

Q&A-онбординг + профиль-пакет

Endpoint

POST <https://functions.yandexcloud.net/d4enmnrrjc4cllph4250>

Назначение

Функция ведёт **динамическое интервью** (до 40-50 вопросов) и, при желании, формирует готовый набор для профиля:

- **Старт** — отдаёт 5 универсальных вопросов.
- **Диалог** — на основании уже данных ответов генерирует следующий вопрос, раскрывающий новую грань.
- **Финал** (`finalize=true` и ≥ 5 ответов) — возвращает:
 1. `facts` — 5-6 ярких фактов для профиля,
 2. `bio` — два варианта био (A/B),
 3. `openers` — 3 ice-breaker фразы для чата.

Примеры запросов

1. Получить стартовую пятёрку вопросов

```
curl -X POST https://functions.yandexcloud.net/d4enmnrrjc4cllph4250 \
-H "Content-Type: application/json" \
-d '{ "user_id": "demo-user", "answers": [] }'
```

Ответ

```
{
  "questions": [
    "Какое занятие делает тебя по-настоящему счастливым?",
    "Какое последнее приключение подарило тебе яркие эмоции?",
    "Есть ли песня/фильм, отражающие твою личность? Почему?",
    "Какая привычка помогает тебе проживать день лучше?",
    "Если бы у тебя было 24 свободных часа, как бы ты их потратил?"
  ]
}
```

2. Запрос следующего вопроса (дано < 5 ответов)

```
curl -X POST https://functions.yandexcloud.net/d4enmnrrjc4cllph4250 \
-H "Content-Type: application/json" \
-d '{
  "user_id": "demo-user",
  "answers": [
    "Люблю пешие походы в горы.",
    "Недавно ездил автостопом по Алтаю!"
  ]
}'
```

Ответ

```
{
  "question": "Есть ли традиция, к которой ты возвращаешься каждое лето?"
}
```

3. Финализация: факты + био + открывашки

(≥ 5 ответов + `finalize=true`)

```
curl -X POST https://functions.yandexcloud.net/d4enmnrrjc4cllph4250 \
-H "Content-Type: application/json" \
-d '{
  "user_id": "demo-user",
  "answers": [
    "Люблю пешие походы в горы.",
    "Автостоп по Алтаю — лучшее приключение года!",
    "Песня моей души — «Here Comes The Sun».",
    "Утренняя пробежка 5 км делает день ярче.",
    "Если день свободен — еду к озеру читать и встречать рассвет.",
    "Тренируюсь не бояться пауз в разговоре."
  ],
  "finalize": true
}'
```

Ответ-пример

```
{
  "facts": [
    "Обожает горные походы",
    "Бегают по утрам 5 км",
    "Собирает автостоп-истории",
    "Ловит рассветы с книгой у озера",
    "Учится ценить паузы в беседе"
  ],
  "bio": {
    "bioA": "Любитель гор, утренних пробежек и рассветов у озера. Всегда в поиске нового приключения и честного диалога.",
    "bioB": "👣🏔️ Бегу к солнцу, ловлю автостоп-истории и читаю на берегу озера. Напишешь, пока не наступил рассвет? 🌅📖"
  },
  "openers": [
    "Где бы ты встретил свой идеальный рассвет — в горах или у воды? 🌅",
    "Какой саундтрек включаешь, когда начинается приключение?"
  ]
}
```

"Твоя самая длинная пауза в разговоре — вызов или комфорт?"

```
]
}
```

Параметры

Поле	Тип / Формат	Описание
<code>user_id</code>	string	Идентификатор пользователя
<code>answers</code>	array[string]	Список ответов на предыдущие вопросы (может быть пустым)
<code>facts</code>	array[string] (opt)	Уже выделенные факты (клиент может хранить)
<code>finalize</code>	boolean (opt)	<code>true</code> → вернуть профиль-пакет (работает при ≥ 5 ответах)

Если `answers` пустой → возвращается `questions`.

Если `answers` < 5 и `finalize` не указан → возвращается `question`.

Если `finalize=true` и ответов ≥ 5 → возвращаются `facts`, `bio`, `openers`.

код:

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

# — creds —————
sdk = YCloudML(folder_id=YC_FOLDER_ID, auth=YC_API_KEY)

# — 5 универсальных старт-вопросов —————
DEFAULT_QUESTIONS = [
```

```

"Какое занятие делает тебя по-настоящему счастливым?",
"Какое последнее приключение подарило тебе яркие эмоции?",
"Есть ли песня или фильм, отражающие твою личность? Почему?",
"Какая привычка помогает тебе проживать день лучше?",
"Если бы у тебя было 24 свободных часа, как бы ты их потратил?"
]

```

— helper: вытянуть 1 факт из ответа —————

```

def extract_fact_llm(answer: str, taken: List[str]) → str:
    prompt = (
        "Выдели одно короткое интересное утверждение (до 12 слов), "
        "которое могло бы стать фактом профиля.\n" + answer +
        "\nВерни ТОЛЬКО это предложение без комментариев."
    )
    fact = sdk.models.completions("yandexgpt-lite").run(
        [{"role": "user", "text": prompt}]
    )[0].text.strip()
    return fact if fact and fact not in taken else ""

```

— prompt builders —————

```

def build_next_q_prompt(answers: List[str], facts: List[str]) → str:
    a_block = "\n".join(f"Q{i+1}: {t}" for i, t in enumerate(answers))
    f_block = "\n".join("• " + f for f in facts) or "—"
    return (
        "Ты – AI-коуч по знакомствам. Пользователь уже ответил:\n" + a_block +
        "\n\nФакты:\n" + f_block +
        "\n\nСформулируй ОДИН новый вопрос (≤120 символов), "
        "который раскроет новую грань личности на основе того что пользователю"
    )

```

```

def build_bio_prompt(facts: List[str]) → str:
    bullets = "\n".join("• " + f for f in facts)
    return (
        "На основе фактов составь два био.\n" +
        "Факты:\n" + bullets + "\n\n" +
        "Требования:\n- биоA: дружелюбное, 2–3 предложения.\n" +

```

```

    "- bioB: креативное + emoji.\n" +
    'Верни JSON вида {"bioA": "...", "bioB": "..."} без markdown.'
)

def build_openers_prompt(facts: List[str]) → str:
    joined = ", ".join(facts)
    return (
        "Ты – мастер ice-breakers. На основе фактов (" + joined + ") "
        "придумай 3 оригинальные фразы-открывашки (≤140 символов). "
        "Верни JSON-массив строк без лишнего текста."
    )

# — основной flow —————
def qa_flow(payload: Dict) → Dict:
    answers = payload.get("answers", [])
    facts = payload.get("facts", [])
    finalize = payload.get("finalize", False)

    # 1 нет истории – стартовые вопросы
    if not answers:
        return {"questions": DEFAULT_QUESTIONS}

    # 2 <5 ответов и не finalize – следующий вопрос
    if len(answers) < 5 and not finalize:
        prompt = build_next_q_prompt(answers, facts)
        q = sdk.models.completions("yandexgpt").configure(
            temperature=0.7
        ).run([{"role": "user", "text": prompt}])[0].text.strip()
        return {"question": q}

    # 3 finalize & ≥5 ответов – выдать facts + bio + openers
    if finalize and len(answers) >= 5:
        if not facts: # извлечь, если список пуст
            joined = "\n".join(answers)
            prompt_f = (
                "Выдели 5–6 самых ярких коротких фактов (3-10 слов) "

```

```

        "из ответов ниже. Верни JSON-массив строк без комментариев.\n"
    )
    raw = sdk.models.completions("yandexgpt-lite").run(
        [{"role": "user", "text": prompt_f}]
    )[0].text.strip()
    raw = re.sub(r"```[a-z]*|```", "", raw).strip()
    facts = json.loads(raw)

    # био
    bio_raw = sdk.models.completions("yandexgpt").run(
        [{"role": "user", "text": build_bio_prompt(facts)}]
    )[0].text.strip()
    bio = json.loads(re.sub(r"```[a-z]*|```", "", bio_raw).strip())

    # открывашки
    op_raw = sdk.models.completions("yandexgpt").run(
        [{"role": "user", "text": build_openers_prompt(facts)}]
    )[0].text.strip()
    openers = json.loads(re.sub(r"```[a-z]*|```", "", op_raw).strip())

    return {"facts": facts, "bio": bio, "openers": openers}

# fallback
return {"error": "Invalid state"}

```