import org.telegram.telegrambots.bots.TelegramLongPollingBot

import org.telegram.telegrambots.meta.TelegramBotsApi

import org.telegram.telegrambots.meta.api.methods.send.SendMessage

import org.telegram.telegrambots.meta.api.objects.Update

import org.telegram.telegrambots.updatesreceivers.DefaultBotSession

// Этот класс — сердце нашего бота

class ServiceLinkBot : TelegramLongPollingBot("YOUR\_BOT\_TOKEN") {

// Здесь бот сообщает свое имя пользователя

override fun getBotUsername(): String = "service\_link\_bot"

// Этот метод вызывается каждый раз, когда кто-то пишет боту

override fun onUpdateReceived(update: Update) {

// Проверяем, что сообщение действительно текстовое

if (update.hasMessage() && update.message.hasText()) {

val messageFromUser = update.message.text

val chatId = update.message.chatId

val userName = update.message.from.firstName

// Обрабатываем команды

when (messageFromUser) {

"/start" -> sendWelcomeMessage(chatId, userName)

"📝 Оставить заявку" -> askForClientName(chatId)

"ℹ️ О нас" -> sendAboutInfo(chatId)

"📞 Контакты" -> sendContacts(chatId)

else -> {

// Здесь будет логика обработки ответов на вопросы (имя, техника, описание поломки)

// Для простоты примера просто отвечаем

sendTextMessage(chatId, "Спасибо за ваше сообщение! Для начала работы нажмите «📝 Оставить заявку».")

}

}

}

}

// Функция отправки приветственного сообщения с кнопками

private fun sendWelcomeMessage(chatId: Long, userName: String) {

val text = "Привет, $UserName! Добро пожаловать в Service Link Bot — сервис по ремонту техники. Чем могу помочь?"

// Создаем клавиатуру с кнопками

val keyboard = ReplyKeyboardMarkup(

listOf(

listOf(KeyboardButton("📝 Оставить заявку")),

listOf(KeyboardButton("ℹ️ О нас"), KeyboardButton("📞 Контакты"))

)

)

keyboard.resizeKeyboard = true // Делаем кнопки компактными

keyboard.oneTimeKeyboard = false // Клавиатура не скрывается после нажатия

val message = SendMessage(chatId.toString(), text)

message.replyMarkup = keyboard

execute(message)

}

// Функция, которая спрашивает имя клиента

private fun askForClientName(chatId: Long) {

sendTextMessage(chatId, "Для оформления заявки нам потребуется некоторая информация. Как вас зовут?")

// Здесь мы должны запомнить, что на следующем шаге ждем имя

// Для этого нужно использовать базу данных или просто временное хранилище

}

// Функция отправки информации о компании

private fun sendAboutInfo(chatId: Long) {

val text = """

🛠️ «Service Link» — это профессиональный сервис по ремонту бытовой и цифровой техники.

Мы работаем с:

• Смартфонами и планшетами

• Ноутбуками и компьютерами

• Телевизорами и аудиотехникой

• Бытовая техника (холодильники, стиральные машины)

Наши мастера имеют большой опыт и используют оригинальные запчасти.

""".trimIndent()

sendTextMessage(chatId, text)

}

// Функция отправки контактов

private fun sendContacts(chatId: Long) {

val text = """

📍 Мы находимся: г. Москва, ул. Примерная, д. 10

🕒 Время работы: Пн-Пт с 9:00 до 21:00, Сб-Вс с 10:00 до 18:00

📞 Наш телефон: +7 (495) 123-45-67

✉️ Email: info@Service-link.ru

🌐 Сайт: www.service-link.ru

""".trimIndent()

sendTextMessage(chatId, text)

}

// Вспомогательная функция для отправки простого текстового сообщения

private fun sendTextMessage(chatId: Long, text: String) {

val message = SendMessage(chatId.toString(), text)

execute(message)

}

}

// Главная функция, которая запускает бота

fun main() {

val botsApi = TelegramBotsApi(DefaultBotSession::class.java)

try {

botsApi.registerBot(ServiceLinkBot())

println("Бот успешно запущен!")

} catch (e: Exception) {

e.printStackTrace()

}

}