

# tg\_bot @LifeStatsBot

см. gif с полным демо бота.

## Профиль пользователя

### Создание и настройка профиля (/set\_profile)

При первом запуске бот предлагает создать профиль, чтобы настроить индивидуальные параметры.

- **Пользователь вводит:**
  - Дату рождения.
  - Вес.
  - Рост.
  - Желаемый уровень потребления воды.
  - Желаемый уровень потребления калорий.
  - Желаемый уровень активности в день.
  - Город.
  - Пол.
- **Особенности ввода норм:**
  - Если пользователь указывает **0 при вводе желаемой нормы калорий, воды или активности**, бот **автоматически рассчитает** эти параметры на основе стандартных формул.
  - Если пользователь **указывает значения, они используются вместо** автоматически рассчитанных **норм**.

---

### Обновление данных профиля (/set\_profile)

Параметры профиля можно изменять в любое время (например, если изменился вес или уровень активности).

Однако:

- **Рассчитанные нормы (калорий, воды и активности) пересчитываются только в начале нового дня. Это обеспечивает согласованность данных для дневной статистики.**
- **Пользователь может вручную указать новые нормы или оставить расчёт на усмотрение бота, введя 0.**

---

## Логирование данных

Бот позволяет вести учёт всех важных параметров для отслеживания здоровья:

### Логирование воды (/log\_water)

- Пользователь может ввести количество выпитой воды.
- Бот суммирует потребление за день и сообщает, сколько осталось до нормы.

### Автоматическая коррекция нормы:

- Если пользователь предоставляет свою **локацию**, бот использует погодные данные (API погоды) для анализа температуры в текущем городе.
- Если температура превышает **25°C**, бот автоматически увеличивает дневную норму воды, так как в жару организму требуется больше жидкости.

### Логирование калорий (/log\_calories)

- Ручной ввод калорий.

- Добавление еды по названию (например: "яблоко" или "куриная грудка").
- Бот автоматически ищет калорийность продукта через API OpenFoodFacts.
- Хранение списка съеденного за день и его калорийность.

#### Логирование активности (/log\_workout) [↗](#)

- Ввод данных о физической активности (например: "бег 30 минут").
  - Поддержка расчёта потраченных калорий на основе вида активности и продолжительности.
  - Напоминание о необходимости большего потребления воды в результате физической нагрузки.
- 

#### Анализ прогресса (/check\_progress) [↗](#)

- Графическое отображение прогресса по:
  - **Вода (мл):** сколько выпито и сколько осталось до дневной нормы.
  - **Калории (ккал):** сколько калорий потреблено, сколько осталось до нормы.
  - **Активность (минуты):** текущая активность, цель и сожжённые калории.

Пользователь может выбрать отображение статистики за:

- День (/check\_progress).
- Неделю в виде графиков (/stats).