



## Актуальные зарплаты в IT

110k

100k

150k

120k

180k

**empenoso**

27 янв в 03:25

# Как автоматизировать свою переписку с девушкой в Telegram при помощи Python

Простой

5 мин

31K

Мессенджеры\*, API\*, Python\*, Node.JS\*

Кейс

Бывает так: девушка ждет внимания, а вы увлечены работой, хобби или ещё чем-то, и забываете о ней. Установка отложенных сообщений — идеальный компромисс. Эта статья не о создании чат-бота, который будет "любить" за вас. Речь о ненавязчивом напоминании для самого себя и создании точки входа для настоящего диалога.

## Зачем это всё?

Автоматизация позволяет отправлять заранее подготовленные, теплые и заботливые сообщения без излишнего пафоса в случайное время в заданном диапазоне. Представьте, "утро начинается не с кофе", а с вашего сообщения. Вот пример списка фраз для утреннего приветствия:

```
greetings = [  
    "Доброе утро! Как ты себя чувствуешь сегодня?",
```

РЕКЛАМА

**Хостинг aéza**

Твой проект этого достоин

```
"Доброе утро, как ты себя сегодня чувствуешь?",  
"Привет! Как ты, как сны?",  
"Как твои дела, готовность к этому дню?",  
"Доброе утро! Как настроение, что планируешь на день?",  
"Здравствуй! Как ты себя чувствуешь после сна?"
```

```
]
```

Даже пара теплых слов не оставит девушку равнодушной, и это перейдет уже в настоящий разговор.

Автоматизация не заменяет заботу, а помогает ее проявить, напоминая о важном человеке, даже если вы очень заняты.

## Puppeteer: почему это не лучший вариант

Изначально я пытался автоматизировать переписку через веб-версию Telegram, используя [Puppeteer](#). Идея казалась простой: скрипт заходит в Telegram, отправляет сообщения.

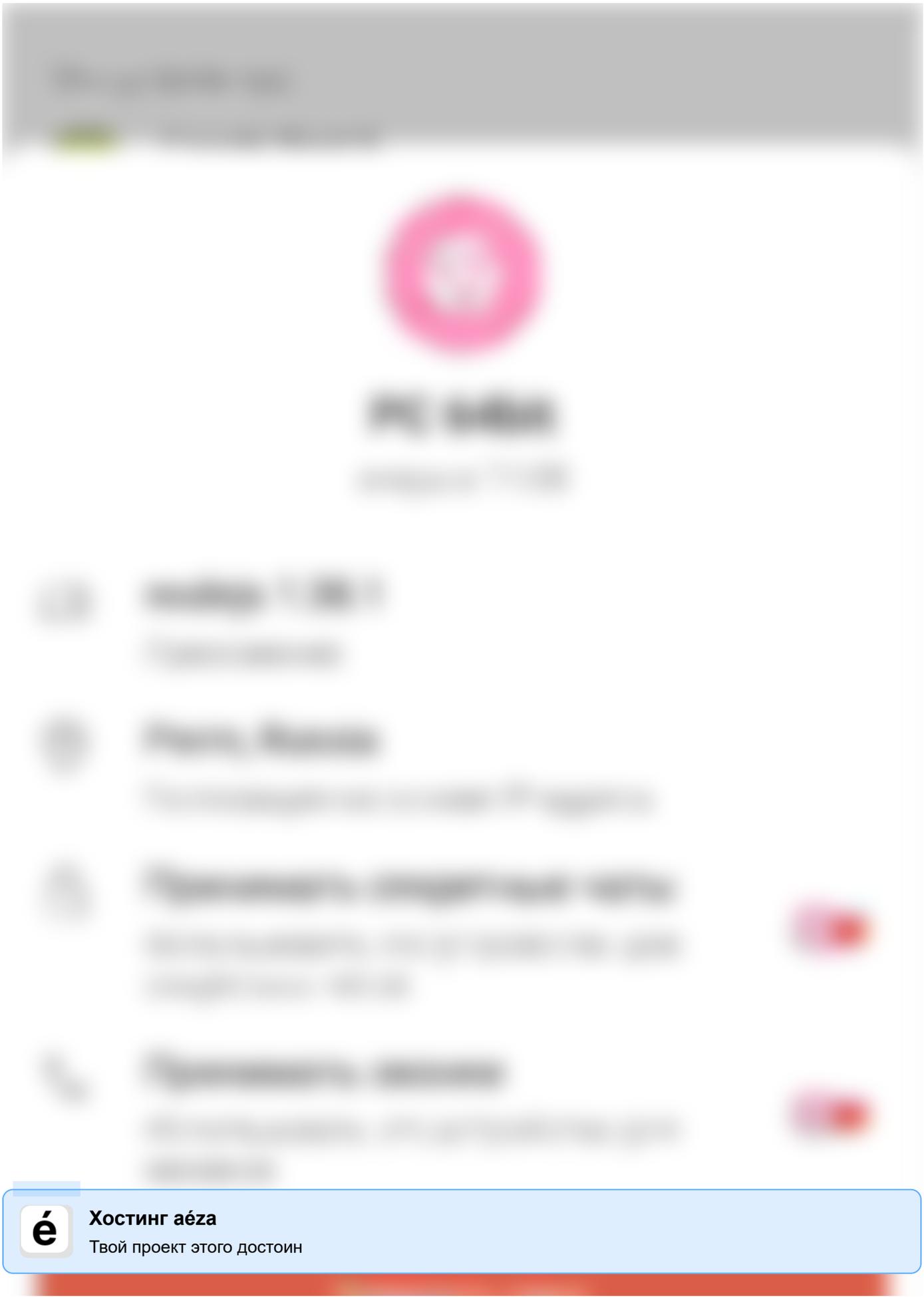
Был создан даже модуль авторизации через QR-код с двухфакторной аутентификацией. На практике же возникли сложности: изменения разметки веб-версии Telegram, сложности с реализацией отложенной отправки сообщений. Но для WhatsApp, например, есть подобная уже готовая [JavaScript-библиотека для автоматизации задач whatsapp-web.js](#).



**Хостинг aeza**

Твой проект этого достоин





**Хостинг aéza**

Твой проект этого достоин

## Преимущества Telethon:

- Простая установка ( `pip install telethon` ).
- Дополнительные возможности: чтение диалогов, создание чатов и каналов, скачивание медиафайлов.

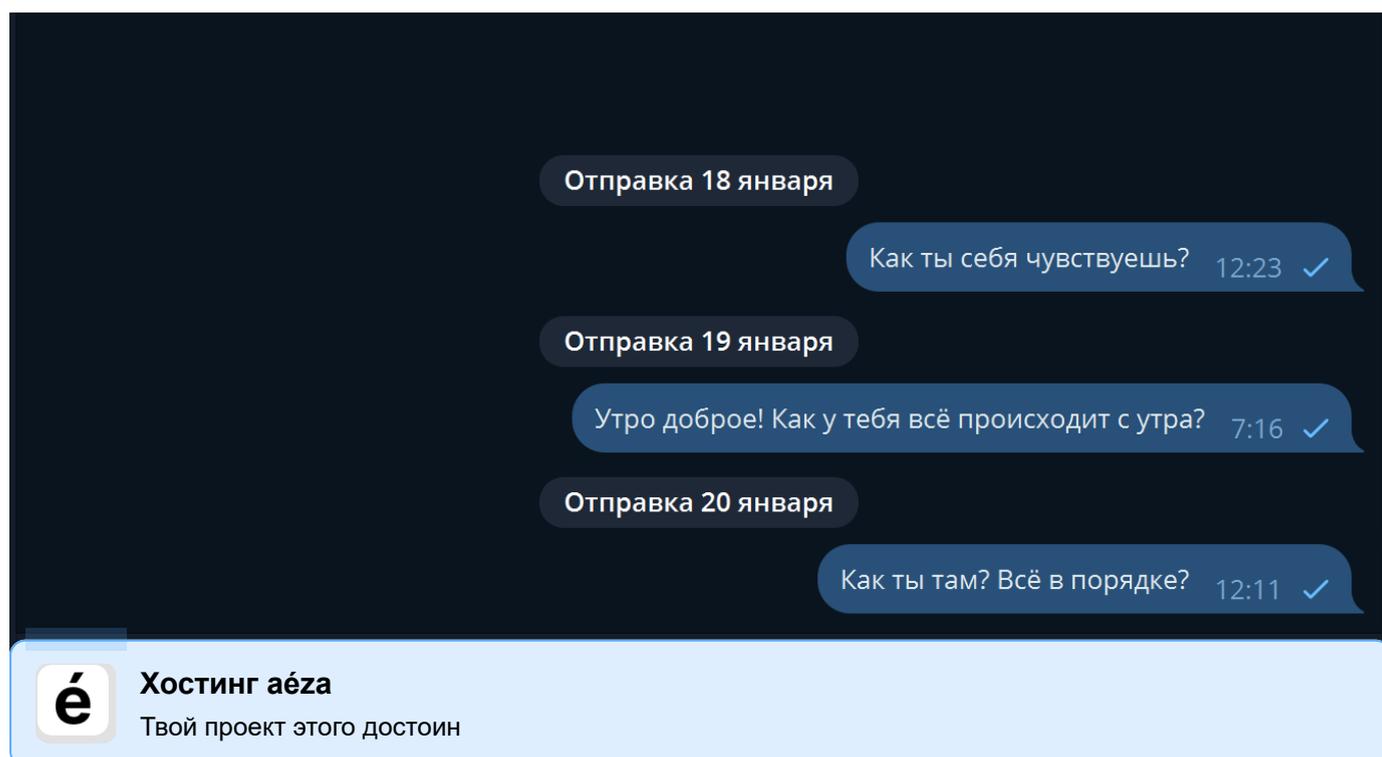
С Telethon можно создавать сценарии общения. Главное - не переусердствовать, чтобы не попасть под бан аккаунта в Telegram.

## Как избежать бана

Telegram не любит автоматизацию и тщательно следит за подозрительной активностью. Поэтому, если вы не хотите потерять доступ к своему аккаунту, нужно быть предельно аккуратным.

Чего точно не стоит делать? Массовая рассылка одинаковых сообщений. Но вам это не понадобится, ведь цель скрипта — общение с одним человеком, а не спам.

Как обезопасить себя? Используйте скрипт только для личного общения. Это уменьшает риск привлечь внимание системы.



Соблюдайте [лимиты API](#), описанные в статье на Хабре. Например, не отправляйте больше 10 медиасообщений за 5 минут - даже с лёгкой анимацией это вызовет ошибку.

## Скрипт: первая версия

```
# pip install telethon python-dotenv pytz

import asyncio
from telethon import TelegramClient
from datetime import datetime, timedelta, time
import random
import pytz
from dotenv import load_dotenv
import os

# Загрузка конфигурации
load_dotenv()
api_id = os.getenv("api_id")
api_hash = os.getenv("api_hash")
session_name = 'session_name'

# Таймзона и сообщения
ekb_timezone = pytz.timezone('Asia/Yekaterinburg')
messages = ["Доброе утро! Как твои дела?", "Привет! Надеюсь, у тебя всё отлично!"]

async def schedule_messages(client, user, messages, days):
    for day in range(days):
        send_time = ekb_timezone.localize(
            datetime.combine(datetime.now().date() + timedelta(days=day + 1),
                             time(random.randint(8, 10), random.randint(0, 59)))
        )
        message = random.choice(messages)
        await client.send_message(user, message, schedule=send_time.astimezone(pytz.UTC))
        print(f"Сообщение запланировано: {message} на {send_time}")

async def main():
    client = TelegramClient(session_name, api_id, api_hash)
    await client.start()
```



**Хостинг aéza**

Твой проект этого достоин

```
asyncio.run(main())
```

## Подготовка:

1. Установите необходимые библиотеки: `pip install telethon python-dotenv pytz` .
2. Создайте файл `.env` и добавьте туда `api_id` и `api_hash` , полученные на [my.telegram.org](https://my.telegram.org).
3. Замените `USERNAME` на имя пользователя или номер телефона.

## Полная версия скрипта

Полная версия скрипта автоматизации своей переписки с девушкой в Telegram представлена ниже. Скрипт подключается к аккаунту через API, загружает данные аутентификации из `.env` , и планирует отправку случайных сообщений (утренних и дневных) для заданных пользователей в определённое время:

```
# pip install telethon python-dotenv pytz

import sys
sys.stdout.reconfigure(encoding='utf-8')

import asyncio
from telethon import TelegramClient, utils
from datetime import datetime, timedelta, time
import random
from dotenv import load_dotenv
import os
import pytz

# Загрузка переменных окружения из .env
load_dotenv()
api_id = os.getenv("api_id") # Ваш API ID
api_hash = os.getenv("api_hash") # Ваш API Hash
```



**Хостинг aéza**

Твой проект этого достоин

```
greetings = [  
    "Доброе утро! Как ты себя чувствуешь сегодня?",  
    "Привет, как настроение? Готовность к новому дню?",  
    "Доброе утро! Что нового на сегодня?",  
    "Утро доброе! Как у тебя всё происходит с утра?",  
    "Доброе утро, как ты себя сегодня чувствуешь?",  
    "Привет! Как ты, как сны?",  
    "Как твои дела, готовность к этому дню?",  
    "Доброе утро! Как настроение, что планируешь на день?",  
    "Здравствуй! Как ты себя чувствуешь после сна?"  
]  
  
supportive_messages = [  
    "Как ты там? Всё в порядке?",  
    "Как проходит твой день?",  
    "Надеюсь, день не слишком утомительный для тебя!",  
    "Как настроение? Всё ли хорошо?",  
    "Не забывай делать перерывы, если устала!",  
    "Как ты? Чем занимаешься сейчас?",  
    "Если что-то нужно, не стесняйся мне писать.",  
    "Надеюсь, день приносит тебе только положительные моменты!",  
    "Как ты себя чувствуешь? Хочешь поговорить?"  
]  
  
# Таймзона Екатеринбурга  
ekb_timezone = pytz.timezone('Asia/Yekaterinburg')  
  
async def schedule_messages(client, user, messages, start_hour, end_hour, days, descrip  
    """Планирует сообщения для заданного пользователя на несколько дней."""  
    for day in range(days):  
        # Вычисляем дату и время в Екатеринбурге  
        now_ekb = datetime.now(ekb_timezone)  
        schedule_date_ekb = now_ekb.date() + timedelta(days=day + 1)  
        schedule_time_ekb = ekb_timezone.localize(datetime.combine(schedule_date_ekb, t  
  
        # Конвертируем время в UTC для Telethon  
        schedule_time_utc = schedule_time_ekb.astimezone(pytz.UTC)  
  
        message_text = random.choice(messages)  
  
        try:
```

**Хостинг aéza**

Твой проект этого достоин

```

        print(f"❌ Ошибка при отправке сообщения для {description}: {e}")

async def main():
    client = TelegramClient(session_name, api_id, api_hash)
    await client.start()

    print("🚀 Запуск программы...")
    try:
        tasks = []

        user_Name1 = await client.get_entity("empenoso") # имя пользователя
        tasks.append(schedule_messages(client, user_Name1, greetings, 7, 7, 3, "пользователь"))
        tasks.append(schedule_messages(client, user_Name1, supportive_messages, 12, 12, 3, "пользователь"))

        user_Name2 = await client.get_input_entity("+7912XXXXX") # непосредственно номер
        tasks.append(schedule_messages(client, user_Name2, greetings, 7, 7, 3, "пользователь"))

        await asyncio.gather(*tasks)
        print("🎉 Все сообщения успешно настроены.")
    except Exception as e:
        print(f"❌ Ошибка: {e}")
    finally:
        await client.disconnect()
        print("👋 Соединение с Telegram завершено.")

if __name__ == '__main__':
    asyncio.run(main())

```

Часовой пояс Екатеринбурга используется потому что я живу в Перми.

## Заключение

Данная автоматизация — инструмент для выражения эмоций. Скрипт не заменит живое общение, но поможет не забыть проявить внимание. Отложенные сообщения — это лишь начало. Не переставайте общаться лично. Автоматизируйте с душой, искренностью и чувством меры!



**Хостинг aéza**

Твой проект этого достоин

27 января 2025 г.

**Теги:** telethon, puppeteer, девушки

**Хэбы:** Мессенджеры, API, Python, Node.JS

## Редакторский дайджест



Присылаем лучшие статьи раз в месяц

Электронпочта



**174** **30.5**

Карма Рейтинг

**Михаил Шардин @empenoso**

Автоматизация / Данные / Финансы / Умные дома

Подписаться



[Сайт](#) [Сайт](#) [Github](#)



**Хостинг aéza**

Твой проект этого достоин

Комментарии 184

## Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ



**labyrinth**

20 часов назад

### Зачем Яндекс.Браузеру эти данные?



Простой



2 мин



16K

Кейс

+146

50

83



**nmgtech**

18 часов назад

### Я перешёл на Firefox и обратно возвращаться не намерен



Простой



5 мин



34K

Мнение

Перевод

+86

65

285



**BabayMazay**

19 часов назад



**Хостинг аéза**

Твой проект этого достоин

Тutorial

+46

15

5



vital\_pavlenko

14 часов назад

## Как AI захватывает Хабр, и почему это всех бесит

Простой 3 мин 3.5K

Мнение

+43

11

75



DimDimDimDimDim

20 часов назад

## Liberux NEXX: Linux-смартфон с 32 ГБ ОЗУ и аппаратными переключателями

4 мин 11K

+42

16

44



laurelido

20 часов назад

## Полезные паровозики. Часть 1: введение в железнодорожный моделизм

Простой 10 мин 2.4K

Обзор

+41

23

29



CyberPaul

16 часов назад

## Персоналки от Erpson — удивительные компьютеры из прошлого

Простой 6 мин 1.8K



Хостинг aEza

Твой проект этого достоин

 +33 9 0

wytear

18 часов назад

## Маскирование данных от А до Я

 Простой  14 мин  1.5K[Обзор](#) +24 32 12

PatientZero

21 час назад

## Оптимизация 3D-текстурирования для мобильных игр при помощи градиентов

 Простой  8 мин  1.1K[Кейс](#)[Перевод](#) +21 23 5

rakovskij\_stanislav

13 часов назад

## Вредоносные пакеты deepseeek и deepseekai были опубликованы в Python Package Index

 Простой  3 мин  1.8K +20 6 0

## Виммельбухи и визуальные головоломки: поиск предметов после вечеринки

[Турбо](#)[Показать еще](#)

Хостинг aéza

Твой проект этого достоин

## ИСТОРИИ



Где работать в IT в 2025: СИГМА



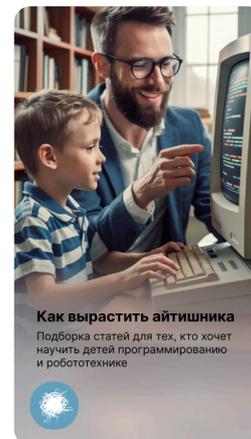
Отправьте эйчару валентинку



Что для вас важно в вашей работе?



Вышла Janus-Pro-7B



Как вырастить айтишника



Где ком

## КУРСЫ

**SE** Проектирование сложных API: OpenAPI + AsyncAPI

15 февраля 2025 · Systems Education

**SE** Интеграция систем. Разработка требований и основы проектирования

27 февраля 2025 · Systems Education

**SE** Systems Analyst Bootcamp: Проектировщик корпоративных информационных систем

20 марта 2025 · Systems Education

**SE** Проектирование и реализация очередей в брокерах RabbitMQ и Apache Kafka

22 марта 2025 · Systems Education

**K** Java для начинающих

По факту набора · Университет «Синергия»

[Больше курсов на Хабр Карьере](#)

## МИНУТОЧКУ ВНИМАНИЯ



**Хостинг aéza**

Твой проект этого достоин



**Рейтинг лучших IT-работодателей 2024**



**Зажигательные IT-ивенты, чтобы растопить вечную мерзлоту**



**Иди со мной, если хочешь на перекур: будущее ИИ на заводах**

## РАБОТА

Data Scientist

63 вакансии

Node.js разработчик

38 вакансий

Django разработчик

22 вакансии

Python разработчик

78 вакансий

JavaScript разработчик

127 вакансий

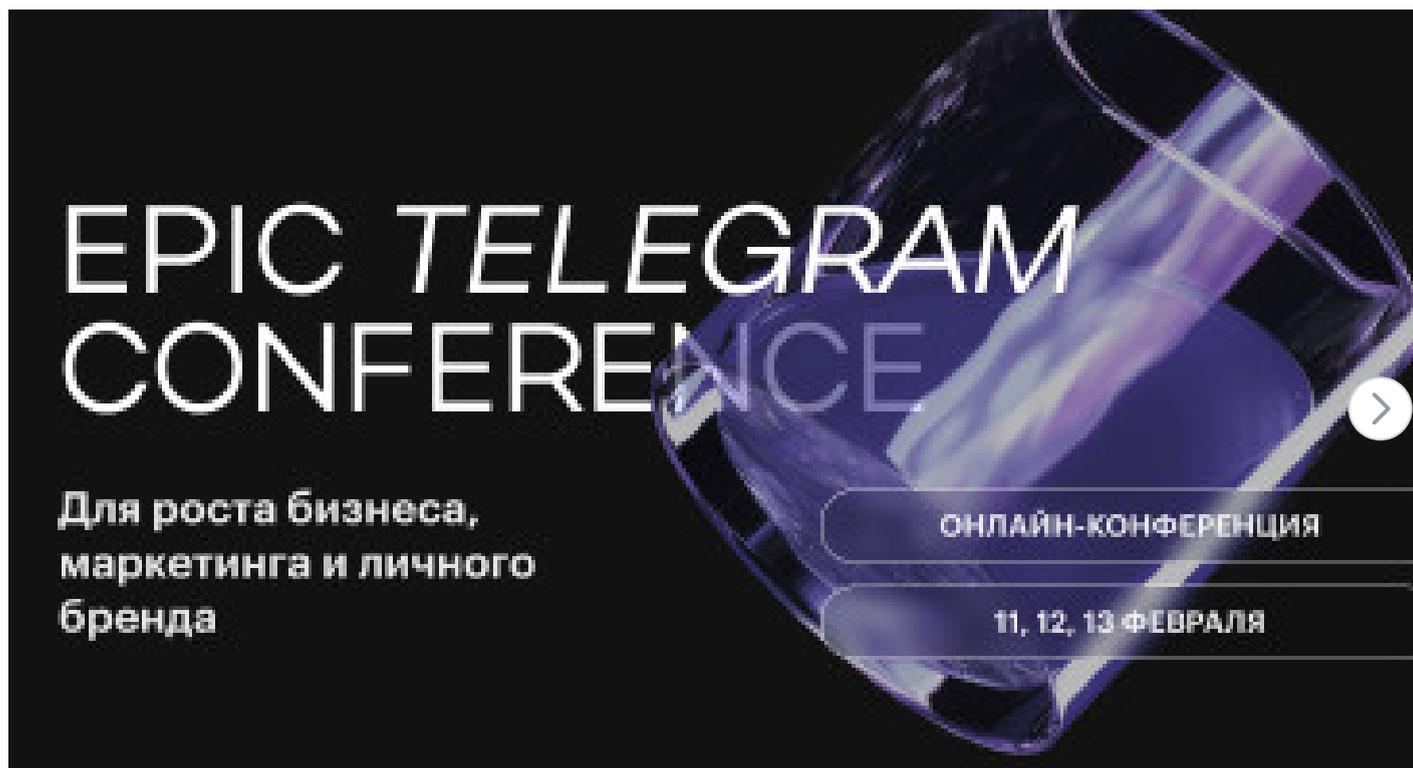
[Все вакансии](#)

## БЛИЖАЙШИЕ СОБЫТИЯ



**Хостинг aéza**

Твой проект этого достоин



11 – 13 февраля

## Epic Telegram Conference

Онлайн

Маркетинг

[Больше событий в календаре](#)

### Хабр



Настройка ссылки



**Хостинг aéza**

Твой проект этого достоин

Техническая поддержка

© 2006–2025, Habr



**Хостинг aéza**

Твой проект этого достоин