Ниже — полное ТЗ для разработки «Царь такси». Ключевые допущения зафиксированы: Сезон = календарный месяц. Каждый новый месяц — новый сезон. Администратор на каждый сезон задаёт дневную цель; месячная цель рассчитывается как дневная × число дней месяца. Роль «Король» переименована в «Царь». Название мотивации: «Царь такси».

1. Краткое описание модели и терминов

1.1. Назначение

- «Царь такси» — ежемесячная мотивационная программа для водителей такси. Участники нарабатывают часы работы на линии. Итоги каждого сезона (месяца) формируют рейтинги и смену ролей.

1.2. Сезон

- Сезон равен календарному месяцу (1–последний день месяца).

- Каждый новый месяц — новый сезон.

1.3. Роли и иерархия

- Царь — 1 участник.

- Сотники — до 10 участников (каждый ведёт до 10 Десятников).

- Десятники — до 100 участников (каждый ведёт до 10 Водителей).

- Водители — до 1000 участников.

- Идеальная полная структура: 1 + 10 + 100 + 1000 = 1111.

1.4. Идентификация пользователей

- Ключ — номер телефона (E.164).

- В Telegram бот привязывает пользователя по отправленному контакту.

1.5. Импорт фактов

- Ежедневно администратор импортирует XLSX: колонка «телефон», далее колонки дат (Excel serial или ISO). В ячейках — часы (целые/дробные).

1.6. Метрики результата

- Личный результат (персональные часы).

- Командный результат (сумма результатов подчинённых).

- Итог = личные + командные.

1.7. Цели

- На каждый сезон администратор задаёт «дневную цель» (часов/день для одной единицы — водителя).

- Цель сезона для единицы = дневная цель × число календарных дней сезона.

- Цели ролей масштабируются по фактическому числу «единиц» в их иерархии (см. раздел 2.2).

1.8. Ранжирование и переходы ролей

- Ранги в группах считаются по итогам сезона. Переходы ролей — по фиксированным правилам (см. раздел 2.5).

2. Подробная логика расчётов, целей, ролей и переходов

2.1. Параметры сезона

- Администратор задаёт:

- Название сезона (например, «Февраль 2025»), даты (1–последний день месяца).

- Список участников (телефоны).

- Дневную цель часов (например, 6.7 ч/день).

- Система рассчитывает:

- Количество дней месяца (28–31).

- Цель сезона для одной единицы: дневная × дни месяца.

2.2. Цели ролей (масштабирование)

- Водитель: цель = 1 × цель сезона единицы.

- Десятник: цель = (1 + число подчинённых водителей) × цель сезона единицы.

- Сотник: цель = (1 + Σ по его десятникам (1 + число их водителей)) × цель сезона единицы.

- Царь: цель = (1 + Σ по сотникам (1 + Σ по десятникам (1 + водители))) × цель сезона единицы.

- Если группы неполные, цели автоматически уменьшаются пропорционально фактическому составу.

2.3. Результаты по дням и сезону

- Водитель: личные часы из импорта по дате; сезон = сумма по дням.

- Десятник (за день): персональные часы + сумма часов всех его водителей за день; сезон = сумма по дням.

- Сотник (за день): персональные часы + сумма итогов всех его десятников за день; сезон = сумма по дням.

- Царь (за день): персональные часы + сумма итогов всех сотников за день; сезон = сумма по дням.

- % выполнения цели = сезонный итог / сезонная цель.

2.4. Ранжирование

- Водители внутри группы Десятника: по сезонному итогу, при равенстве — больше личные часы → раньше достигнутый % цели → случайно.

- Десятники внутри группы Сотника: по сезонному итогу; тай-брейки как выше.

- Сотники: общий список по сезонному итогу; тай-брейки как выше.

2.5. Переходы ролей между сезонами

- Царь и Сотники:

- Лучший Сотник (место 1) становится новым Царём.

- Действующий Царь понижается до Сотника.

- Сотники мест 2–5 — сохраняют роль Сотника.

- Сотники мест 6–10 — понижаются до Десятников.

- Освободившиеся места Сотников занимают «лучшие 5 Десятников среди первых мест Десятников» (см. ниже).

- Десятники:

- У каждого Сотника есть «первое место Десятника» (1-й в его группе). Собрать все эти «первые места» и отсортировать по итогам сезона.

- Топ-5 повышаются до Сотников (чтобы закрыть места выбывших).

- Места 6–10 среди «первых мест» остаются Десятниками.

- Все остальные Десятники (кто не «первое место») — понижаются до Водителей.

- Водители:

- У каждого Десятника есть «первое место Водителя» (1-й в его группе). Собрать все «первые места».

- Отсортировать; лучшие 90 повышаются до Десятников.

- Остальные Водители остаются Водителями.

- Масштабирование: если фактическое число Сотников/Десятников меньше идеала, пропорционально уменьшать количество повышений/понижений (например, если Сотников 6, выбывают места 6, а повышаем соответствующее количество «первых мест» Десятников).

2.6. Первый сезон (первый месяц программы)

- Все участники стартуют Водителями.

- По итогам:

- 1 место — Царь.

- 2–11 места — 10 Сотников.

- 12–111 места — 100 Десятников.

- Остальные — Водители.

2.7. Перемешивание групп каждый сезон

- После завершения переходов ролей:

- Случайно распределить Десятников по Сотникам равномерно (разница по числу ≤ 1).

- Случайно распределить Водителей по Десятникам равномерно (разница ≤ 1).

- Цель — уравнять шансы, снизить риск договорённостей.

2.8. Новые пользователи

- Подают заявку в лист ожидания.

- Вступают со следующего сезона в роли Водителя.

- При лимите общей численности замещают худших Водителей прошлого сезона. Без лимита — добавляются сверх, с равномерной ротацией.

2.9. Антифрод

- Флаги аномалий: дневные часы > конфигурируемого порога (напр. > 16 ч), резкие всплески относительно медианы.

- Отчёт админу, ручная проверка.

- Ежемесячная ротация групп.

3. Техническое ТЗ для разработчиков

3.1. Архитектура

- Компоненты:

- Telegram-бот (webhook).

- Бэкенд API (может быть объединён с ботом): аутентификация, импорт, расчёты, переходы ролей, распределения, уведомления, админ-команды.

- БД (PostgreSQL).

- Планировщик/очередь заданий (cron/worker).

- Файловое хранилище для импортов (S3/локально).

- Требования:

- Производительность: 2–5 тыс. пользователей, импорты до 50 тыс. строк/день; пересчёт < 10 минут.

- Доступность 99%, ежедневные бэкапы БД.

- Безопасность: разграничение ролей (user/admin), хранение ПДн, протоколирование админ-операций.

3.2. База данных

- Схема описана в разделе 7 (SQL).

3.3. Алгоритмы расчёта

- После каждого импорта:

- Пересчитать aggregates\_daily (персональные/командные/итог по ролям).

- Пересчитать aggregates\_season (сумма по дням, цели с учётом фактической структуры, % выполнения).

- Обновить ранги (внутри групп и среди Сотников).

- В конце сезона:

- Зафиксировать итоги.

- Применить переходы ролей (см. 2.5).

- Выполнить ротацию групп (см. 2.7).

- Инициализировать роли и цели нового сезона.

3.4. API/бот

- Авторизация:

- Пользователь: привязка телефона через sendContact в боте. Бэкенд сопоставляет с users.phone.

- Админ: отдельный токен X-Admin-Key для админ-эндпойнтов.

- Команды бота — см. раздел 6 (UX).

- REST API — см. раздел 7 (OpenAPI).

3.5. Импорты

- Формат: XLSX, колонка «phone», далее — колонки дат (Excel serial или ISO). Значения — часы (>=0).

- Обработка:

- Валидация телефонов и дат.

- UPSERT по (user\_id, work\_date).

- Лог ошибок (imports.errors\_json).

- Флоу:

- Админ загружает → система парсит → пишет в hours\_raw → запускает пересчёт.

3.6. Задачи по расписанию

- Ежедневно 01:00 — пересчёт агрегатов за прошедший день.

- Ежедневно 10:00 — рассылка ежедневных уведомлений.

- В последний день сезона после 23:59:

- Финальный пересчёт.

- Применение переходов ролей.

- Ротация групп.

- Подготовка следующего сезона с нулевыми данными.

3.7. Уведомления

- Пользовательские ежедневные уведомления — см. раздел 6.2.

- Системные:

- О пропусках часов > N дней.

- О смене роли в конце сезона.

- О принятии заявки в участие.

4. Публичные правила мотивационной программы (для пользователей)

- «Царь такси» — ежемесячные сезоны. На сезон задаётся дневная цель, месячная цель считается по числу дней.

- Ваши результаты:

- Водитель — личные часы.

- Десятник — личные + сумма часов водителей.

- Сотник — личные + результаты его десятников.

- Царь — личные + результаты всех сотников.

- Роли меняются по результатам сезона согласно правилам раздела 2.5.

- Группы перемешиваются каждый сезон.

- Новые участники вступают со следующего сезона в роли Водителя.

- Нельзя искажать данные или передавать аккаунт. За нарушения — блокировка.

- Программа не является трудовыми отношениями и не гарантирует призы (если не оговорено отдельно).

- Организатор может менять правила, информируя участников заранее.

5. Юридическая часть: согласие на обработку ПДн и комплаенс

5.1. Согласие на обработку ПДн (шаблон)

- Я, ФИО, даю согласие [Оператору, адрес, ИНН/ОГРН] на обработку моих персональных данных для участия в программе «Царь такси»: номер телефона, Telegram ID/username, факты часов работы по дням, показатели выполнения, переписка в боте, технические метаданные. Цели: расчёт результатов и рейтингов, уведомления, коммуникации, администрирование участия, анализ и улучшение программы.

- Операции: сбор, запись, систематизация, хранение, уточнение, извлечение, использование, передача уполномоченным лицам оператора и подрядчикам по договорам, обезличивание, блокирование, удаление, уничтожение.

- Срок обработки: на период участия и 12 месяцев после окончания, либо иные сроки по закону.

- Право отзыва согласия по [контактам оператора]; ознакомлен(а) с Политикой обработки ПДн [ссылка].

- Разрешаю трансграничную передачу ПДн (если сервисы размещены за пределами РФ).

5.2. Политика обработки ПДн (кратко)

- Какие данные собираем, цели, основания (согласие), доступ ограничен уполномоченными; сроки хранения; права субъекта ПДн; контакты оператора/DPO.

5.3. Пользовательское соглашение (тезисы)

- Условия участия, запреты на фрод, санкции (блокировка/исключение), отсутствие гарантий наград (если не объявлены), право на изменение правил.

6. Макеты меню бота (UX), тексты ежедневных уведомлений

6.1. Онбординг

- /start:

- «Добро пожаловать в Царь такси. Для участия требуется согласие на обработку персональных данных и привязка номера телефона.»

- Кнопки: «Принять условия», «Политика конфиденциальности».

- После согласия — запрос контакта (отправить номер).

- Ответы:

- «Вы подключены. Текущая роль: <роль>.»

- «Вас нет в текущем сезоне. Подать заявку на следующий сезон?» (кнопка «Подать заявку») → «Заявка принята.»

6.2. Главное меню (ReplyKeyboard)

- Моя роль

- Моя статистика

- Моя команда

- Моё место в группе

- Мои цели

- Контакты

- Помощь

- Политика

- Выйти из программы

6.3. Ответы на команды

- Моя роль: «Ваша роль: <Царь/Сотник/Десятник/Водитель>. Руководитель: <имя> (если есть). Подчинённые: <N> (если есть).»

- Моя статистика:

- Водитель: «Сегодня: X ч. Сезон: A/B (C%). Прогноз: D%.»

- Десятник: «Лично: X. Команда: Y. Сезон: A/B (C%). Прогноз: D%.»

- Сотник: «Лично: X. Команда: Y. Сезон: A/B (C%). Прогноз: D%.»

- Царь: «Команда: Y. Сезон: A/B (C%). Прогноз: D%.»

- Моя команда:

- Десятник: топ-5 водителей с часами и %.

- Сотник: топ-5 десятников.

- Царь: топ-5 сотников.

- Моё место в группе:

- Водитель: «Ваше место среди водителей у Десятника <имя>: N/M.»

- Десятник: «Ваше место среди десятников у Сотника <имя>: N/M.»

- Сотник: «Ваше место среди сотников: N/T.»

- Мои цели: «Дневная цель: x ч/день. Цель сезона (единица): B ч. Ваша цель с учётом структуры: T ч.»

- Контакты: руководитель + подчинённые + кнопки «Написать».

- Помощь: краткая справка по ролям, расчётам и переходам.

- Политика: краткий текст + ссылка.

- Выйти из программы: подтверждение, выход с нового сезона.

6.4. Переписка по иерархии

- Водитель ↔ Десятник, Десятник ↔ Сотник, Сотник ↔ Царь (можно отключить админом).

6.5. Админ-меню

- Загрузить XLSX с часами

- Создать/активировать/закрыть сезон

- Пересчитать агрегаты

- Ротация групп

- Управление пользователями

- Лист ожидания

- Вкл/выкл переписку

6.6. Ежедневные уведомления (шаблоны)

- Водитель: «Царь такси: День <дата>. Лично сегодня: <X> ч. За сезон: <A>/<B> ч (<C>%). Место в группе: <N>/<M>. Прогноз: <D>% к концу.»

- Десятник: «Царь такси: День <дата>. Лично: <X> ч. Команда: <Y> ч. Сезон: <A>/<T> ч (<C>%). Топ-3 водителей: 1) <Имя1> <h1> ч; 2) <Имя2> <h2> ч; 3) <Имя3> <h3> ч.»

- Сотник: «Царь такси: День <дата>. Лично: <X> ч. Команда: <Y> ч. Сезон: <A>/<T> ч (<C>%). Топ-3 десятников: …»

- Царь: «Царь такси: День <дата>. Команда: <Y> ч. Сезон: <A>/<T> ч (<C>%). Топ-3 сотников: …»

- Напоминание о простое: «Вы не на линии уже <n> дней. Дневная цель: <x> ч.»

- Закрытие сезона: «Сезон завершён. Ваш итог: <A>/<T> (<C>%). Ваша новая роль с нового сезона: <роль>.»

7. SQL-схема БД и OpenAPI спека

7.1. SQL-схема (PostgreSQL)

CREATE TABLE users (

id BIGSERIAL PRIMARY KEY,

phone VARCHAR(20) UNIQUE NOT NULL,

tg\_user\_id BIGINT,

full\_name VARCHAR(255),

status VARCHAR(20) NOT NULL DEFAULT 'active', -- active, blocked, waiting

consent\_given\_at TIMESTAMP NULL,

created\_at TIMESTAMP NOT NULL DEFAULT NOW(),

updated\_at TIMESTAMP NOT NULL DEFAULT NOW()

);

CREATE TABLE seasons (

id BIGSERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

start\_date DATE NOT NULL,

end\_date DATE NOT NULL,

days\_count INT NOT NULL,

daily\_target\_hours NUMERIC(6,2) NOT NULL,

monthly\_unit\_target NUMERIC(8,2) NOT NULL,

status VARCHAR(20) NOT NULL DEFAULT 'planned' -- planned, active, closed

);

CREATE TABLE role\_assignments (

id BIGSERIAL PRIMARY KEY,

season\_id BIGINT NOT NULL REFERENCES seasons(id) ON DELETE CASCADE,

user\_id BIGINT NOT NULL REFERENCES users(id) ON DELETE CASCADE,

role VARCHAR(20) NOT NULL, -- tsar, sotnik, desyatnik, driver

sotnik\_id BIGINT NULL REFERENCES users(id), -- руководитель для десятников/водителей

desyatnik\_id BIGINT NULL REFERENCES users(id), -- руководитель для водителей

group\_index INT,

UNIQUE (season\_id, user\_id)

);

CREATE INDEX idx\_role\_assignments\_role ON role\_assignments(role);

CREATE TABLE hours\_raw (

id BIGSERIAL PRIMARY KEY,

user\_id BIGINT NOT NULL REFERENCES users(id) ON DELETE CASCADE,

work\_date DATE NOT NULL,

hours NUMERIC(5,2) NOT NULL CHECK (hours >= 0),

import\_id BIGINT,

UNIQUE (user\_id, work\_date)

);

CREATE INDEX idx\_hours\_raw\_date ON hours\_raw(work\_date);

CREATE TABLE aggregates\_daily (

id BIGSERIAL PRIMARY KEY,

user\_id BIGINT NOT NULL REFERENCES users(id) ON DELETE CASCADE,

season\_id BIGINT NOT NULL REFERENCES seasons(id) ON DELETE CASCADE,

work\_date DATE NOT NULL,

role VARCHAR(20) NOT NULL,

personal\_hours NUMERIC(8,2) NOT NULL DEFAULT 0,

team\_hours NUMERIC(10,2) NOT NULL DEFAULT 0,

total\_hours NUMERIC(10,2) NOT NULL DEFAULT 0,

UNIQUE (user\_id, season\_id, work\_date)

);

CREATE INDEX idx\_agg\_daily\_user\_date ON aggregates\_daily(user\_id, work\_date);

CREATE TABLE aggregates\_season (

id BIGSERIAL PRIMARY KEY,

user\_id BIGINT NOT NULL REFERENCES users(id) ON DELETE CASCADE,

season\_id BIGINT NOT NULL REFERENCES seasons(id) ON DELETE CASCADE,

role VARCHAR(20) NOT NULL,

personal\_total NUMERIC(10,2) NOT NULL DEFAULT 0,

team\_total NUMERIC(12,2) NOT NULL DEFAULT 0,

total NUMERIC(12,2) NOT NULL DEFAULT 0,

target NUMERIC(12,2) NOT NULL DEFAULT 0,

target\_percent NUMERIC(6,3) NOT NULL DEFAULT 0,

rank\_in\_group INT,

sotnik\_rank INT,

UNIQUE (user\_id, season\_id)

);

CREATE INDEX idx\_agg\_season\_role ON aggregates\_season(role);

CREATE TABLE imports (

id BIGSERIAL PRIMARY KEY,

file\_name VARCHAR(255) NOT NULL,

uploaded\_by BIGINT REFERENCES users(id),

uploaded\_at TIMESTAMP NOT NULL DEFAULT NOW(),

rows\_count INT NOT NULL DEFAULT 0,

status VARCHAR(20) NOT NULL DEFAULT 'processed', -- processed, failed

errors\_json TEXT

);

CREATE TABLE waitlist (

id BIGSERIAL PRIMARY KEY,

phone VARCHAR(20) NOT NULL,

full\_name VARCHAR(255),

added\_at TIMESTAMP NOT NULL DEFAULT NOW(),

status VARCHAR(20) NOT NULL DEFAULT 'new' -- new, approved, rejected

);

CREATE TABLE messaging\_permissions (

id BIGSERIAL PRIMARY KEY,

leader\_id BIGINT NOT NULL REFERENCES users(id) ON DELETE CASCADE,

member\_id BIGINT NOT NULL REFERENCES users(id) ON DELETE CASCADE,

direction VARCHAR(30) NOT NULL, -- leader\_to\_member, member\_to\_leader

allowed BOOLEAN NOT NULL DEFAULT TRUE,

UNIQUE (leader\_id, member\_id, direction)

);

CREATE TABLE audit\_logs (

id BIGSERIAL PRIMARY KEY,

actor\_id BIGINT REFERENCES users(id),

action VARCHAR(100) NOT NULL,

entity\_type VARCHAR(50),

entity\_id BIGINT,

payload\_json TEXT,

created\_at TIMESTAMP NOT NULL DEFAULT NOW()

);

7.2. OpenAPI 3.0 (минимальная спека)

openapi: 3.0.3

info:

title: Царь такси API

version: 1.0.0

servers:

- url: https://api.example.com

paths:

/auth/telegram/webhook:

post:

summary: Webhook для Telegram-бота

operationId: botWebhook

responses:

'200': { description: ok }

/me:

get:

summary: Профиль текущего пользователя

security: [{ bearerAuth: [] }]

responses:

'200':

description: OK

content:

application/json:

schema: { $ref: '#/components/schemas/UserProfile' }

/me/stats:

get:

summary: Статистика пользователя за сезон

security: [{ bearerAuth: [] }]

responses:

'200':

description: OK

content:

application/json:

schema: { $ref: '#/components/schemas/MyStats' }

/me/team:

get:

summary: Командная статистика (для Десятника/Сотника/Царя)

security: [{ bearerAuth: [] }]

responses:

'200': { description: OK }

/me/rank:

get:

summary: Моё место в группе

security: [{ bearerAuth: [] }]

responses:

'200': { description: OK }

/me/contacts:

get:

summary: Контакты руководителя и подчинённых

security: [{ bearerAuth: [] }]

responses:

'200': { description: OK }

/admin/seasons:

post:

summary: Создать новый сезон (месяц)

security: [{ adminKey: [] }]

requestBody:

required: true

content:

application/json:

schema: { $ref: '#/components/schemas/CreateSeason' }

responses:

'201': { description: Created }

/admin/seasons/{id}/activate:

post:

summary: Активировать сезон

security: [{ adminKey: [] }]

parameters:

- name: id

in: path

required: true

schema: { type: integer }

responses:

'200': { description: OK }

/admin/seasons/{id}/close:

post:

summary: Закрыть сезон, применить переходы ролей, ротацию

security: [{ adminKey: [] }]

parameters:

- name: id

in: path

required: true

schema: { type: integer }

responses:

'200': { description: OK }

/admin/imports:

post:

summary: Загрузка XLSX с часами

security: [{ adminKey: [] }]

requestBody:

required: true

content:

multipart/form-data:

schema:

type: object

properties:

file: { type: string, format: binary }

responses:

'200': { description: Processed }

/admin/recalculate:

post:

summary: Пересчёт агрегатов

security: [{ adminKey: [] }]

responses:

'200': { description: OK }

/admin/redistribute:

post:

summary: Случайная ротация групп

security: [{ adminKey: [] }]

responses:

'200': { description: OK }

components:

securitySchemes:

bearerAuth:

type: http

scheme: bearer

adminKey:

type: apiKey

in: header

name: X-Admin-Key

schemas:

CreateSeason:

type: object

required: [name, start\_date, end\_date, daily\_target\_hours]

properties:

name: { type: string }

start\_date: { type: string, format: date }

end\_date: { type: string, format: date }

daily\_target\_hours: { type: number }

UserProfile:

type: object

properties:

id: { type: integer }

phone: { type: string }

full\_name: { type: string }

role: { type: string, enum: [tsar, sotnik, desyatnik, driver] }

leader:

type: object

properties:

id: { type: integer }

full\_name: { type: string }

role: { type: string }

season:

type: object

properties:

id: { type: integer }

name: { type: string }

start\_date: { type: string, format: date }

end\_date: { type: string, format: date }

daily\_target\_hours: { type: number }

monthly\_unit\_target: { type: number }

MyStats:

type: object

properties:

role: { type: string }

personal\_today: { type: number }

team\_today: { type: number }

total\_today: { type: number }

personal\_season: { type: number }

team\_season: { type: number }

total\_season: { type: number }

target\_season: { type: number }

target\_percent: { type: number }

forecast\_percent: { type: number }

8. Тестовые сценарии переходов ролей (синтетика) и конвертер дат из Excel

8.1. Сценарии

A. Полная структура (1111 участников)

- По итогам сезона проверить:

- Лучший Сотник → Царь, прежний Царь → Сотник.

- Сотники 2–5 остаются; 6–10 → Десятники.

- Топ-5 из «первых мест Десятников» → Сотники; 6–10 остаются Десятниками.

- Все остальные Десятники → Водители.

- Топ-90 из «первых мест Водителей» → Десятники.

- Проверить равномерную ротацию групп.

B. Неполные группы (например, 420 участников)

- Определить пропорциональное число Сотников/Десятников.

- Цели лидеров должны масштабироваться по фактическим «единицам».

- Переходы ролей — с пропорциональным количеством слотов.

C. Тай-брейки

- При равенстве total сортировать по большим личным, затем по более раннему достижению цели, затем случайно.

D. Новые пользователи и лимит

- Принимать X новых водителей на следующий сезон; при лимите вытеснять худших водителей прошлого сезона.

E. Пропуски данных и аномалии

- Нули в днях без импорта; уведомления о простоях.

- Дни > 16 часов — флаг «подозрительно»; расчёты не блокируются.

F. Первый сезон

- См. раздел 2.6: роли распределяются по местам 1, 2–11, 12–111, остальные.

G. Трёхкратный цикл сезонов

- Проверить, что после нескольких сезонов никто не остаётся без валидной привязки; количества ролей в допустимых диапазонах.

8.2. Пример конвертера дат Excel serial → дата

JavaScript (Node.js)

function excelSerialToDate(serial) {

const excelEpoch = new Date(Date.UTC(1899, 11, 30));

const msPerDay = 24 \* 60 \* 60 \* 1000;

return new Date(excelEpoch.getTime() + serial \* msPerDay);

}

// d.toISOString().slice(0,10) → YYYY-MM-DD

Python

from datetime import datetime, timedelta

def excel\_serial\_to\_date(serial: int):

excel\_epoch = datetime(1899, 12, 30)

return (excel\_epoch + timedelta(days=serial)).date()