

Генерация описания профиля

Endpoint

POST <https://functions.yandexcloud.net/d4e3uer658357b9igkin>

Назначение

Облачная функция генерирует **3 варианта био** для дейтинга:

- `/ai/bio` : генерация на основе фактов
- `/ai/bio/rewrite` : доработка текущего био

Примеры запросов

1. Генерация нового био

Запрос:

```
curl -X POST https://functions.yandexcloud.net/d4e3uer658357b9igkin/ai/bio \
-H "Content-Type: application/json" \
-d '{
  "user_facts": {
    "профессия": "продуктовый дизайнер",
    "хобби": "пешие походы",
    "питомец": "кот Барсик"
  },
  "tone": "дружелюбный",
  "length": "среднее"
}'
```

Ответ:

```
{
  "bio_variants": [
    "Продуктовый дизайнер, любящий пешие походы. Барсик всегда рядом — мой компас и спутник.",
    "Создаю продукты днём, гуляю с Барсиком по тропам вечером. Присоединишься?",
    "💻🎨 Продуктовый дизайнер. 🥾 Люблю походы. 🐱 Барсик — босс моих приключений!"
  ]
}
```

2. Рерайт текущего био

Запрос:

```
curl -X POST https://functions.yandexcloud.net/d4e3uer658357b9igkin/ai/bio/rewrite \
-H "Content-Type: application/json" \
-d '{
  "current_bio": "Я дизайнер, люблю походы и Барсика.",
  "tone": "юмористический"
}'
```

Ответ:

```
{
  "rewritten_variants": [
    "Дизайнер с котом и кроссовками. Всё, что нужно для счастья — это Барсик и тропа.",
    "По будням — дизайнер, по выходным — лесной эльф. Барсик подтверждает.",
    "👟 Я бегу в поход, 🎨 рисую идеи, 🐱 Барсик — мой менеджер по настроению."
  ]
}
```

```
]
}
```

Параметры

Поле	Описание
<code>user_facts</code>	Словарь с фактами о пользователе
<code>current_bio</code>	Строка с существующим био
<code>tone</code>	Желаемый стиль: <code>дружелюбный</code> , <code>юмористический</code> , и т.д.
<code>length</code>	Размер био: <code>короткое</code> , <code>среднее</code> , <code>длинное</code>

```
import os
import json
import re
from typing import List
from yandex_cloud_ml_sdk import YCloudML

sdk = YCloudML(folder_id=FOLDER_ID, auth=API_KEY)

def extract_json_array(text: str) → list:
    text = text.strip()
    if text.startswith("```"):
        text = re.sub(r"^```(?:json)?", "", text).strip()
        text = re.sub(r"```$", "", text).strip()
    return json.loads(text)

def build_bio_prompt(user_facts: dict, tone: str, length: str) → str:
    fact_text = "\n".join(f"{k}: {v}" for k, v in user_facts.items())
    return (
        "Ты креативный копирайтер для дейтингового профиля.\n"
        f"Вот факты о пользователе:\n{fact_text}\n\n"
        f"Тон: {tone}\n"
        f"Желаемая длина: {length}\n\n"
        "Сгенерируй 3 разных варианта био:\n"
    )
```

```

"1. Классический\n"
"2. Минималистично, лаконично\n"
"3. Оригинальный с шуткой на основе фактов или чем-то остроумным\n"
"Верни JSON-массив из трёх строк, без markdown."
)

```

```

def generate_bio_variants(user_facts: dict, tone: str, length: str) → List[str]:
    prompt = build_bio_prompt(user_facts, tone, length)
    model = sdk.models.completions("yandexgpt-lite").configure(temperature=0.8)
    response = model.run([{"role": "user", "text": prompt}])
    return extract_json_array(response[0].text.strip())

```

```

def generate_bio_rewrites(current_bio: str, tone: str) → List[str]:
    prompt = (
        "Ты копирайтер, который помогает улучшать био для дейтингового профиля\n"
        f"Вот текущее био:\n{current_bio}\n"
        f"Тон: {tone}\n\n"
        "Сгенерируй 3 новых варианта:\n"
        "1. Уточнённый\n"
        "2. Разговорный\n"
        "3. С emoji\n"
        "Верни JSON-массив из трёх строк, без комментариев."
    )
    model = sdk.models.completions("yandexgpt-lite").configure(temperature=0.7)
    response = model.run([{"role": "user", "text": prompt}])
    return extract_json_array(response[0].text.strip())

```