Алгоритм:

1. Телеграмм -> BotFather
2. /newbot
3. Имя бота: Lab30
4. Ввод: Lab30Bot (Bot или \_bot на конце)
5. Создается бот и возвращается токен, который надо вставить в лабу

Боты — специальные аккаунты в Telegram, созданные для того, чтобы автоматически обрабатывать и отправлять сообщения. Пользователи могут взаимодействовать с ботами при помощи сообщений, отправляемых через обычные или групповые чаты. Логика бота контролируется при помощи HTTPS запросов к нашему [API для ботов](https://tlgrm.ru/docs/bots/api).

2 общих подхода для создания бота (web hook; long pool запрос)

1. **Поясните принцип работы LongPoll-сервера?**

**Длинные опросы** – это самый простой способ поддерживать постоянное соединение с сервером, не используя при этом никаких специфических протоколов (типа WebSocket или Server Sent Events).

**Частые опросы**

Самый простой способ получать новую информацию от сервера – периодический опрос. То есть, регулярные запросы на сервер вида: «Привет, я здесь, у вас есть какая-нибудь информация для меня?». Например, раз в 10 секунд.

В ответ сервер, во-первых, помечает у себя, что клиент онлайн, а во-вторых посылает весь пакет сообщений, накопившихся к данному моменту.

Это работает, но есть и недостатки:

Сообщения передаются с задержкой до 10 секунд (между запросами).

Даже если сообщений нет, сервер «атакуется» запросами каждые 10 секунд, даже если пользователь переключился куда-нибудь или спит. С точки зрения производительности, это довольно большая нагрузка.

Так что, если речь идёт об очень маленьком сервисе, подход может оказаться жизнеспособным, но в целом он нуждается в улучшении.

**[Длинные опросы](https://learn.javascript.ru/long-polling" \l "dlinnye-oprosy)**

«Длинные опросы» – гораздо лучший способ взаимодействия с сервером.

Они также очень просты в реализации, и сообщения доставляются без задержек.

Как это происходит:

1. Запрос отправляется на сервер.
2. Сервер не закрывает соединение, пока у него не возникнет сообщение для отсылки.
3. Когда появляется сообщение – сервер отвечает на запрос, посылая его.
4. Браузер немедленно делает новый запрос.

Для данного метода ситуация, когда браузер отправил запрос и удерживает соединение с сервером, ожидании ответа, является стандартной. Соединение прерывается только доставкой сообщений.

Если соединение будет потеряно, скажем, из-за сетевой ошибки, браузер немедленно посылает новый запрос.

**Long Polling** — это технология, которая позволяет получать данные о новых событиях с помощью «длинных запросов». Сервер получает запрос, но отправляет ответ на него не сразу, а лишь тогда, когда произойдет какое-либо событие (например, придёт новое сообщение), либо истечет заданное время ожидания.

В сфере веб-разработки наряду с API распространилось понятие **вебхук** (англ. [webhook](https://en.wikipedia.org/wiki/Webhook" \t "_blank)). Рост популярности этого стандарта связан с тем, что всё больше действий в вебе можно описать в терминах событий. Триггерами, запускающими вебхук, могут служить, например, отправка кода в репозиторий или публикация комментария.

Вебхук – это механизм оповещения о происходящих в системе событиях посредством [функций обратных вызовов](https://ru.wikipedia.org/wiki/Callback_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)). Когда случается интересующее клиента событие, сервер отправляет HTTP-запрос на URL-адрес, предоставленный клиентом для приема вебхуков.