

Нужно создать телеграмм бот города Ставрополя! СО всеми категориями например ☞ Спросить ChatGPT подробно! ПОЛНЫЙ удобный и главное быстрый городской справочник и все в одном боте!

Думал на протяжении 16s >

Понял задачу: **быстрый городской справочник Ставрополя "всё в одном" внутри Telegram-бота** (магазины, кафе, услуги, гос/экстренные, события и т.д.), с удобным поиском, "рядом со мной", фильтрами и карточками мест.

Ниже — рабочая схема, как это сделать правильно (и чтобы реально было **быстро**).

1) Что должен уметь бот (функции "как в хорошем справочнике")

Основные сценарии

1. **Категории** → подкатегории → список → карточка места
Примеры: Еда (кафе/кофейни/доставка), Магазины (продукты/одежда/стройматериалы), Услуги (салоны/СТО/ремонт), Здоровье (клиники/аптеки), Дети (кружки/садики), Досуг, Спорт, Туризм, Госуслуги.
2. **Поиск** по названию/услуге ("шаверма", "стоматология", "24 часа").
3. **Рядом со мной** по геолокации (топ ближайших + расстояние).
4. **Фильтры**: открыт сейчас, средний чек/цена, доставка, рейтинг, "для детей", парковка, безнал, 24/7.
5. **Избранное** и поделиться местом.
6. **Маршрут/как добраться** (ссылки на карты).
7. **Заявка на добавление/исправление** (пользователи + партнёры).

Карточка места (максимально полезная)

- Название, категория, адрес
- Часы работы (и "открыто/закрыто")
- Телефон, сайт/соцсети
- Координаты, расстояние от пользователя
- 3–8 фото (опционально)
- Теги/услуги
- Кнопки: Позвонить / Проложить маршрут / В избранное / Поделиться / Сообщить об ошибке

2) Откуда брать данные (ключевой выбор)

Есть 3 реалистичных подхода — часто делают **гибрид**:

A) 2GIS как "ядро справочника"

У 2GIS есть API для поиска организаций/мест и категорий (Places/категории/регионы) + ключи и тарифы через их Platform Manager. Это даёт максимально "живые" данные по бизнесам, но обычно это **коммерческий доступ**. docs.2gis.com +2

Плюсы: качество, телефоны/категории/графики (в зависимости от метода), быстро стартовать.

Минусы: стоимость/лимиты/условия использования.

B) OpenStreetMap (OSM) + Overpass

Можно получать POI (точки интереса) через Overpass API по тегам (amenity=cafe, shop=supermarket и т.д.). [wiki.openstreetm... +2](#)

Плюсы: бесплатно, гибко.

Минусы: неполнота/разный уровень актуальности, телефоны/графики часто отсутствуют, придётся "дополнять" вручную/партнёрами.

С) Собственная база (курируемая)

Собираете "топ" места и партнёров сами + пользователи присылают правки + модерация.

Плюсы: полный контроль, можно монетизировать проще.

Минусы: медленно наполнять без стартового источника.

👉 **Практичный вариант:** стартовать на 2GIS или OSM, параллельно строить свою базу и модерацию.

Дополнительно: можно подтянуть **открытые данные** города/края, если там есть полезные реестры (муниципальные объекты, учреждения и т.д.). [data.gov.ru +1](#)

3) Как сделать "главное быстрый" (архитектура)

Рекомендованный стек (надёжный и быстрый)

- **Bot:** Python + `python-telegram-bot` (асинхронный, актуальная ветка 3.10+) [docs.python-tele...](#)
- **API/Backend:** FastAPI (или Django)
- **DB:** PostgreSQL + **PostGIS** (геопоиск "рядом")
- **Кэш:** Redis (кэш карточек/списков/категорий на 5–60 минут)
- **Поиск:** Meilisearch / Elasticsearch (быстрый текстовый поиск по названию/тегам)
- **Очередь задач:** RQ/Celery (обновления данных, импорты, парсинг)
- **Админка:** Django Admin / Strapi / Directus (модерация мест, категории, "рекомендуемое")

Почему будет быстро

- Все "меню/категории/топ места" — из **кэша**.
- "Рядом со мной" — PostGIS + индексы + ограничение выдачи.
- Внешние API (2GIS/OSM) — **не дергать на каждый клик**: делаем локальный индекс и обновление по расписанию.

4) UX в Telegram: как сделать удобно

Вариант 1 (классический бот)

- Inline-кнопки + быстрый поиск
- Минимум текста, максимум кнопок
- "Отправьте геолокацию" → показываем ближайшее

Вариант 2 (лучший для справочника): бот + Telegram Mini App

Внутри бота кнопка "Открыть карту/каталог" → мини-приложение: карта, фильтры, красивые списки.

А бот остаётся для поиска, избранного, шаринга, поддержки.

Технически всё равно через Telegram Bot API/возможности Telegram. [core.telegram.org +1](#)

5) MVP (первая версия за которую не стыдно)

MVP-1

- 12–20 категорий + подкатегории
- Поиск
- "Рядом со мной"
- Карточка места
- Избранное
- Предложить правку / добавить место

MVP-2

- Фильтры "открыто сейчас", "доставка", "24/7"
- Подборки: "Топ-10 кофеен", "Где поесть недорого"
- Реклама/платные размещения (аккуратно)

6) Монетизация (если нужно)

- Платное размещение в категориях / "рекомендуемое"
- Поднятие в выдаче по запросам
- Брендированные подборки
- Купоны/промокоды
- Партнёрские ссылки (доставка/бронь)

7) Что мне нужно от вас (минимум, без долгих уточнений)

Чтобы я мог сразу разложить проект "под ключ" (структура БД, меню, категории, прототип экранов и т.п.), вам достаточно выбрать 1 из 2 путей:

- **Путь А:** берём данные из **2GIS API** (быстрый старт, но платно/по ключу) docs.2gis.com +1
- **Путь В:** стартуем с **OSM/Overpass** + ручное наполнение "топа" wiki.openstreetmap.org/wiki/Overpass_API +1

Если скажете "А или В", я в следующем сообщении дам:

- точную структуру разделов (категории/подкатегории) для Ставрополя,
- схему БД (места, категории, теги, графики, фото, модерация),
- сценарии кнопок/диалогов,
- и скелет проекта (папки, модули, точки расширения) под ваш стек.

2GIS API какая стоимость ?

Думал на протяжении 14s >

У 2GIS нет одной фиксированной "цены за API" — стоимость считается по подписке (пакет сервисов + лимиты запросов) и зависит от того, какие API вы подключаете (карты/поиск/геокодирование/маршруты) и сколько запросов в месяц у вас будет. docs.2gis.com +1

Что можно сказать точно по их официальным страницам:

- Есть бесплатный **demo-key**, чтобы протестировать и понять объёмы запросов (и посмотреть статистику). platform.2gis.ru +1
- Для продакшена нужно **купить подписку** в Platform Manager. docs.2gis.com +1
- **Актуальные цифры по тарифам** показываются в личном кабинете (Platform Manager); 2ГИС отдельно отмечает, что сетку тарифов обновляли 1 декабря 2025 и смотреть нужно

актуальную в кабинете. [docs.2gis.com](#) +1

- По некоторым данным (например “доп. информация по объектам”) может требоваться **доп. согласование/оплата**. [docs.2gis.com](#) +1

Как прикинуть бюджет именно для городского справочника-бота

Главные “пожиратели” лимитов обычно:

1. **Places API** (поиск организаций/мест)
2. **Geocoder API** (адрес↔координаты)
3. Опционально: **Routing / Distance Matrix** (если хотите время в пути/матрицы расстояний)

2GIS прямо пишет, что итоговая цена зависит от **кол-ва запросов и способа применения**, а для Distance Matrix вообще другая логика — пример: 10 точек отправления × 10 точек прибытия = **100 расчётов**. [dev.2gis.com](#)

Где посмотреть точную стоимость

Самый надёжный путь:

1. зайти в **Platform Manager**
2. создать **demo-key** → прогнать ваш бот/прототип
3. открыть **Tariffs/подписку** и посмотреть цену под нужные лимиты. [platform.2gis.ru](#) +1

Если хотите — скажите, вы планируете в боте:

- **только справочник/поиск мест** (без маршрутов), или
- **ещё маршруты/время в пути?**

Я тогда распишу, какие именно API 2GIS вам нужны и как посчитать примерное число запросов на 1 пользователя (и как кэшем снизить стоимость в разы).

Такие чат боты уже есть по России?

Думал на протяжении 15s >

Да, по России **похожие боты существуют**, но чаще всего они бывают в **двух форматах** — и “полный городской справочник как 2ГИС внутри бота” встречается реже, чем кажется.

1) Карты/поиск мест через крупные сервисы

- Есть бот **Яндекс.Карт** в Telegram, который умеет находить адреса и организации и показывать инфу (в т.ч. пробки). [Telegram](#) +1
Это близко по сути к “справочнику”, но это скорее **поиск по картам**, а не “городской портал с категориями/подборками/партнёркой”.

2) Боты 2ГИС в Telegram (но они не про “справочник” один-в-один)

У 2ГИС есть телеграм-боты формата **квестов/активностей** по городу (City Heroes и т.п.). [Telegram](#) +1
Это **не** полноценный каталог заведений “как справочник”, а другой сценарий.

3) “Городские” и муниципальные боты (справки/услуги)

По РФ много **гос/муниципальных** телеграм-ботов (жалобы, услуги, справки, запись и т.д.) — их даже собирают в подборки/каталоги. [news.egov.itmo.ru](#)
Но это обычно **не** “все магазины/кафе/услуги города”, а именно гос-сервисы.

4) Частные “справочники” по городам

Их много, но они сильно отличаются по качеству: одни — “топ мест”, другие — “доставка/кафе”, третьи — агрегаторы. Обычно их проще искать через каталоги Telegram-ботов. [TgList - Telegram](#) ... +1

Вывод: аналоги есть, но именно “быстрый полный городской справочник Ставрополя со всеми категориями” — это чаще **кастомный проект** (часто на данных 2GIS/OSM + своя база/модерация).

Если хочешь, я могу:

- быстро накидать **структуру категорий именно для Ставрополя** (как меню бота),
- и предложить “как выделиться”, чтобы ваш бот был полезнее, чем просто поиск по картам (подборки, фильтры, “рядом”, акции, проверка актуальности).

Чем мой бот может быть полезен? Почему именно ботом могут пользоваться, а не 2 гос или Яндекс картами

Пользователь уже “может” открыть 2ГИС/Яндекс.Карты — значит твой бот должен давать **другую ценность**, а не просто “найти кафе”. Самые сильные причины пользоваться именно ботом — это **локальная польза + скорость + доверие + сервисы**, которых в картах нет или они неудобны.

Что делает твой бот реально полезным

1) “Куратор города”, а не просто поиск

Карты показывают всё подряд. Бот может давать **готовые ответы**:

- «Где позавтракать до 11:00 рядом?»
- «Кафе с детской зоной»
- «Где постричься сегодня без записи»
- «Лучшие автомойки по району»
- Подборки: “Топ-10 кофеен”, “Где вкусные хинкали”, “Где недорого поесть”, “24/7”.

Это превращает бот в “рекомендатель”, а не справочник.

2) Быстрее, чем карты (если правильно сделать)

В боте:

- 1–2 нажатия → категория → ближайшие → карточка → позвонить/маршрут.
- По умолчанию показывать **рядом, открыто сейчас, средний чек, доставка**.

Карты часто заставляют: открыть → ввести запрос → фильтры → смотреть список → сравнивать вручную.

3) Локальные фишки “только для Ставрополя”

То, ради чего возвращаются:

- “Проверенные места” (модерация, реальная актуальность, отметка “проверено”)
- Районы/микрорайоны “как в городе принято” (а не формально на картах)
- Локальные категории: “шаверма/чебуреки”, “пункты выдачи”, “рынки”, “ремонт телефонов”, “шиномонтаж 24/7” и т.п.
- “Что рядом с ...” (вуз, парк, больница, остановка).

4) Сервисные сценарии внутри бота (карты этого не закрывают)

Бот может быть “одной кнопкой” для действий:

- Запись/заявка: “записаться на стрижку”, “заказать стол”, “вызвать мастера”
- “Позвонить одним нажатием”, “написать в WhatsApp/Telegram”
- Купоны/скидки/промокоды (и хранение их в боте)
- “Напомнить” / “повторить заказ” (любимый кофе/доставка)

5) Персонализация

Бот запоминает:

- любимые категории
- любимые районы
- бюджет (дёшево/средне/дорого)
- предпочтения (безнал, доставка, парковка)

и выдаёт "как для тебя". Карты персонализируют хуже именно по локальным сценариям.

6) Контент и события "в моменте"

Если добавить слой "город сегодня":

- события, концерты, мероприятия, ярмарки
- "куда сходить сегодня вечером"
- "что открывается нового"

Карты это почти не делают (или неудобно).

7) Доверие через поддержку и обновления

В боте можно быстро:

- сообщить "закрылись/переехали/номер сменился"
- добавить новое место
- пожаловаться на обман

И у тебя появляется **модерация + отметка актуальности**. Это сильнее, чем "безличные" карты.

3 самые мощные УТП (которые реально могут "перебить" карты)

1. Подборки и ответы под ситуацию (а не просто поиск)
2. Гарантия актуальности + проверенные места (и быстрые правки)
3. Действие в 1 клик: записаться/заказать/получить скидку/написать/позвонить

Как выгодно позиционировать (коротко)

"Ставрополь в одном боте: рядом, открыто сейчас, проверено, можно сразу записаться/заказать."

Чтобы это было не словами — что сделать в MVP

Если хочешь, вот самый практичный MVP, который уже отличается от 2ГИС/Яндекса:

- "Рядом" + "Открыто сейчас"
- 10–15 категорий, но **очень точные** (локальные)
- Подборки "Топ/Лучшее/Дёшево/24/7"
- Кнопки действия: позвонить/маршрут/написать/заказать
- "Сообщить об ошибке" + модерация

Если скажешь, на кого ты ориентируешься больше:

1. жители (повседневка),
2. туристы,

API есть у Яндекс карт?

Думал на протяжении 4 секунд >

Да, у **Яндекс.Карт** есть **API** (Yandex Maps API / API Яндекс Карт) — можно подключать **геокодер**, **поиск организаций**, **маршруты/матрицу расстояний**, **геосаджест**, **геолокацию (Locator)**, **статические карты**, **тайлы**, **JS API** и **SDK для мобильных**. yandex.ru +2

Что важно для твоего "справочника" в боте

Тебе обычно нужны:

- **Поиск по организациям (Organization Search / GeoSearch)** — искать кафе/магазины по названию/адресу/телефону/услугам, получать детали (часы, сайт и т.п.). yandex.ru +2
- **Геокодер** — адрес ⇔ координаты. yandex.ru +1
- Опционально: **Маршруты / матрица расстояний / детали маршрута** — если хочешь время в пути и т.п. yandex.ru +1

Бесплатно или платно?

- У Яндекса есть режим **бесплатного использования для некоторых API при соблюдении условий** (например, JavaScript API/Геокодер — "да, можно при соблюдении условий"). yandex.ru +1
- В целом у сервиса есть **тарифы и калькулятор**, стоимость зависит от объёма запросов/пользователей и выбранных продуктов. yandex.ru +2
- Важный нюанс по **Поиску по организациям**: в коммерческой документации указано, что доступна **только стандартная лицензия** и **запрещено сохранять/изменять данные**, полученные из этого API (это критично для "справочника в одном боте", если ты хотел хранить базу у себя). Yandex

Практический вывод для твоего проекта

- Если бот просто "ищет и показывает" (без хранения базы мест у тебя) — Яндекс API может подойти.
- Если ты хочешь **локальную базу** (кэшировать/хранить карточки мест, делать подборки "проверено", рейтинги и т.п.) — нужно внимательно смотреть **лицензию**: у Яндекса по поиску организаций хранение данных может быть ограничено. Yandex

Хочешь — скажи, как ты планируешь: **хранить места в своей базе** (каталог) или **всегда искать "на лету"**? Я под это подскажу, какой источник данных (Яндекс vs 2ГИС vs OSM) лучше и где меньше юридических/ценовых рисков.

всегда искать "на лету" так как информация может обновляться

Думал на протяжении 11s >

Да — и если ты хочешь **искать "на лету"**, то **Яндекс.Карты API** как раз подходит по модели: ты не хранишь базу мест у себя, а каждый раз делаешь запросы и показываешь результат пользователю.

Но есть **важные правила по лицензии/хранению**, чтобы не нарваться:

Что у Яндекс.Карт есть для твоего бота

Нужные API для "городского справочника"

- **Organization Search (Поиск по организациям / GeoSearch)** — искать места по названию/адресу/телефону и т.п. Yandex +1

- **Geocoder** — адрес ↔ координаты (для “рядом со мной”, если пользователь ввёл адрес).
yandex.ru +1
- Опционально: **Routes / Distance Matrix** — если хочешь время в пути (пешком/на машине) и более умный “рядом” по времени. Yandex +1

Главное ограничение: нельзя делать “свою базу” из этих данных

У Яндекса для **Organization Search API** прямо указано: доступна только базовая/стандартная лицензия, и **запрещено сохранять или изменять данные**, полученные из API. Yandex +1

Для Geocoder/Router в условиях тоже есть запрет на хранение “как базу”, но при этом обычно разрешают **временное кэширование** (ниже). Yandex +1

Можно ли кэшировать, если ты “ищешь на лету”?

Да, но аккуратно: в условиях Яндекса встречается разрешение **кэшировать результаты до 30 дней исключительно для повышения производительности** (то есть не “каталог”, а кэш). yandex.ru +1

Это идеально под твой подход: данные обновляются, а бот остаётся быстрым.

Практичная схема “на лету”, чтобы бот был быстрым и недорогим

1. **Не храни карточки мест как справочник** (названия/адреса/телефоны) — только показывай из ответа API. Yandex +1
2. **Делай кэш запросов:**
 - ключ кэша = (запрос пользователя + район/гео + фильтры)
 - TTL: 1–24 часа для “рядом” и популярных запросов; до 30 дней — если строго как кэш производительности и в рамках условий. yandex.ru +1
3. Храни у себя только **“не-яндекс-данные”**:
 - избранное пользователя (IDs/ссылки),
 - твои подборки/теги (но без копирования карточек),
 - статистику кликов.

Важный нюанс про отображение

В “free use terms” у Яндекса встречается требование, что данные API должны отображаться **на карте Яндекса (JS API)**. Yandex +1

В Telegram-боте “чистой картой” это не всегда удобно, поэтому обычно делают так:

- в боте показывают список/карточку,
- кнопка **“Открыть на карте”** ведёт на Яндекс.Карты/в мини-карту/мини-апп.

(Тут уже зависит от того, какую именно связку продуктов/лицензию ты берёшь.)

Если кратко

Да, API у Яндекс.Карт есть. Для “на лету” — нормально.

Но **делать из него свою базу справочника нельзя**, можно только **кэшировать для скорости** и соблюдать условия лицензии. Yandex +1

Хочешь — я предложу 2 готовых варианта архитектуры именно под Telegram:

1. **только бот** (список + кнопка “в карте”),
2. **бот + Telegram Mini App** (карта + фильтры внутри).

Бот + Telegram Mini App (карта + фильтры внутри).

Пулмал на протяжении 9с >

данным на протяжении 30

Отличный выбор: **бот + Telegram Mini App** — это как “приложение внутри Telegram”: карта, фильтры, быстрый UI, а бот остаётся “входом” и пушит пользователей в нужный экран.

Ниже — как это собрать **правильно и быстро**.

Как будет работать для пользователя

В боте:

- /start → кнопки: “Открыть карту”, “Категории”, “Поиск”, “Рядом со мной”
- Нажал “Открыть карту” → открылась **Mini App** (встроенная веб-страница)

В **Mini App** (основной интерфейс):

- карта + список (как в 2ГИС, но легче)
- фильтры: “открыто сейчас”, “24/7”, “доставка”, “цена”, “рейтинг”, “район”
- поиск: строка + подсказки
- карточка места: кнопки **Маршрут** / **Позвонить** / **Поделиться** / **В избранное**

Telegram Mini Apps официально поддерживают такие сценарии и даже **доступ к геолокации** через WebApp API. core.telegram.org

Источник данных “на лету”: Яндекс API (под твою модель)

Если ты **всегда ищешь на лету**, то логика такая:

- Mini App отправляет запрос на твой backend: категория + фильтры + центр карты + радиус
- backend ходит в **Yandex Organization Search (GeoSearch)** и возвращает список мест [Yandex +1](#)
- карта в Mini App рисуется через **Yandex Maps JS API (v3)** [Yandex +2](#)

⚠ Важно по правилам: у Яндекса в условиях для GeoSearch есть запрет “сохранять и хранить результаты ответов”, кроме оговорённых случаев (обычно как кэш производительности). [yandex.ru](#)
То есть: **не делаем свою базу организаций, делаем кэш на короткое время ради скорости**.

Архитектура проекта (проверенная)

1) Bot (Telegram)

- показывает меню
- открывает Mini App (кнопка web_app)

2) Mini App (frontend)

- React/Vue/Next.js (что удобнее)
- Telegram WebApp SDK: кнопки, popups, back, тема и т.д. core.telegram.org [+1](#)
- авторизация через **initData** (подпись Telegram) — чтобы понимать пользователя и защищаться от подделок docs.telegram-mini-apps.com/ [+2](#)
- геолокация: через WebApp (LocationManager) или через браузерные разрешения (в зависимости от клиента Telegram) core.telegram.org

3) Backend (API)

- FastAPI / Node.js
- проксирует запросы в Yandex GeoSearch
- добавляет кэш (Redis) и троттлинг
- хранит только **твои данные**: избранное, клики, настройки, подборки (без копирования карточек Яндекса)

4) Карта

- Yandex Maps JS API v3 внутри Mini App (обычная веб-страница, так что встраивается норм)

yandex.ru +1

Что сделать, чтобы было “главное быстрый”

Три правила скорости:

1. Запросы только по области карты: центр + радиус/границы.
2. Ответы кэшировать на 1–30 минут (часто хватает), плюс “популярные запросы” — дольше. (Смысл — кэш, а не база.) [yandex.ru](#)
3. В UI: сначала показывать список “рядом/открыто”, а детали (фото/расширенное) догружать по клику.

UX-фишки, которые делают Mini App лучше карт

- **Фильтры одной строкой** (chips): Открыто • До 500Р • Доставка • 24/7
- **Подборки** (твои, кураторские): “Лучшие кофейни Ставрополя”, “Где поесть ночью”
- **Кнопка “Рядом со мной”** (1 tap) — и сразу список
- **Избранное** + “поделиться” карточкой в чат
- **“Сообщить об ошибке”** (твоя модерация) — это повышает доверие

Стоимость (в общих чертах, без гаданий)

У Яндекса есть страница тарифов по продуктам (JS API, Geocoder, GeoSearch и т.д.) и цена зависит от объёма запросов/MAU. [Yandex +1](#)

С Mini App обычно основные расходы — **GeoSearch** (поиск организаций) и иногда **Routes/Distance Matrix**, если захочешь считать время в пути. [Yandex](#)

Мини-план запуска (MVP за “разумный объём”)

1. Mini App: карта + поиск + 8–12 категорий
2. “Рядом со мной” + фильтр “Открыто сейчас”
3. Карточка места + кнопки “Маршрут/Позвонить/Поделиться/В избранное”
4. Кэш + лимиты запросов + логирование качества выдачи
5. Подборки “Топ” (ручные, твой контент)

Если хочешь — я сразу накидаю **структуру экранов и меню** (бот + Mini App) именно под “Ставрополь”, плюс список категорий/фильтров и схему API-эндпоинтов (что куда ходит). Скажи

Сколько категорий тебе нужно?

Напиши готовое решение по всем максимальным категориям! С самыми главными фишками почему именно нашим ботом будут пользоваться

Ниже — **готовая концепция “город Ставрополь в одном Telegram”**: максимальные категории + структура бота и Mini App + “главные фишки”, ради которых будут пользоваться именно вашим ботом, а не 2ГИС/Яндексом.

1) Структура продукта: бот + Mini App

В Telegram-боте (быстрые действия)

Главное меню:

1. 🗺️ Открыть карту (Mini App)
2. 📍 Рядом со мной
3. 🍽️ Еда
4. 🛍️ Магазины
5. 🛠️ Услуги
6. 📅 Афиша
7. ⚡ Экстренно / Важно
8. ⭐ Избранное
9. ➕ Добавить место / Исправить данные
10. 💬 Поддержка / Реклама

Бот нужен для "входа", быстрых кнопок и пуша пользователей в Mini App.

В Mini App (основной интерфейс)

- Карта + список (переключатель "Карта / Список")
- Поиск (строка + подсказки)
- Фильтры-чипсы сверху (1 тап):
 - "Открыто сейчас"
 - "24/7"
 - "Доставка"
 - "Безнал"
 - "Парковка"
 - "С детьми"
 - "До 500Р / 1000Р / 2000Р"
 - "Рядом (1 км / 3 км / 5 км)"
- Карточка места: маршрут/позвонить/написать/в избранное/поделиться/сообщить об ошибке

2) Максимальные категории (полный городской справочник)

🍽️ Еда и напитки

- Кафе
- Рестораны
- Кофейни
- Фастфуд
- Пиццерии
- Суши/роллы
- Бургеры
- Шашлык/мангал
- Столовые
- Пекарни/кондитерские
- Бар/паб
- Кальянные
- Доставка еды
- Банкетные залы
- Кейтеринг
- Вегетарианское/ПП
- Завтраки
- "Ночью поесть" (подборка)



Магазины и покупки

- Продукты
- Алкомаркеты (если решите показывать)
- Рынки/ярмарки
- Одежда/обувь
- Детские товары
- Канцтовары/книги
- Электроника/смартфоны
- Компьютеры/сервис
- Бытовая техника
- Мебель/интерьер
- Строительные/ремонт
- Сантехника/электрика
- Товары для дома
- Цветы/подарки
- Косметика/парфюм
- Зоотовары
- Аптеки
- Оптика
- Пункты выдачи (Ozon/WB/Яндекс/СДЭК)
- Спецмагазины: рыбалка/охота, спорттовары



Услуги (самая востребованная категория)

Красота

- Парикмахерские
- Барбершопы
- Маникюр/педикюр
- Брови/ресницы
- Косметология
- Массаж
- SPA/баня/сауна
- Солярий/тату/пирсинг

Ремонт и дом

- Сантехник
- Электрик
- Ремонт квартир
- Мастер на час
- Клининговые услуги
- Химчистка/прачечная
- Дезинфекция/дезинсекция
- Замки/вскрытие (аккуратно в правилах)
- Окна/двери
- Мебель на заказ
- Вывоз мусора/грузчики

Авто

- СТО/автосервис
- Шиномонтаж
- Автомойки
- Автозапчасти
- Автоэлектрик

- Детейлинг
- Эвакуатор
- Автострахование
- Техосмотр

Бизнес-услуги

- Типографии/копицентры
- Фото/печать
- Реклама/полиграфия
- Бухгалтерия/юристы
- Нотариусы
- Переводы
- Курьерские услуги
- Ремонт телефонов/ноутбуков



Здоровье и медицина

- Клиники/медцентры
- Стоматологии
- Анализы/лаборатории
- Диагностика (УЗИ/МРТ/КТ)
- Врачи частные
- Ветклиники
- Аптеки (дублируется, но важно)
- Реабилитация/физиотерапия
- Психологи/психотерапия
- Оптички/линзы



Образование и дети

- Детсады (частные/гос)
- Школы/лицеи
- Репетиторы
- Языковые школы
- Кружки/секции
- Детские центры развития
- Подготовка к ЕГЭ/ОГЭ
- Танцы/музыка/рисование
- Логопеды



Спорт и активный отдых

- Фитнес-клубы
- Тренажёрные залы
- Бассейны
- Йога/пилатес
- Единоборства
- Стадионы/спортплощадки
- Прокат спортивного инвентаря
- Велосипеды/самокаты
- Тир/картинг (если есть)



Досуг и афиша

- Кинотеатры
- Театры
- Концерты

- Музеи/выставки
- Парки/скверы
- Квесты
- Боулинг/бильярд
- Игровые/детские комнаты
- Антикафе/коворкинги
- Клубы/вечеринки
- "Куда сходить сегодня" (автоподборка)



Туризм и размещение

- Отели/гостиницы
- Хостелы
- Апартаменты
- Туры/экскурсии
- Достопримечательности
- Смотровые места
- Инфо для туристов
- Сувениры



Госуслуги и город

- МФЦ
- Администрации/районы
- Налоговая
- Соцзащита/пенсионное
- Почта
- Суд/ФССП (по желанию)
- Паспортные/регистрация (по желанию)
- ЖКХ/управляющие компании (тут лучше "справка + заявки")
- Библиотеки



Экстренно / Важно

- Полиция/дежурные части
- Скорая/травмпункты
- Пожарные/МЧС
- Аварийные службы ЖКХ
- Эвакуатор
- Дежурные аптеки
- "Куда обратиться сейчас" (умная кнопка)



Транспорт и логистика

- Такси (кнопки на приложения/телефоны)
- Автобусы/маршруты (если будете подключать данные)
- Автовокзалы
- ЖД вокзал
- Парковки/стоянки
- Заправки
- Шеринг/прокат
- СДЭК/Почта/курьеры (дублируется, но важно)



Недвижимость (если нужно)

- Аренда/покупка
- Агентства
- Новостройки (инфоблоки)

- Оценка/ипотека (партнёрки)

3) "Фишки", которые реально заставят пользоваться вашим ботом

Вот набор УТП, который карты не дают так удобно:

1) "Ситуационные ответы", а не просто поиск

Кнопки-решения:

- «Где поесть рядом *прямо сейчас*»
- «Аптека дежурная/ночная»
- «Шиномонтаж 24/7»
- «Кофе с собой до 150Р»
- «Где с детьми (детская зона)»

Это экономит время: люди приходят за готовым вариантом, а не за списком.

2) Проверка актуальности "живыми" метками

Вы вводите статусы:

-  **Проверено** (адрес/телефон/время подтверждены недавно)
-  "Обновлено X дней назад"
-  "Есть жалобы на неверный график"

Карты часто содержат "мертвые" точки и пользователю непонятно, что реально работает.

3) "Действие в 1 тап"

В карточке места кнопки:

- Позвонить
- Написать (WhatsApp/Telegram)
- Построить маршрут
- Забронировать / Записаться (если подключите партнёров)
- Купон/скидка

Люди пользуются тем, где меньше шагов.

4) Подборки, которые хочется открывать каждый день

- "Топ недели"
- "Новые места"
- "Лучшее в районе"
- "События сегодня"
- "Где выгодно" (скидки/комбо)

У карт это не так "по-человечески" собрано.

5) Локальная "городская логика"

Районы/местные названия, понятные жителям: бот "говорит языком города".

Плюс категории, которые именно в быту нужны: "ремонт телефонов", "копицентры", "грузчики", "эвакуатор".

6) Избранное + "мои места"

Пользователь быстро собирает:

- любимые кофе/аптеки/СТО
- места "рядом с домом/работой"

И возвращается, потому что это "его личный справочник".

7) Добавление мест и правки за 30 секунд

Кнопка "исправить" прямо в карточке → вы модерите.

Это постепенно делает ваш бот **самым актуальным**.

4) Супер-быстрый сценарий (то, что надо реализовать первым)

MVP, который уже "убивает" обычные карты удобством:

- Mini App: карта + список + поиск
- "Рядом со мной" + "Открыто сейчас"
- 12 главных категорий (как выше)
- Карточка места: маршрут/позвонить/избранное/поделиться
- Подборки: "Топ", "24/7", "С детьми", "Доставка"
- "Сообщить об ошибке" + "Добавить место"

5) Как это подать в описании бота (готовый текст)

"Ставрополь в одном боте: места рядом, открыто сейчас, подборки по районам, быстрые фильтры и избранное. Проверяем актуальность и добавляем новые места по заявкам."

Если хочешь, я могу прямо следующим сообщением дать:

- **точное меню кнопок + сценарии диалогов** (что пользователь нажимает и что видит),
- **структуру фильтров и тегов** (чтобы выдача была умной),
- **и шаблоны карточек** (какой текст/кнопки/поля).

Создай готового бота, что я загрузил себе на хостинг

Думал на протяжении 14m 54s >

Готово: я собрал **полный рабочий шаблон** "городской справочник Ставрополя" как **Telegram-бот + Telegram Mini App** (карта + фильтры внутри), чтобы ты мог просто залить на хостинг и запустить.

[Скачать проект \(zip\)](#)

Что внутри и как это работает

1) Telegram Bot

- Команда `/start`
- Кнопка «Открыть карту» → открывает Mini App прямо в Telegram
- Кнопка «Отправить локацию» → открывает карту "рядом со мной"

2) Mini App (в Telegram)

- Карта (Яндекс JS API v3) + список результатов
- Поиск по тексту + фильтр по категориям
- "Рядом со мной"
- "Открыто сейчас" (в MVP сделано через добавление фразы в запрос; дальше можно улучшить логикой по Hours)

3) Backend (FastAPI)

- Отдаёт Mini App (html/js/css)
- `/api/search` — проксирует запросы в **Yandex Organization Search / Places API**
- **Поиск “на лету”** + небольшой кэш на 5 минут для скорости (без создания своей базы)

Yandex endpoint и параметры запроса (`text/lang/type=biz/ll/spn/results` и т.д.) — официальная схема.

[yandex.ru](#) +1

Проверка подписи Telegram Mini App (`initData`) реализована на сервере (можно включить/выключить). [GitHub](#) +1

Подключение Яндекс JS API v3 (`script v3/?apikey=...`) — по официальной документации. [Yandex](#) +1

Как запустить у себя (самый простой вариант)

1. Разархивируй проект
2. Открой файл `.env.example`, скопируй в `.env`
3. Заполни:
 - `BOT_TOKEN` — токен от @BotFather
 - `PUBLIC_BASE_URL` — твой домен (например `https://stavropolbot.ru`)
 - `YANDEX_MAPS_JS_API_KEY` — ключ JS карты
 - `YANDEX_ORGSEARCH_API_KEY` — ключ поиска организаций
4. В @BotFather обязательно:
 - `/setdomain` → укажи домен `PUBLIC_BASE_URL` (иначе Mini App не откроется)
5. Запуск одной командой:

```
bash
```

 Копировать код

```
docker compose up -d --build
```

Проверка:

- `https://ТВОЙ_ДОМЕН/` — откроется Mini App
- `https://ТВОЙ_ДОМЕН/api/health` — должно вернуть `{"ok":true}`

“Максимальные категории” (что уже заложено)

В проекте есть крупное дерево категорий (правится в `api/categories.py`) по группам:

- **Еда** (кафе, рестораны, доставка, бары, пекарни, столовые...)
- **Магазины** (продукты, одежда, электроника, аптеки, зоотовары, рынки, пункты выдачи...)
- **Услуги** (красота, дом/ремонт/клининг, грузчики, мастера...)
- **Авто** (СТО, мойки, шиномонтаж, эвакуатор, АЗС...)
- **Бизнес** (юристы, нотариусы, копицентры, ремонты техники...)
- **Здоровье** (клиники, стоматологии, анализы, диагностика, психологи, вет...)
- **Дети/обучение**
- **Спорт**
- **Досуг**
- **Туризм**
- **Город/госуслуги**
- **Экстренное**
- **Транспорт/логистика**

(Это уже “широкий максимум” для MVP. Можно ещё расширять под Ставрополь: районы, микрорайоны, локальные теги и подборки.)

Почему твоим ботом будут пользоваться, а не 2ГИС/Яндекс

Вот **реальные "фишки"**, ради которых люди открывают именно **TG-бот**, даже если карты уже есть:

1. **Всё в одном месте, без установки приложений**
 - Telegram уже стоит у всех → "открыл и нашёл".
2. **Супер-быстрый сценарий "мне надо сейчас"**
 - "Рядом со мной → еда → открыто сейчас" в 2–3 клика.
 - Без лишних экранов, рекламы и "потери" пользователя.
3. **Подборки и локальная польза (то, чего нет в картах)**
 - "Где поесть после 23:00"
 - "Лучшие шашлыки Ставрополя"
 - "Пункты выдачи рядом"
 - "Куда сходить с детьми сегодня"
 - Это можно делать как "витрины" внутри Mini App + пуши в TG.
4. **Сервисные действия прямо в боте**
 - "Позвонить"
 - "Построить маршрут"
 - "Добавить в избранное"
 - "Поделиться точкой другу"
 - Дальше можно добавить: бронирование, купоны, запись, заявки.
5. **Всегда актуально (поиск на лету)**
 - Ты не ведёшь базу вручную → данные подтягиваются актуальные.
 - Это ровно твоя идея "искать на лету".

Важно про стоимость/лицензию API (чтобы не влететь)

По **Yandex Organization Search API** официально есть только базовая лицензия, и условия запрещают сохранять/модифицировать данные в виде своей базы (поэтому в проекте сделан подход "на лету + короткий кэш"). Yandex

Что дальше могу быстро добавить в этот проект (самое полезное для монетизации)

- **Избранное** (уже подготовлен экран "panel=favs", можно подключить БД)
- **Персональные подборки по районам Ставрополя**
- **Купоны/акции** (платное размещение)
- **Партнёрские карточки "первым в списке"**
- **Модерация/сообщить об ошибке** (кнопка → заявка в админку)

Если хочешь — скажи, на чём будешь хостить (VPS/Timeweb/Render/etc.) и какой домен планируешь: я подскажу оптимальную схему деплоя (HTTPS, nginx, webhook vs polling) без усложнений.

Timeweb Домен: cn79488-wordpress-5e13e.tw1.ru

Думал на протяжении 51s >

Ок, под **Timeweb** (и твой домен `cn79488-wordpress-5e13e.tw1.ru`) самый надёжный способ — **VPS/VDS + Docker + Nginx + Let's Encrypt**. На "обычном" WordPress-хостинге фоновые процессы (бот) обычно **нельзя держать постоянно**, поэтому ниже даю рабочую схему именно для VPS.

1) Подготовь VPS в Timeweb

1. В панели Timeweb создай **VDS/VPS** (Ubuntu 22.04/24.04).
2. Узнай **IP** сервера.
3. В DNS домена `cn79488-wordpress-5e13e.tw1.ru` поставь:
 - **A-запись** → на IP твоего VPS

Если этот домен сейчас занят WordPress-хостингом и ты не хочешь его “переносить”, то сделай отдельный **поддомен** (например `guide.cn79488-wordpress-5e13e.tw1.ru`) и направь уже его на VPS.

2) Подключись по SSH

```
bash
```

[Копировать код](#)

```
ssh root@IP_ТВОЕГО_VPS
```

3) Установи Docker + Compose

Ubuntu:

```
bash
```

[Копировать код](#)

```
apt update && apt -y install ca-certificates curl gnupg unzip  
curl -fsSL https://get.docker.com | sh  
apt -y install docker-compose-plugin
```

Проверка:

```
bash
```

[Копировать код](#)

```
docker --version  
docker compose version
```

4) Залей проект на сервер и распакуй

Вариант А (с компьютера через scp)

На своём ПК:

```
bash
```

[Копировать код](#)

```
scp stavropol-guide-bot.zip root@IP_ТВОЕГО_VPS:/root/
```

На сервере:

```
bash
```

[Копировать код](#)

```
mkdir -p /opt/stavropol-guide-bot  
unzip /root/stavropol-guide-bot.zip -d /opt/  
cd /opt/stavropol-guide-bot/stavropol-guide-bot
```

(Можно удалить `__pycache__`, они не нужны)

```
bash
```

[Копировать код](#)

```
rm -rf bot/__pycache__ api/__pycache__ || true
```

5) Настрой .env

```
bash
```

[Копировать код](#)

```
cp .env.example .env  
nano .env
```

Заполни так:

- BOT_TOKEN=... (из @BotFather)
- PUBLIC_BASE_URL=https://cn79488-wordpress-5e13e.tw1.ru
- YANDEX_MAPS_JS_API_KEY=...
- YANDEX_ORGSEARCH_API_KEY=...
- VERIFY_INITDATA=false (пока можно так; потом включим)

6) Подними проект Docker'ом

```
bash
```

[Копировать код](#)

```
docker compose up -d --build
docker compose ps
```

Проверка локально на сервере:

```
bash
```

[Копировать код](#)

```
curl http://127.0.0.1:8080/api/health
```

Должно вернуть {"ok":true}.

7) Поставь Nginx + HTTPS (обязательно для Mini App)

```
bash
```

[Копировать код](#)

```
apt -y install nginx certbot python3-certbot-nginx
```

Создай конфиг:

```
bash
```

[Копировать код](#)

```
nano /etc/nginx/sites-available/stavropol-guide
```

Вставь (замени домен, если используешь поддомен):

```
nginx
```

[Копировать код](#)

```
server {
    server_name cn79488-wordpress-5e13e.tw1.ru;

    location / {
        proxy_pass http://127.0.0.1:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Включи сайт и перезапусти:

```
bash
```

[Копировать код](#)

```
ln -sf /etc/nginx/sites-available/stavropol-guide /etc/nginx/sites-enabled/stavropol-gu
nginx -t && systemctl reload nginx
```

Выпусти SSL:

```
bash
```

[Копировать код](#)

```
certbot --nginx -d cn79488-wordpress-5e13e.tw1.ru
```

Проверь из браузера:

- `https://cn79488-wordpress-5e13e.tw1.ru/api/health`

8) Настрой BotFather под Mini App

В @BotFather:

- `/setdomain` → укажи: `cn79488-wordpress-5e13e.tw1.ru`
(или твой поддомен, если выбрал его)

После этого:

- зайдя в бота → `/start` → “Открыть карту” — Mini App должен открыться.

9) (Рекомендуется) Закрывать наружу порт 8080

Чтобы снаружи торчали только 80/443:

```
bash
```

 Копировать код

```
apt -y install ufw
ufw allow 22
ufw allow 80
ufw allow 443
ufw --force enable
```

Если у тебя сейчас только WordPress-хостинг (без VPS)

Тогда бот нормально “24/7” не запустить — нужен **VPS** (или внешний PaaS типа Render/Fly.io и т.п.).

Сам Mini App (статический сайт) можно теоретически положить на хостинг, но **боту всё равно** нужен постоянно работающий процесс.

Напиши 2 вещи — и я дам точные настройки без “вилами по воде”

1. У тебя уже есть **VPS на Timeweb** или пока только WordPress-хостинг?
2. Ты хочешь, чтобы бот работал на **этом домене** (заменив WordPress) или лучше сделать **поддомен** (например `guide.cn79488-wordpress-5e13e.tw1.ru`), а WordPress оставить как есть?

Timeweb хостинг

Понял: **Timeweb хостинг (обычный, не VPS)**. На таком хостинге **нельзя нормально держать Telegram-бота 24/7** (фоновые процессы/демоны запрещены или убиваются), поэтому “просто залить и он будет работать” — обычно не получится.

Но решение есть — и оно простое. Дам 3 варианта, от самого правильного к компромиссным.

Вариант 1 (рекомендую): VPS/VDS на Timeweb (самый правильный)

Почему: бот — это постоянно работающий процесс. На VPS можно:

- запустить Docker/PM2/systemd

- поставить HTTPS
- держать стабильный webhook
- всё будет работать всегда

Что нужно: минимальный VPS (дешёвый) + домен/поддомен.

На твоём хостинге можно оставить WordPress, а бота поставить на поддомен:

- `bot.cn79488-wordpress-5e13e.tw1.ru` или `guide...`

Вариант 2: Хостинг + внешний сервер для бота (бюджетно)

Если VPS не хочешь, можно:

- **Mini App (сайт)** держать на Timeweb-хостинге (как обычные файлы)
- **Backend + Telegram bot** держать на любом PaaS (Render/Fly.io/Railway) или другом VPS

Так тоже нормально: пользователи открывают Mini App с Timeweb, а запросы идут на внешний API.

Вариант 3 (компромисс): запуск бота через cron на хостинге

Можно пытаться запускать "лонг-поллинг" бота через cron каждые N минут, но:

- бот будет постоянно падать/залипать
- задержки
- webhook нормально не поднять
- Mini App + API будет страдать

Я не советую, если хочешь "быстрый справочник".

Что можно сделать прямо сейчас на Timeweb-хостинге (без VPS)

✅ **Mini App (фронт)** можно разместить на хостинге как сайт.

❌ **Бот + API** — нужно где-то, где можно держать процесс постоянно.

Самый удобный путь для тебя (без потери WordPress)

1. Берём **поддомен** для бота: `guide.cn79488-wordpress-5e13e.tw1.ru`
2. На Timeweb хостинге оставляем WordPress как есть
3. Покупаем **минимальный VPS в Timeweb** только под бота (это реально самый беспроblemный вариант)
4. Настраиваем HTTPS + webhook

Скажи одно — и я дам точные шаги под твой случай

Ты готов:

A) подключить **VPS в Timeweb** (лучше всего),

или

B) поставить бота на **внешний сервис (Render/Fly/Railway)**, а Mini App оставить на хостинге?

Если выберешь A или B — я дам пошаговую инструкцию "нажми туда → вставь сюда", без лишней теории.

