10

O‘NINCHI MA’RUZA

Korxona ilovalarini dasturlash · 10-mavzu

Asinxron qayta ishlash

Navbat arxitekturasi · Job va dispatch · Workerlar · Qayta urinish va idempotentlik · Rejalashtirish · Bildirishnomalar

Queues | Redis | Scheduler | Idempotentlik | Notifications

ISINISH

Buxgalter “Oylik savdo hisoboti” tugmasini bosdi: 50 000 qator, tayyorlash 90 soniya. Nima bo‘ladi?

nginx va php-fpm odatda 60 soniyada so‘rovni uzadi. Hozircha javob bermang — taxmin qiling.

1

Foydalanuvchi 90 soniya oq ekranga qarab turadimi? Brauzer nima ko‘rsatadi?

2

Shu vaqtda yana 20 kishi hisobot so‘rasa, qolgan foydalanuvchilarning buyurtmalari nima bo‘ladi?

3

Buxgalter sabri tugab tugmani yana bosdi. Nechta hisobot tayyorlanadi?

O‘QUV MAQSADLARI

Bu ma’ruza oxirida siz quyidagilarni bajara olasiz…

1. **Tushuntirish** — sinxron va asinxron qayta ishlash farqini hamda navbat (producer → queue → worker) arxitekturasini
2. **Yozish** — ShouldQueue job sinfini va uni dispatch, onQueue, delay, afterCommit bilan yuborishni
3. **Sozlash** — Redis navbatini, queue:work workerlarini Supervisor yoki Docker orqali, Horizon monitoringini
4. **Qo‘llash** — tries, backoff, timeout, failed() va failed_jobs bilan ishonchli qayta urinishlarni
5. **Loyihalash** — takroriy bajarilishga chidamli (idempotent) fon vazifalarini: ShouldBeUnique, qulf, belgi
6. **Rejalashtirish** — routes/console.php dagi Schedule bilan davriy vazifalarni Toshkent vaqtida
7. **Yuborish** — Notification sinfi orqali mail va database kanallariga bildirishnomalarni

01

Asinxronlik va navbat arxitekturasi

Sinxron so‘rov chegarasi · producer → queue → worker · drayverlar · Redis

Sinxron / asinxron

Queue

Redis

202 Accepted

So‘rov faqat vazifani qo‘yadi, ishni worker bajaradi

Og‘ir yoki tashqi tizimga bog‘liq ish HTTP so‘rovdan ajratiladi: so‘rov millisekundlarda javob beradi, ish fon rejimida bajariladi.

Job

Bajariladigan ishning tavsifi: sinf nomi + parametrlar. Navbatga JSON payload sifatida yoziladi.

Queue (navbat)

Joblar ombori: Redis ro‘yxati `queues:reports`. Birinchi kelgan — birinchi bajariladi (FIFO).

Worker

`php artisan queue:work` — navbatdan jobni olib, `handle()` ni chaqiradigan uzoq yashovchi jarayon.

202 Accepted

“So‘rov qabul qilindi, natija keyinroq” degan HTTP javob — asinxron API uchun to‘g‘ri kod.

Asinxronlikning narxi: natija darhol emas, “oxir-oqibat” tayyor bo‘ladi (eventual consistency) va qo‘shimcha jarayonlarni kuzatish kerak.

Hisobot so‘rovining ikki yo‘li

✗ Sinxron: hammasi so‘rov ichida

- Controller 90 s davomida SQL va Excel bilan band
- 60 s da nginx 504 qaytaradi — fayl yo‘qoladi
- php-fpm jarayoni band: boshqalar navbatda turadi
- Qayta bosish — ikkinchi og‘ir ish boshlanadi

✓ Asinxron: so‘rov → navbat → worker

- Controller `GenerateSalesReport::dispatch()` qiladi
- 50 ms da `202 Accepted` qaytadi
- Worker jobni bajaradi, xato bo‘lsa qayta urinadi
- Tayyor bo‘lgach — bildirishnoma va yuklab olish havolasi

Qoida: foydalanuvchi natijani shu zahoti ko‘rishi shart bo‘lmagan har qanday og‘ir ish — navbat nomzodi.

▶ INTERAKTIV

Sinxronmi yoki navbatgami?

Har bir amalni to‘g‘ri qutiga joylashtiring. (Interaktiv slaydlarda bosib bajariladi.)

Buyurtmani tasdiqlashda qoldiqni rezerv qilish

Oylik savdo hisobotini Excel ga eksport qilish

Login va parolni tekshirish

Kam qoldiq haqida 15 omborchiga e-mail

Formadagi STIR 9 raqamligini tekshirish

1C ga kunlik savdolarni yuborish

Buyurtma summasini QQS bilan hisoblash

Yuklangan 10 000 qatorli CSV ni import qilish

So‘rov ichida (sinxron)

Navbatga (asinxron)

Navbatni Redis ga ulash

```
# .env
QUEUE_CONNECTION=redis
REDIS_CLIENT=phpredis
REDIS_HOST=redis
REDIS_PORT=6379
REDIS_QUEUE=default
REDIS_QUEUE_RETRY_AFTER=90

# Redis ichida navbatni ko‘rish (docker compose exec redis sh)
# redis-cli LLEN omborpro_database_queues:reports
```

▶ NATIJA

```
(integer) 3
```

`QUEUE_CONNECTION=redis`

`config/queue.php` dagi `redis` ulanishi tanlanadi

`REDIS_CLIENT=phpredis`

PHP kengaytmasi (tez); muqobil — `predis/predis` paketi

`REDIS_QUEUE_RETRY_AFTER`

band qilingan job necha soniyadan keyin “yo‘qolgan” deb qaytariladi

`queues:reports`

har bir navbat — Redis ro‘yxati; kechiktirilganlar `:delayed` to‘plamida

Kalit oldidagi prefiks `REDIS_PREFIX` dan olinadi (standart: ilova nomi + `_database_`).

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

1-bo‘lim yakuni: Asinxronlik va navbat arxitekturasi

1

Nima uchun og‘ir hisobotni controller ichida tayyorlash boshqa foydalanuvchilarga ham zarar qiladi?

Savol 1

2

Navbat arxitekturasining uch qismini ayting.

Savol 2

3

Qoldiqni rezerv qilishni navbatga qo‘yish nima uchun xato?

Savol 3

02

Job sinflari va yuborish (dispatch)

make:job · ShouldQueue · SerializesModels · onQueue, delay, afterCommit · chain va batch

ShouldQueue | dispatch | afterCommit | Bus::chain

JOB SINFI

GenerateSalesReport — Laravel 12 shabloni

```php
namespace App\Jobs;
use App\Models\User;
use App\Services\ReportService;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Foundation\Queue\Queueable;

class GenerateSalesReport implements ShouldQueue
{
    use Queueable;
    public function __construct(
        public User $user, public string $from, public string $to,
    ) {}

    public function handle(ReportService $reports): void
    {
        $reports->exportSales($this->user, $this->from, $this->to);
    }
}
```

▶ NATIJA

```
$ php artisan make:job GenerateSalesReport
   INFO  Job [app/Jobs/GenerateSalesReport.php] created
successfully.
```

implements ShouldQueue

belgi interfeysi: usiz job sinxron bajariladi

use Queueable

Laravel 12 traiti: dispatch, navbat sozlamalari, modellarni seriyalash

`public User $user`

navbatga faqat `User` sinfi va `id` yoziladi, worker modelni qayta o‘qiydi

`handle(ReportService $reports)`

metod parametrlari konteynerdan in’ektsiya qilinadi

DISPATCH

Controller faqat vazifani qo‘yadi

```php
public function store(Request $request): JsonResponse
{
    $data = $request->validate([
        'from' => ['required', 'date'],
        'to' => ['required', 'date', 'after_or_equal:from'],
    ]);

    GenerateSalesReport::dispatch(
        $request->user(), $data['from'], $data['to'],
    )->onQueue('reports');

    return response()->json([
        'message' => 'Hisobot tayyorlanmoqda',
    ], 202);
}
```

▶ NATIJA

```
HTTP/1.1 202 Accepted
{"message": "Hisobot tayyorlanmoqda"}
```

`validate([...])`

noto‘g‘ri sana navbatga tushmasin — tekshiruv sinxron

`->onQueue('reports')`

og‘ir hisobotlar alohida navbatda, e-mail larni to‘smaydi

202

“qabul qilindi” — natija keyin; `200` emas

YUBORISH VARIANTLARI

delay, afterCommit, dispatchIf, chain

```php
// 5 daqiqadan keyin bajarilsin
SendLowStockAlert::dispatch()->delay(now()->addMinutes(5));

// tranzaksiya muvaffaqiyatli commit bo‘lgandan keyingina
DB::transaction(function () use ($order) {
    $this->stock->ship($order);
    SendLowStockAlert::dispatch()->afterCommit();
});

// shart bajarilsa
SendLowStockAlert::dispatchIf($order->items()->exists());

// ketma-ket: biri yiqilsa, keyingisi ishga tushmaydi
Bus::chain([
    new GenerateSalesReport($user, '2026-10-01', '2026-10-31'),
    new SendLowStockAlert(),
])->onQueue('reports')->dispatch();
```

▶ NATIJA

```
Redis: queues:default:delayed  ← 1 job (5 daqiqa)
Redis: queues:default          ← commit dan keyin 1 job
Redis: queues:reports          ← zanjirning 1-jobi
```

`delay(...)`

job `:delayed` to‘plamida vaqti kelguncha turadi

`afterCommit()`

rollback bo‘lsa job umuman yuborilmaydi

`Bus::chain`

keyingi job oldingisi muvaffaqiyatli tugagandagina navbatga qo‘yiladi

`Bus::batch`

parallel joblar + `then/catch/finally`; `Batchable` trait va `job_batches` jadvali kerak

Navbat ichida nima bor: JSON payload

```php
$queue = new SplQueue();
$push = function (string $job, array $data) use ($queue): void {
    $queue->enqueue(json_encode(['job' => $job, 'data' => $data]));
};

$push('GenerateSalesReport', ['user_id' => 7, 'from' => '2026-10-01']);
$push('SendLowStockAlert', []);
echo 'Navbatda: ' . count($queue) . PHP_EOL;

while (!$queue->isEmpty()) {
    $p = json_decode($queue->dequeue(), true);
    echo "Bajarildi: {$p['job']} " . json_encode($p['data']) . PHP_EOL;
}
```

▶ NATIJA

```
Navbatda: 2
Bajarildi: GenerateSalesReport {"user_id":7,"from":"2026-10-01"}
Bajarildi: SendLowStockAlert []
```

`json_encode`

Laravel ham jobni JSON payload qilib saqlaydi (sinf nomi + seriyalangan obyekt)

SplQueue

FIFO: birinchi qo‘yilgan birinchi olinadi — Redis ro‘yxati ham shunday

`['user_id' => 7]`

modelning o‘zi emas, identifikatori — SerializesModels g‘oyasi

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

2-bo‘lim yakuni: Job sinflari va yuborish (dispatch)

1

Job sinfida `ShouldQueue` yozilmasa, `dispatch()` nima qiladi?

Savol 1

2

`SerializesModels` modelni navbatga qanday ko‘rinishda yozadi?

Savol 2

3

`afterCommit()` qaysi xatodan saqlaydi?

Savol 3

03

Workerlar: queue:work, Supervisor, Horizon

Uzoq yashovchi jarayon · navbat ustuvorligi · timeout va retry_after · deploy · monitoring

queue:work | Supervisor | Horizon | queue:restart

queue:work parametrlari

```
# ishlab chiqish: har job uchun kod qayta yuklanadi (sekin)
php artisan queue:listen

# production: uzoq yashovchi jarayon
php artisan queue:work redis \
    --queue=reports,default \
    --tries=3 --timeout=60 --sleep=3 \
    --max-jobs=500 --max-time=3600

# deploydan keyin: joriy job tugagach workerlar chiqadi
php artisan queue:restart
```

▶ NATIJA

```
  INFO  Processing jobs from the [reports,default] queues.
2026-10-27 10:15:02 App\Jobs\GenerateSalesReport ..... RUNNING
2026-10-27 10:15:04 App\Jobs\GenerateSalesReport . 1.84s DONE
2026-10-27 10:15:04 App\Jobs\SendLowStockAlert ....... RUNNING
2026-10-27 10:15:05 App\Jobs\SendLowStockAlert .. 512ms FAIL
```

`--queue=reports,default`

ustuvorlik: `reports` bo‘shamaguncha `default` olinmaydi

`--timeout=60`

`retry_after` (90) dan kichik bo‘lishi shart

`--max-jobs / --max-time`

xotira o‘sishidan himoya: worker o‘zi chiqib, qayta ishga tushiriladi

`queue:restart`

yangi kod faqat workerlar qayta ishga tushganda yuklanadi

SUPERVISOR

/etc/supervisor/conf.d/omborpro-worker.conf

```
[program:omborpro-worker]
process_name=%(program_name)s_%(process_num)02d
directory=/var/www/omborpro
command=php artisan queue:work --queue=reports,default --max-time=3600
user=www-data
numprocs=2
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
stopwaitsecs=3600
redirect_stderr=true
stdout_logfile=/var/www/omborpro/storage/logs/worker.log
```

▶ NATIJA

```
$ sudo supervisorctl reread && sudo supervisorctl update
$ sudo supervisorctl status
omborpro-worker:omborpro-worker_00   RUNNING   pid 2114, uptime
0:00:12
omborpro-worker:omborpro-worker_01   RUNNING   pid 2115, uptime
0:00:12
```

`numprocs=2`

ikkita parallel worker jarayoni

`autorestart=true`

worker yiqilsa yoki `--max-time` da chiqsa — qayta ishga tushadi

`stopwaitsecs=3600`

to‘xtatishda eng uzun job tugashini kutadi (timeout dan katta bo‘lsin)

`user=www-data`

fayllar va loglar veb-server bilan bir xil egada yaratiladi

MONITORING

Horizon va kuzatiladigan ko‘rsatkichlar

Ko‘rinmaydigan fon jarayoni — eng xavfli jarayon. Navbat to‘xtasa, foydalanuvchi buni bir necha soatdan keyin sezadi.

1 Horizon paneli

`/horizon`: o‘tkazuvchanlik, bajarilish vaqti, muvaffaqiyatsiz joblar, teglar. Faqat Redis bilan ishlaydi.

2 Kod sifatidagi sozlama

`config/horizon.php`: supervisorlar, navbatlar, `maxProcesses`, `balance => 'auto'`.

3 Kirish huquqi

`viewHorizon` Gate: faqat `admin` roli. Panelda payloadlar ko‘rinadi!

4 Backlog

Navbat uzunligi o‘sib borsa — workerlar yetmayapti yoki biri osilib qolgan.

5 Kutish vaqti

Eng eski job necha daqiqa kutyapti? Foydalanuvchi tajribasining asosiy o‘lchovi.

6 Deploy

Horizon bilan `php artisan horizon:terminate` — `queue:restart` o‘rniga.

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

3-bo‘lim yakuni: Workerlar: queue:work, Supervisor, Horizon

1

Nima uchun deploydan keyin `queue:restart` kerak?

Savol 1

2

`--queue=reports,default` yozuvida tartib nimani anglatadi?

Savol 2

3

`timeout` `retry_after` dan katta bo‘lsa nima bo‘ladi?

Savol 3

MA’RUZADAGI MASHQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

1-topshiriq: Navbatlar ustuvorligi

Worker `--queue=reports,default,low` kabi navbatlar ro‘yxati bilan ishga tushirilgan: u avval birinchi navbatni to‘liq bo‘shatadi, keyin ikkinchisiga o‘tadi va h.k. Har bir navbat ichida joblar kelish tartibida (FIFO) olinadi. Joblar ro‘yxati kelish tartibida berilgan: har biri `['id' => int, 'queue' => string]`. Worker tinglamaydigan navbatdagi joblar bajarilmaydi. Navbatlar ro‘yxatida takroriy nom bo‘lsa, faqat birinchisi hisobga olinadi. Navbatlar ro‘yxati bo‘sh bo‘lsa — `InvalidArgumentException`.

Imzo: `function navbat_tartibi(array $navbatlar, array $joblar): array`

Qaytaradi: Bajarilish tartibidagi job `id` lari ro‘yxati (`list<int>`).

MISOL · CHAQIRUV

```
navbat_tartibi(['reports', 'default'], [['id' => 1, 'queue' =>
'default'], ['id' => 2, 'queue' => 'reports'], ['id' => 3,
'queue' => 'default']])
```

MISOL · NATIJA

```
[2,1,3]
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

04

Qayta urinishlar, failed jobs va idempotentlik

tries · backoff · failed() · failed_jobs · at-least-once · ShouldBeUnique · WithoutOverlapping

tries / backoff

failed_jobs

Idempotentlik

ShouldBeUnique

QAYTA URINISH

tries, backoff, timeout va failed()

```php
class SendLowStockAlert implements ShouldQueue
{
    use Queueable;

    public int $tries = 5;            // jami urinishlar
    public int $maxExceptions = 3;    // istisnolar chegarasi
    public int $timeout = 60;         // retry_after (90) dan kichik
    public array $backoff = [10, 60, 300];

    public function failed(?Throwable $e): void
    {
        Log::error('Kam qoldiq xabari yuborilmadi', [
            'error' => $e?->getMessage(),
        ]);
    }
}
```

▶ NATIJA

```
$ php artisan queue:failed
  2026-10-27 09:07:41 b1e7c0f2-... redis@default
App\Jobs\SendLowStockAlert
```

`$tries = 5`

1 ta asosiy + 4 ta qayta urinish

`$backoff = [10, 60, 300]`

10 s, 60 s, keyin har safar 300 s kutish

`failed(?Throwable $e)`

urinishlar tugaganda bir marta chaqiriladi

`$maxExceptions`

release() bilan qaytarishlar emas, haqiqiy istisnolar soni

Kechikishlar jadvali qanday hisoblanadi

```php
function backoff(int $attempt, array $steps): int
{
    return $steps[$attempt - 1] ?? $steps[count($steps) - 1];
}

$steps = [10, 60, 300];
$tries = 5;
for ($a = 1; $a <= $tries; $a++) {
    if ($a === $tries) {
        echo "$a-urinish: xato -> failed_jobs" . PHP_EOL;
        break;
    }
    $d = backoff($a, $steps);
    echo "$a-urinish: xato, $d s kutish" . PHP_EOL;
}
```

▶ NATIJA

```
1-urinish: xato, 10 s kutish
2-urinish: xato, 60 s kutish
3-urinish: xato, 300 s kutish
4-urinish: xato, 300 s kutish
5-urinish: xato -> failed_jobs
```

`$steps[$attempt - 1]`

n-urinishdan keyingi kechikish — massivning (n−1)-elementi

?? oxirgi element

massiv tugasa, oxirgi qiymat takrorlanadi

`$a === $tries`

oxirgi urinish ham yiqilsa — `failed_jobs` va `failed()`

FAILED JOBS

Muvaffaqiyatsiz joblar bilan ishlash

Buyruq	Vazifasi	Qachon
`queue:failed`	`failed_jobs` ro‘yxati: uuid, navbat, sinf, vaqt	Har kuni / alert kelganda
`queue:retry <uuid>`	Bitta jobni navbatga qaytaradi	Sabab tuzatilgandan keyin
`queue:retry all`	Barcha muvaffaqiyatsizlarni qaytaradi	Tashqi tizim tiklangach
`queue:forget <uuid>`	Bitta yozuvni o‘chiradi	Job endi kerak emas
`queue:flush`	Barcha yozuvlarni o‘chiradi	Ehtiyotkorlik bilan!
`queue:prune-failed --hours=168`	7 kundan eski yozuvlarni tozalaydi	Rejalashtirilgan vazifa

Payload ichida shaxsiy ma’lumot bo‘lishi mumkin — `failed_jobs` ni cheksiz saqlamang.

IDEMPOTENTLIK

Bir kunda bitta xabar: ShouldBeUnique + belgi

```php
class SendLowStockAlert implements ShouldQueue, ShouldBeUnique
{
    use Queueable;
    public int $uniqueFor = 3600;

    public function handle(ProductRepository $products): void
    {
        $key = 'low-stock-alert:' . now()->toDateString();
        $items = $products->lowStock();
        if ($items->isEmpty() || Cache::has($key)) {
            return;
        }
        $users = User::role('omborchi')->where('is_active', 1)->get();
        $note = new LowStockNotification($items->toArray());
        Notification::send($users, $note);
        Cache::put($key, true, now()->endOfDay());
    }
}
```

▶ NATIJA

```
09:00  dispatch → bajarildi, 3 omborchiga xabar
09:00  takroriy dispatch → ShouldBeUnique: navbatga qo‘yilmadi
11:30  queue:retry → Cache::has → return (xabar takrorlanmadi)
```

ShouldBeUnique

bir xil job navbatda turgan bo‘lsa, yangisi qo‘yilmaydi

`$uniqueFor = 3600`

qulf ko‘pi bilan 1 soat ushlanadi

`Cache::has($key)`

bugun yuborilganmi? — tekshirish yon ta’sirdan oldin

`Cache::put(...)`

belgi yon ta’sirdan keyin: yiqilsa, qayta urinish xatni yuboradi

▶ INTERAKTIV

Qayta urinish va idempotentlik

To‘g‘ri javobni tanlang. (Interaktiv slaydlarda bosib bajariladi.)

1 `$tries = 4`, `$backoff = [5, 30]`. **3-urinishdan oldin necha soniya kutiladi?**

A) 5
B) 30
C) 35
D) 0

2 **Job 120 s ishlaydi, `retry_after = 90`, `timeout = 150`. Nima bo‘ladi?**

A) Job 150 s da to‘xtatiladi, takrorlanmaydi
B) Hech narsa — hammasi to‘g‘ri
C) 90 s da job navbatga qaytadi va ikkinchi worker uni parallel bajaradi
D) Job darhol failed_jobs ga tushadi

3 **Qaysi amal tabiatan idempotent?**

A) `paid_amount += 150000`
B) Hisobotni `sales-2026-10.xlsx` nomi bilan qayta yozish
C) Har chaqiriqda e-mail yuborish
D) `stock_movements` ga yangi qator qo‘shish

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

4-bo‘lim yakuni: Qayta urinishlar, failed jobs va idempotentlik

1

Vaqtinchalik va doimiy xatoga bittadan misol keltiring. Qaysi birini qayta urinish foydali?

Savol 1

2

Nima uchun navbatdagi job ikki marta bajarilishi mumkin? Ikki sababni ayting.

Savol 2

3

Bajarilganlik belgisini yon ta’sirdan oldin qo‘yish nima uchun xato?

Savol 3

MA’RUZADAGI MASHQ

Har kim o‘z qurilmasida · 10 daqiqa · O‘rta · 20 ball

2-topshiriq: Qayta urinish yoki failed_jobs?

Job `$urinish`-urinishda yiqildi (urinishlar 1 dan sanaladi). Job sozlamalari: `$tries` — jami ruxsat etilgan urinishlar, `$backoff` — kechikishlar massivi (soniya). Agar `$urinish >= $tries` bo‘lsa, job muvaffaqiyatsiz (`fail`). Aks holda qayta uriniladi: kechikish `$backoff[$urinish - 1]`, massiv qisqa bo‘lsa — uning oxirgi elementi, massiv bo‘sh bo‘lsa — 0. `$urinish < 1` yoki `$tries < 1` bo‘lsa — `InvalidArgumentException`.

Imzo: `function keyingi_qadam(int $urinish, int $tries, array $backoff): array`

Qaytaradi: `['action' => 'retry', 'delay' => int]` yoki `['action' => 'fail', 'delay' => 0]` — kalitlar aynan shu tartibda.

MISOL · CHAQIRUV

```
keyingi_qadam(1, 5, [10, 60, 300])
```

MISOL · NATIJA

```
{"action":"retry","delay":10}
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

05

Vazifalarni rejalashtirish (Task Scheduling)

routes/console.php · Schedule::job / command · chastota · timezone · onOneServer · cron

Schedule | routes/console.php | cron | Asia/Tashkent

routes/console.php

```php
use App\Jobs\SendLowStockAlert;
use Illuminate\Support\Facades\Schedule;

Schedule::job(new SendLowStockAlert())
    ->weekdays()
    ->at('09:00')
    ->timezone('Asia/Tashkent')
    ->onOneServer();

Schedule::command('queue:prune-failed --hours=168')
    ->dailyAt('02:00')
    ->timezone('Asia/Tashkent');

Schedule::command('rates:fetch')
    ->cron('30 9 * * 1-5')
    ->withoutOverlapping()
    ->appendOutputTo(storage_path('logs/schedule.log'));
```

▶ NATIJA

```
$ php artisan schedule:list
  0  9 * * 1-5  App\Jobs\SendLowStockAlert ........ Next Due: 21
hours from now
  0  2 * * *    php artisan queue:prune-failed --hours=168 ...
Next Due: 14 hours from now
  30 9 * * 1-5  php artisan rates:fetch ......... Next Due: 21
hours from now
```

`Schedule::job(...)`

jobni navbatga qo‘yadi — og‘ir ishni worker bajaradi

`->weekdays()->at('09:00')`

dushanba–juma 09:00 (cron: `0 9 * * 1-5`)

`->timezone('Asia/Tashkent')`

aks holda UTC bo‘yicha — Toshkentda 14:00

`->onOneServer()`

bir nechta serverda ham bir marta; umumiy Redis kesh kerak

CHASTOTA

Ko‘p ishlatiladigan chastota usullari

Usul	Cron ko‘rinishi	OmborPro misoli
`everyFiveMinutes()`	`*/5 * * * *`	Dashboard keshini yangilash
`hourly()`	`0 * * * *`	1C dan buyurtma holatlarini tortish
`dailyAt('02:00')`	`0 2 * * *`	Eski loglar va failed jobsni tozalash
`weekdays()->at('09:00')`	`0 9 * * 1-5`	Kam qoldiq xabari
`weeklyOn(1, '08:00')`	`0 8 * * 1`	Haftalik savdo xulosasi
`monthlyOn(1, '06:00')`	`0 6 1 * *`	O‘tgan oy hisobotini tayyorlash

Cron maydonlari: daqiqa, soat, oy kuni, oy, hafta kuni (0 — yakshanba, 1 — dushanba).

ISHGA TUSHIRISH

Scheduler production da

1

Cron qatori

```
* * * * * cd /var/www/omborpro && php artisan schedule:run >> /dev/null 2>&1
```

2

Yoki Docker

Alohida `scheduler` servisi: `php artisan schedule:work`, faqat bitta replika

3

Tekshirish

`schedule:list` — jadval va keyingi vaqt; `schedule:test` — darhol sinash

4

Kuzatish

`appendOutputTo`, `onFailure`, `pingOnSuccess` — vazifa jim yiqilmasin

5

Ko‘p server

`onOneServer()` + umumiy Redis kesh

Scheduler ishlamasa xato chiqmaydi — shunchaki hech narsa bo‘lmaydi. Heartbeat monitoringi shuning uchun kerak.

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

5-bo‘lim yakuni: Vazifalarni rejalashtirish (Task Scheduling)

1

Laravel 12 da rejalashtirilgan vazifalar qaysi faylda yoziladi?

Savol 1

2

`->timezone('Asia/Tashkent')` yozilmasa va `APP_TIMEZONE=UTC` bo‘lsa, `dailyAt('09:00')` Toshkent vaqti bilan qachon ishlaydi?

Savol 2

3

`onOneServer()` nima uchun kerak va u nimaga tayanadi?

Savol 3

MA’RUZADAGI MASHQ

Har kim o‘z qurilmasida · 10 daqiqa · O‘rta · 20 ball

3-topshiriq: Cron ifodasi vaqtga mosmi?

Soddalashtirilgan cron ifodasi 5 maydondan iborat: daqiqa (0–59), soat (0–23), oy kuni (1–31), oy (1–12), hafta kuni (0–6, 0 — yakshanba). Har bir maydon vergul bilan ajratilgan qismlardan iborat bo‘lishi mumkin; qism — `*`, son (`9`), oraliq (`1-5`) yoki qadam (`*/15`: qiymat − maydon minimumi qadamga bo‘linadi). Vaqt `Y-m-d H:i` formatida. Barcha 5 maydon mos kelsa — `true`. Maydonlar soni 5 emas yoki vaqt formati noto‘g‘ri bo‘lsa — `InvalidArgumentException`.

Imzo: `function cron_mos(string $ifoda, string $vaqt): bool`

Qaytaradi: `true` — vazifa shu daqiqada ishga tushadi, aks holda `false`.

MISOL · CHAQIRUV	MISOL · NATIJA
`cron_mos('0 9 * * 1-5', '2026-10-27 09:00')`	`true`

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

06

Bildirishnomalar (Notifications)

Notification va Mailable · via() · toMail · toArray · database kanali · navbatga qo‘yish

Notification | mail | database | Notifiable

NOTIFICATION SINFI

LowStockNotification: kanallar va database

```php
class LowStockNotification extends Notification implements ShouldQueue
{
    use Queueable;
    public function __construct(public array $items) {}

    public function via(object $notifiable): array
    {
        return $notifiable->email ? ['mail', 'database'] : ['database'];
    }

    public function toArray(object $notifiable): array
    {
        return [
            'count' => count($this->items),
            'skus' => array_column($this->items, 'sku'),
        ];
    }
}
```

▶ NATIJA

```
notifications.data = {"count":2,"skus":["UN-001","UN-014"]}
```

implements ShouldQueue

har bir qabul qiluvchi × kanal — alohida job

`public array $items`

oddiy massiv: worker da hisoblangan ustunlar yo‘qolmaydi

`via()`

kanallarni shartga qarab tanlash mumkin

`toArray()`

`database` kanali uchun `data` ustuniga JSON

MAIL KANALI

toMail(): MailMessage quruvchisi

```php
public function toMail(object $notifiable): MailMessage
{
    $mail = (new MailMessage())
        ->subject('Kam qoldiq: ' . count($this->items) . ' ta mahsulot')
        ->greeting("Assalomu alaykum, {$notifiable->name}!")
        ->line('Quyidagi mahsulotlar minimal qoldiqdan kam:');

    foreach (array_slice($this->items, 0, 10) as $i) {
        $mail->line("{$i['sku']} {$i['name']}: {$i['available']} ta");
    }

    return $mail
        ->action('Qoldiqlarni ochish', url('/stocks'))
        ->line('Bu xat OmborPro tomonidan avtomatik yuborildi.');
}
```

▶ NATIJA

```
Mavzu: Kam qoldiq: 2 ta mahsulot
Assalomu alaykum, Dilshod!
Quyidagi mahsulotlar minimal qoldiqdan kam:
UN-001 Un 1-nav 50 kg: 4 ta
UN-014 Shakar 50 kg: 0 ta
[ Qoldiqlarni ochish ]
```

subject / greeting / line

matn qatorlari — tayyor HTML shablonga joylanadi

array_slice(..., 0, 10)

xat qisqa bo‘lsin; to‘liq ro‘yxat — havola orqali

action(...)

asosiy tugma: foydalanuvchi keyingi qadamni biladi

notify, Notification::send, unreadNotifications

```php
// bitta foydalanuvchiga
$user->notify(new LowStockNotification($items));

// rol bo‘yicha ko‘p foydalanuvchiga (spatie/laravel-permission)
$users = User::role('omborchi')->where('is_active', true)->get();
Notification::send($users, new LowStockNotification($items));

// tizimda hisobi yo‘q manzilga
Notification::route('mail', 'ombor@example.uz')
    ->notify(new LowStockNotification($items));

// panel: o‘qilmaganlar soni va hammasini o‘qilgan deb belgilash
$count = $user->unreadNotifications()->count();
$user->unreadNotifications->markAsRead();
```

▶ NATIJA

```
SQL: select count(*) from notifications
     where notifiable_id = 7 and read_at is null
→ 3
```

`User::role('omborchi')`

spatie paketining scope si — rol bo‘yicha foydalanuvchilar

`Notification::route`

“on-demand”: modelsiz manzilga yuborish

`unreadNotifications`

`read_at IS NULL` bo‘lgan yozuvlar; Livewire qo‘ng‘iroqchasi uchun

KENG TARQALGAN XATO

Bildirishnomaga nima beriladi?

✗ Noto‘g‘ri: Eloquent kolleksiyasi

- `new LowStockNotification($products)` — SQL da `available` hisoblangan
- Navbatda faqat `Product` id lari saqlanadi
- Worker modellarni qayta o‘qiydi — `available` yo‘q
- Xatda bo‘sh qiymatlar yoki ogohlantirish

✓ To‘g‘ri: kerakli maydonlar massivi

- `$products->map->only(['sku', 'name', 'available'])`
- Payload o‘zi to‘liq — bazaga qayta murojaat yo‘q
- Xat aynan voqea paytidagi holatni ko‘rsatadi
- Payload kichik: faqat kerakli maydonlar

Umumiy qoida: navbatga “nima bo‘lganini” tasvirlaydigan oddiy ma’lumot beriladi, jonli obyekt emas.

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

6-bo‘lim yakuni: Bildirishnomalar (Notifications)

1

Notification va Mailable dan qaysi birini kam qoldiq xabari uchun tanlaysiz va nima uchun?

Savol 1

2

`database` kanali ma’lumotni qayerga yozadi va qaysi metod uni quradi?

Savol 2

3

Navbatga qo‘yiladigan bildirishnomaga Eloquent kolleksiyasi berish nima uchun xavfli?

Savol 3

XULOSA

Asosiy xulosalar

1 Og‘ir ish — navbatga

So‘rov jobni `dispatch` qilib 202 qaytaradi, worker ishni fon rejimida bajaradi.

2 Job = sinf + seriyalangan parametrlar

`ShouldQueue` + `Queueable`; modellar id sifatida saqlanadi, tranzaksiyada `afterCommit()`.

3 Worker — uzoq yashovchi jarayon

Supervisor/Docker kuzatadi, deploydan keyin `queue:restart`, `timeout < retry_after`.

4 Xatoga tayyor bo‘ling

`tries`, `backoff`, `failed()` va `failed_jobs`; sababni tuzatib, keyin `queue:retry`.

5 Kamida bir marta ≠ aynan bir marta

Joblar idempotent: ShouldBeUnique, qulf, belgi — yon ta’sirdan keyin.

6 Jadval kodda, xabar kanallarda

`routes/console.php` da `Schedule` + Asia/Tashkent; Notification — mail va database.

TAKRORLASH

Yakunlashdan oldin sakkizta savol

1. Navbat arxitekturasining uch qismi?
2. Laravel 12 job shablonidagi trait?
3. `afterCommit()` nima qiladi?
4. `--queue=reports,default` ma’nosi?
5. `$backoff = [10, 60]`, 4-qayta urinishdan oldin kechikish?
6. Urinishlar tugagan job qayerga tushadi?
7. Scheduler serverda qanday ishga tushadi?
8. Bildirishnoma kanallari qaysi metodda belgilanadi?

AMALIY MASHG‘ULOT

10-amaliy mashg‘ulot: 5 ta PHP topshiriq va loyiha bosqichi

4 **Kam qoldiqdagi mahsulotlar**
Oson · 10 ball

5 **Kam qoldiq xabari matni**
O‘rta · 20 ball

6 **Idempotentlik belgisi (Cache::add modeli)**
O‘rta · 20 ball

7 **Navbat simulyatori: tries va noyob joblar**
Qiyin · 30 ball

8 **Hisobot davrini oylarga bo‘lish (Bus::batch uchun)**
Qiyin · 30 ball

Telefon kamerasi bilan skanerlang

PHP kod saytda yoziladi va server tomonidan avtomatik tekshiriladi.

Loyiha havolasini ham shu sahifada topshirasiz; o‘qituvchi baho qo‘yadi.

CHIQISH CHIPTASI

Darsni tark etishdan oldin

Javoblaringizni qog‘ozga yozing va o‘qituvchiga topshiring.

1

OmborPro da yana qaysi ikki amalni navbatga o‘tkazgan bo‘lardingiz va nima uchun?

2

`SendLowStockAlert` ikki marta bajarilsa, omborchilar ikki xat olmasligi uchun nima qilinadi?

3

Scheduler 3 kun ishlamaganini qanday qilib tezroq bilib olish mumkin?