

YandexGPT API

# Создание простого чат-бота с Yandex GPT API

Фадеев Виталий Олегович

# Цели работы:

### Теоретическая:

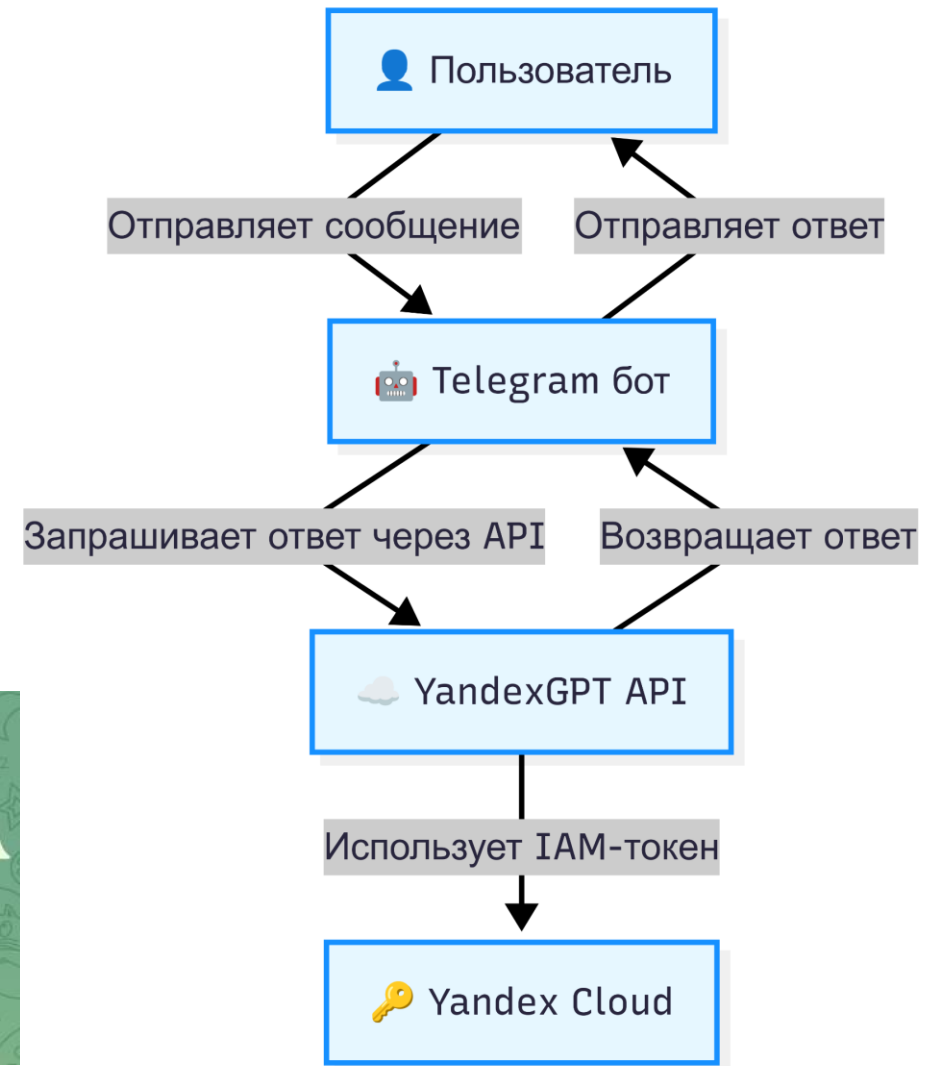
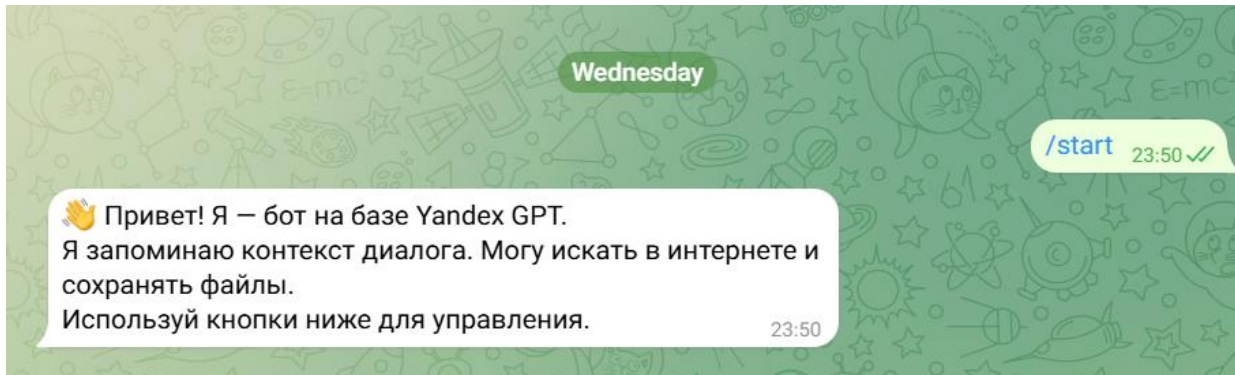
Изучить архитектуру интеграции Telegram Bot API с языковой моделью Yandex GPT, понять принципы аутентификации через JWT токены и механизмы обработки запросов в облачных AI-сервисах.

### Практическая:

Реализовать функционального Telegram-бота с базовой обработкой текстовых сообщений, настроить безопасное подключение к Yandex GPT API через сервисные аккаунты и создать простой интерфейс взаимодействия пользователя с языковой моделью.

# Что делает бот?

1. Получает сообщение от пользователя
2. Отправляет в YandexGPT API
3. Возвращает ответ



# Первичная настройка


## 1. Найдите @BotFather в Telegram

Отправьте команду **/newbot**

Придумайте **имя и username** бота

Получите токен бота

## 2. Установка библиотек



```
pip install python-telegram-bot  
pip install requests  
pip install python-dotenv
```

# ОСНОВЫ БОТА

## Импорты:

```
from telegram import Update, ReplyKeyboardMarkup, KeyboardButton
from telegram.ext import Application, CommandHandler, MessageHandler, filters, ContextTypes
from yandexgpt_client import ask_yandexgpt
```

## Команда /start

```
async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
    welcome_text = "👋 Привет! Я – бот на базе Yandex GPT."
    await update.message.reply_text(
        welcome_text,
        reply_markup=get_main_keyboard()
    )
```

# Обработка сообщений

```
async def handle_message(update: Update, context: ContextTypes.DEFAULT_TYPE):  
    user_message = update.message.text.strip()  
  
    if user_message == "? Помощь":  
        await help_command(update, context)  
        return  
  
    await update.message.reply_text("🧠 Думаю... ⌚")  
  
    bot_response = ask_yandexgpt(user_message)  
    await update.message.reply_text(bot_response)
```

**Главный  
обработчик**

```
def main():  
    app = Application.builder().token(TELEGRAM_TOKEN).build()  
  
    app.add_handler(CommandHandler("start", start))  
    app.add_handler(MessageHandler(filters.TEXT, handle_message))  
  
    app.run_polling()
```

**Запуск бота**

## Запрос к YandexGPT

### Формирование запроса:

```
def ask_yandexgpt(current_user_input: str) -> str:
    messages_payload = []
    messages_payload.append({"role": "system", "text": SYSTEM_PROMPT})
    messages_payload.append({"role": "user", "text": current_user_input})

    MODEL_URI = f"gpt://{FOLDER_ID}/yandexgpt/latest"
```

### Параметры модели

```
payload = {
    "modelUri": MODEL_URI,
    "completionOptions": {
        "stream": False,
        "temperature": 0.7,
        "maxTokens": 2000
    },
    "messages": messages_payload
}
```

### Отправка запроса

```
response = requests.post(YANDEX_GPT_API_URL, headers=get_headers(), json=payload)
result = response.json()
answer = result["result"]["alternatives"][0]["message"]["text"]
return answer
```