

# Backend Guide

## Backend at a Glance

- Firestore-first model with per-user subcollections (reminders, routines, notification\_queue, push\_subscriptions).
- Cron runner reads denormalized queue to dispatch push/SMS/email efficiently.
- Routine generation uses deterministic IDs for idempotency and duplicate prevention.

## Table of Contents

Right-click and update field to generate TOC.

## 1. Firestore Data Model

### 1.1 users/{uid} (metadata)

Field	Type	Description	Example
timezone	string	User default timezone	America/New_York
email	string	User email	user@example.com
smsOptIn	boolean	SMS enabled flag	true
phoneNumber	string	E.164 phone number	+15555550123

### 1.2 users/{uid}/reminders

Field	Type	Description	Example
title	string	Reminder title	Take vitamins
notes	string	Optional notes (truncated in push payload)	With food
due_at	timestamp	Due time	2026-02-20T09:00Z
status	enum	pending done snoozed	pending
timezone	string	Timezone at creation	Europe/London
repeatRule	object null	Repeat logic for cloning	{type:'weekly'}
notifications[]	array	Channel + offsetMinutes + sent flag	[{type:'push', offsetMinutes:0}]
routineId	string?	Link to routine if generated	routine_abc
rootId	string?	Link to original reminder for repeats	rem_123

### 1.3 users/{uid}/routines

Field	Type	Description	Example
title	string	Routine name	Morning routine
active	boolean	Enabled/disabled	true
schedule	object	Type + days	{type:'weekly', days:[1,2,3]}
steps[]	array	Ordered steps with HH:MM and notifications	[{id:'1', time:'07:00'}]
timezone	string	Routine timezone abstraction	America/New_York
lastRun	timestamp	Last generation time	2026-02-20T00:00Z

### 1.4 users/{uid}/notification\_queue

Field	Type	Description	Example
reminderId	string	Parent reminder doc id	rem_123
channel	enum	push sms email	push
notificationId	string	Notification entry id	notif_1
scheduledAt	timestamp	dueAt - offsetMinutes	2026-02-20T08:55Z
dueAt	timestamp	Original due time	2026-02-20T09:00Z
sent	boolean	Dispatch status	false
timezone	string	Timezone context	Europe/London

### 1.5 users/{uid}/push\_subscriptions

Field	Type	Description	Example
endpoint	string	Push endpoint URL	https://...
keys.auth	string	Web Push auth key	base64...
keys.p256dh	string	Web Push p256dh key	base64...
created_at	timestamp	Registration time	2026-02-20T10:00Z

## 2. API Routes

Endpoint	Method	Auth	Purpose	Notes
/app/api/cron/run	GET	CRON_SECRET (Bearer or ?key=)	Minute-runner dispatch core	Queries queue window [now-2m, now]
/app/api/routines/[id]/run	POST	Firebase Auth token	Manual catch-up for routine enable	Generates next 24h block if missing
/app/api/push/subscribe	POST	Firebase Auth token	Register push subscription	Writes to push_subscriptions
/app/api/push/	POST	Firebase Auth	Remove push	Deletes device

unsubscribe		token	subscription	endpoint entry
-------------	--	-------	--------------	----------------

### 3. Cron/Queue Logic

#### Cron Dispatch Flow

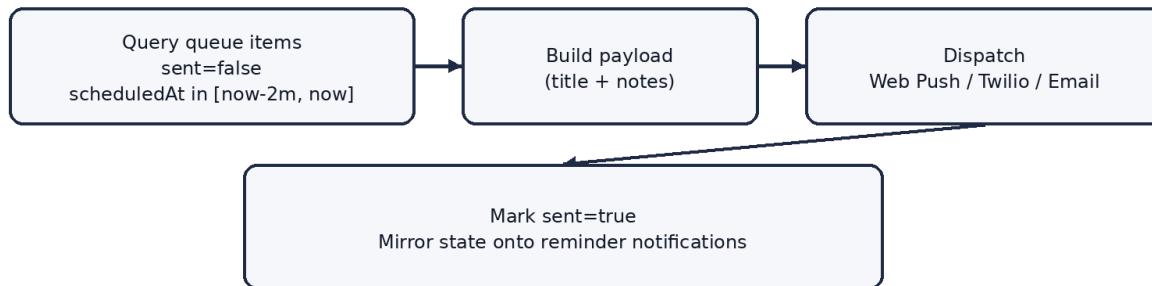


Figure: boxed flow diagram

### 4. Routine Generation Logic

- Rolling 24h deterministic generation: evaluates localNow to localNow + 1 day and maps HH:MM steps into reminder docs.
- Daily 12am run tracked via users/{uid}/meta/routineGenState.lastGenDate (yyyy-MM-dd) to prevent overlap.
- Idempotency: deterministic hash routineId:stepId:dateStr used as document id; set(merge=True) prevents duplicates.
- Disable/delete cascades: removeRoutineQueue() purges future queue footprint and deletes generated reminders tied to routineId.

### 5. Indexes

Index	Query it supports	Needed because
reminders: status ASC, due_at ASC	Pull pending reminders efficiently	Avoids expensive full scans
notification_queue: sent ASC, scheduledAt ASC	Cron retrieves unsent items ordered by scheduledAt	Core path for minute runner
routines: created_at DESC	Client list ordering	Stable ordering without extra reads

## 6. Security Rules (Summary)

### ✓ Best Practice: Tenant isolation

- Use match /users/{uid}/{collection=\*\*} with request.auth.uid == uid to prevent cross-tenant reads/writes.