']['lng'], 2)  country = results[0]['components']['country']  # Проверка наличия ключа 'state'  if 'state' in results[0]['components']:  region = results[0]['components']['state']  return f"Широта: {lat}, Долгота: {lon}\n Страна: {country}. Регион: {region}"  else:  return f"Широта: {lat}, Долгота: {lon}\n Страна: {country}"  else:  return "Город не найден"  except Exception as e:  return f"Возникла ошибка: {e}" |
| Открываем карту в браузере  from tkinter import \*  from opencage.geocoder import OpenCageGeocode  import webbrowser  def get\_coordinates(city, key):  try:  geocoder = OpenCageGeocode(key)  results = geocoder.geocode(city, language='ru')  if results:  lat = round(results[0]['geometry']['lat'], 2)  lng = round(results[0]['geometry']['lng'], 2)  country = results[0]['components'].get('country', 'Страна не определена')  region = results[0]['components'].get('state', 'Регион не определен')  # Получаем URL для OpenStreetMap  osm\_url = f"https://www.openstreetmap.org/?mlat={lat}&mlon={lng}"  return {  "coordinates": f"Широта: {lat}, Долгота: {lng}\nСтрана: {country}\nРегион: {region}",  "map\_url": osm\_url  }  else:  return {"coordinates": "Город не найден", "map\_url": None}  except Exception as e:  return {"coordinates": f"Ошибка: {e}", "map\_url": None}  def show\_coordinates(event=None):  city = entry.get()  result = get\_coordinates(city, key)  label.config(text=result["coordinates"])  # Сохраняем URL в глобальной переменной для доступа из другой функции  global map\_url  map\_url = result["map\_url"]  def show\_map():  if map\_url:  webbrowser.open(map\_url)  # Интерфейс  window = Tk()  window.title("Поиск координат города")  key = '97c595bec990457d975c12c16a4ec4a7'  map\_url = None  # Элементы интерфейса  entry = Entry()  entry.pack()  entry.bind("<Return>", show\_coordinates)  button = Button(text="Поиск", command=show\_coordinates)  button.pack()  label = Label(text="Введите город и нажмите Поиск")  label.pack()  map\_button = Button(text="Показать карту", command=show\_map)  map\_button.pack()  # Запуск приложения  window.mainloop() |
| Добавляем окно с погодой  from tkinter import \*  from opencage.geocoder import OpenCageGeocode  import webbrowser  import requests  # Функция для получения погоды  def get\_weather(lat, lon):  try:  response = requests.get(f"https://api.open-meteo.com/v1/forecast?latitude={lat}&longitude={lon}&current\_weather=true")  weather\_data = response.json()  current\_weather = weather\_data['current\_weather']  return current\_weather  except Exception as e:  return f"Ошибка при получении погоды: {e}"  # Функция для отображения погоды  def show\_weather():  current\_weather = get\_weather(lat, lon) # lat и lon должны быть определены ранее  weather\_window = Toplevel(window)  weather\_window.title("Текущая погода")  if isinstance(current\_weather, str):  label = Label(weather\_window, text=current\_weather)  else:  temperature = current\_weather['temperature']  windspeed = current\_weather['windspeed']  weather\_text = f"Температура: {temperature}°C\nСкорость ветра: {windspeed} км/ч"  label = Label(weather\_window, text=weather\_text)  label.pack()  def get\_coordinates(city, key):  global lat, lon  try:  geocoder = OpenCageGeocode(key)  results = geocoder.geocode(city, language='ru')  if results:  lat = round(results[0]['geometry']['lat'], 2)  lon = round(results[0]['geometry']['lng'], 2)  country = results[0]['components'].get('country', 'Страна не определена')  region = results[0]['components'].get('state', 'Регион не определен')  # Получаем URL для OpenStreetMap  osm\_url = f"https://www.openstreetmap.org/?mlat={lat}&mlon={lon}"  return {  "coordinates": f"Широта: {lat}, Долгота: {lon}\nСтрана: {country}\nРегион: {region}",  "map\_url": osm\_url  }  else:  return {"coordinates": "Город не найден", "map\_url": None}  except Exception as e:  return {"coordinates": f"Ошибка: {e}", "map\_url": None}  def show\_coordinates(event=None):  city = entry.get()  result = get\_coordinates(city, key)  label.config(text=result["coordinates"])  # Сохраняем URL в глобальной переменной для доступа из другой функции  global map\_url  map\_url = result["map\_url"]  def show\_map():  if map\_url:  webbrowser.open(map\_url)  # Интерфейс  window = Tk()  window.title("Поиск координат города")  key = '97c595bec990457d975c12c16a4ec4a7'  map\_url = None  # Элементы интерфейса  entry = Entry()  entry.pack()  entry.bind("<Return>", show\_coordinates)  button = Button(text="Поиск", command=show\_coordinates)  button.pack()  label = Label(text="Введите город и нажмите Поиск")  label.pack()  map\_button = Button(text="Показать карту", command=show\_map)  map\_button.pack()  # Добавьте кнопку для показа погоды в основной интерфейс  weather\_button = Button(text="Показать погоду", command=show\_weather)  weather\_button.pack()  # Запуск приложения  window.mainloop() |