13

O‘N UCHINCHI MA’RUZA

Korxona ilovalarini dasturlash · 13-mavzu

Unumdorlik va kuzatuvchanlik

O‘lchash · N+1 va eager loading · indekslar · Redis kesh · strukturali log · health check va Sentry

N+1 | Indekslar | Redis kesh | Loglash | Sentry

ISINISH

Hisobot sahifasi noutbukda 0,2 soniyada ochiladi, productionda esa 9 soniya. Kod bir xil. Nega?

Javob bermasdan oldin farq qayerdan kelishini taxmin qiling: kodmi, ma’lumotmi, servermi?

1

Lokal bazada 20 ta buyurtma, productionda 200 ming. Bu qaysi so‘rovlarga ta’sir qiladi?

2

Sahifa ochilganda bazaga necha marta murojaat qilinayotganini qanday bilasiz?

3

Foydalanuvchi “sekin” deb yozganda, siz qaysi ma’lumotga qarab muammoni topasiz?

O‘QUV MAQSADLARI

Bu ma’ruza oxirida siz quyidagilarni bajara olasiz…

1. **Tushuntirish** — unumdorlik ko‘rsatkichlari (p50/p95, so‘rovlar soni, xatolar ulushi) va kuzatuvchanlikning uch ustunini
2. **Aniqlash** — N+1 muammosini Debugbar yoki so‘rovlar logi yordamida
3. **Qo‘llash** — eager loading: `with()`, `load()`, `withCount()` va `Model::preventLazyLoading()`
4. **Loyihalash** — PostgreSQL uchun kompozit indekslarni va ularni `EXPLAIN ANALYZE` bilan tekshirishni
5. **Yozish** — Redis asosidagi kesh: `Cache::remember`, teglar va keshni bekor qilish (invalidation)
6. **Sozlash** — strukturali (JSON) loglash, so‘rov identifikatori va maxfiy maydonlarni yashirish
7. **Yaratish** — health check endpoint (`/up` va kengaytirilgan tekshiruvlar) va Sentry orqali xatolar monitoringini
8. **Baholash** — optimallashtirish natijasini “oldin/keyin” o‘lchovlari bilan

01

Unumdorlik va kuzatuvchanlik asoslari

Avval o‘lcha · p50 va p95 · loglar, metrikalar, treyslar · Debugbar va Telescope

p95 | Kuzatuvchanlik | Debugbar | Telescope

KUZATUVCHANLIK

Tizim ichini ko‘rish: uch ustun

Kuzatuvchanlik — “nega sekin?” va “nega xato?” degan savollarga ma’lumot bilan javob berish imkoniyati.

Loglar

Hodisa yozuvlari: kim, qachon, nima qildi. Masalan, `order.confirmed` + `order_id`, `user_id`.

Metrikalar

Vaqt bo‘yicha sonlar: RPS, p95 javob vaqti, xatolar ulushi, navbat uzunligi.

Treyslar

Bitta so‘rovning bosqichlari va har birining davomiyligi: controller → SQL → Redis → API.

Profillash

Dev muhitda bitta sahifani chuqur tahlil qilish: Debugbar, Telescope.

Monitoring “nima buzildi?”, kuzatuvchanlik “nega buzildi?” degan savolga javob beradi.

PERSENTILLAR

O‘rtacha emas, p95 ga qarang

```php
<?php
function persentil(array $ms, int $p): int
{
    sort($ms);
    $rank = (int) ceil($p / 100 * count($ms));
    return $ms[max($rank, 1) - 1];
}

$vaqtlar = [120, 95, 110, 2300, 130, 105, 98, 140, 115, 101,
    125, 99, 135, 1800, 108, 112, 97, 118, 122, 103];

printf("O‘rtacha: %d ms\n", array_sum($vaqtlar) / count($vaqtlar));
printf("p50: %d ms\n", persentil($vaqtlar, 50));
printf("p95: %d ms\n", persentil($vaqtlar, 95));
printf("p99: %d ms\n", persentil($vaqtlar, 99));
```

▶ NATIJA

```
O‘rtacha: 306 ms
p50: 112 ms
p95: 1800 ms
p99: 2300 ms
```

`sort($ms)`

persentil tartiblangan qatorda hisoblanadi

`ceil($p / 100 * n)`

“eng yaqin rang” (nearest-rank) usuli: 20 ta qiymatda p95 — 19-o‘rin

O‘rtacha 306 ms

“yaxshi”dek, lekin har 10-foydalanuvchi ~2 s kutyapti

`p95: 1800 ms`

SLO va sekin so‘rovlarni qidirish aynan shu raqamdan boshlanadi

PROFILERLAR

Debugbar va Telescope — faqat dev uchun

```
# sahifa pastidagi panel: SQL, vaqt, xotira, modellar
composer require barryvdh/laravel-debugbar --dev

# alohida /telescope sahifasi: so'rovlar, job, kesh, log
composer require laravel/telescope --dev
php artisan telescope:install
php artisan migrate

# .env (faqat lokal!)
# APP_DEBUG=true  DEBUGBAR_ENABLED=true  TELESCOPE_ENABLED=true
```

▶ NATIJA

```
Debugbar → Queries: 103 statements, 100 duplicates
Debugbar → Timeline: Application 2.41 s
Debugbar → Models: Order 50, Counterparty 50, OrderItem 150
```

`--dev`

paket faqat `require-dev` ga yoziladi; `composer install --no-dev` uni o‘rnatmaydi

Queries

so‘rovlar soni va takroriy (duplicate) so‘rovlar — N+1 belgisi

Models

qancha Eloquent model yuklangani: xotira sarfini ko‘rsatadi

Production

`APP_DEBUG=false`; profiler paneli hech qachon ochiq bo‘lmaydi

Productionda: strukturali loglar, Sentry, Laravel Pulse (sekin so‘rovlar va endpointlar paneli).

TUSHUNISHNI TEKSHIRISH

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

1-bo‘lim yakuni: Unumdorlik va kuzatuvchanlik asoslari

1

Nima uchun SLO o‘rtacha vaqt bilan emas, p95 bilan yoziladi?

Savol 1

2

Kuzatuvchanlikning uch ustunini sanang va har biriga bittadan misol keltiring.

Savol 2

3

Debugbar productionda yoqilib qolsa, qanday xavf tug‘iladi?

Savol 3

02

N+1 muammosi va eager loading

Lazy loading · with() va load() · withCount() · preventLazyLoading()

N+1 | with() | withCount() | preventLazyLoading

N+1 SIMULYATSIYASI

51 ta so‘rov yoki 2 ta so‘rov

```php
<?php
function sorovlar(int $n, bool $eager): array
{
    $log = ["select * from orders limit $n"];
    if ($eager) {
        $ids = implode(', ', range(1, $n));
        $log[] = "select * from counterparties where id in ($ids)";
        return $log;
    }
    foreach (range(1, $n) as $id) {
        $log[] = "select * from counterparties where id = $id";
    }
    return $log;
}
printf("Lazy (n=50): %d ta so‘rov\n", count(sorovlar(50, false)));
printf("Eager (n=50): %d ta so‘rov\n", count(sorovlar(50, true)));
$s = sorovlar(3, true);
echo $s[1], "\n";
```

▶ NATIJA

```
Lazy (n=50): 51 ta so‘rov
Eager (n=50): 2 ta so‘rov
select * from counterparties where id in (1, 2, 3)
```

1 + N

ro‘yxat uchun 1 ta, har bir qator uchun yana 1 ta so‘rov

where id in (...)

eager loading barcha kalitlarni bitta so‘rovga yig‘adi

n o‘sadi

lazy — chiziqli o‘sadi, eager — o‘zgarmas (munosabatlar soniga teng)

EAGER LOADING

with(), load(), withCount() — OrderController

```php
public function index(): View
{
    $orders = Order::query()
        ->with([
            'counterparty:id,name,inn',
            'warehouse:id,code',
            'items.product:id,sku,name',
        ])
        ->withCount('items')
        ->withSum('items', 'total')
        ->latest()
        ->paginate(50);

    return view('orders.index', compact('orders'));
}

// model allaqachon bor bo'lsa:
// $order->loadMissing('items.product');
```

▶ NATIJA

```
select * from orders order by created_at desc limit 50 offset 0
select id, name, inn from counterparties where id in (3, 7, 12, ...)
select id, code from warehouses where id in (1, 2)
select * from order_items where order_id in (101, 102, ...)
select id, sku, name from products where id in (5, 9, 14, ...)
```

`'counterparty:id,name,inn'`

faqat kerakli ustunlar; `id` albatta bo‘lsin

`'items.product'`

ichma-ich munosabat: avval `order_items`, keyin `products`

`withCount('items')`

`items_count` atributi — subquery orqali, alohida so‘rovsiz

`withSum('items', 'total')`

`items_sum_total` — yig‘indi ham shu so‘rovda

Natija: 101+ so‘rov o‘rniga 5 ta (sahifalashdagi `count(*)` bilan 6 ta).

HIMOYA

Lazy loading’ni dev muhitda taqiqlash

```php
// app/Providers/AppServiceProvider.php
public function boot(): void
{
    // lazy loading -> LazyLoadingViolationException
    Model::preventLazyLoading();

    // productionda sahifa buzilmasin: istisno o'rniga log
    if ($this->app->isProduction()) {
        Model::handleLazyLoadingViolationUsing(
            function (Model $model, string $relation): void {
                Log::warning('lazy_loading', [
                    'model' => $model::class,
                    'relation' => $relation,
                ]);
            });
    }
}
```

▶ NATIJA

```
Illuminate\Database\LazyLoadingViolationException
Attempted to lazy load [counterparty] on model
[App\Models\Order] but lazy loading is disabled.
```

`preventLazyLoading()`

lazy loading — istisno: dev va testda xato darhol ko‘rinadi

`handleLazyLoadingViolationUsing`

istisno o‘rniga o‘z ishlovchingiz: productionda faqat logga yozish

`isProduction()`

`APP_ENV=production` bo‘lsa `true`

`Model::shouldBeStrict()`

qo‘shimcha: yo‘q atribut va `$fillable` dan tashqari maydonni ham taqiqlaydi

Eng sodda variant: `Model::preventLazyLoading(! $this->app->isProduction());` — productionda tekshiruv umuman o‘chadi.

▶ INTERAKTIV

Nechta SQL so‘rov?

Har bir holatda sahifa nechta so‘rov yuborishini hisoblang (sahifalash `count` so‘rovini hisobga olmang). (Interaktiv slaydlarda bosib bajariladi.)

1 **20 ta buyurtma, Blade’da `$order->counterparty->name` chiqariladi, `with()` yo‘q.**

```php
$orders = Order::latest()->take(20)->get();
```

A) 2
B) 20
C) 21
D) 40

2 **Xuddi shu holat, lekin eager loading bilan.**

```php
$orders = Order::with('counterparty')
    ->latest()->take(20)->get();
```

A) 1
B) 2
C) 3
D) 21

3 **20 ta buyurtma va har birida `$order->items()->count()` chaqiriladi.**

```php
$orders = Order::with('items')->take(20)->get();
```

A) 2
B) 21
C) 22
D) 41

4 **`with('items.product')` bilan 20 ta buyurtma (jami 60 ta pozitsiya) yuklanadi.**

```php
$orders = Order::with('items.product')->take(20)->get();
```

A) 3
B) 21
C) 61
D) 81

TUSHUNISHNI TEKSHIRISH

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

2-bo‘lim yakuni: N+1 muammosi va eager loading

1

N+1 muammosi nima va u nega lokal muhitda sezilmaydi?

Savol 1

2

`with()` va `load()` ning farqi nimada?

Savol 2

3

Sikl ichida `$order->items()->count()` o‘rniga nima yozish kerak?

Savol 3

MA’RUZADAGI MASHQ

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

1-topshiriq: N+1: nechta SQL so‘rov?

Sahifada `$n` ta buyurtma chiqariladi (1 ta asosiy so‘rov). `$ishlatiladi` — har bir buyurtma uchun murojaat qilinadigan munosabatlar (har biri `belongsTo`), `$eager` — `with()` ga berilgan munosabatlar. Eager yuklangan har bir munosabat (ishlatilmasa ham) `$n > 0` bo‘lsa 1 ta so‘rov qo‘shadi, `$n = 0` bo‘lsa — 0 ta (Eloquent bo‘sh natija uchun eager so‘rov yubormaydi). Eager yuklanmagan, lekin ishlatilgan har bir munosabat `$n` ta so‘rov qo‘shadi (lazy loading). Takrorlangan nomlar bir marta hisoblanadi. `$n < 0` bo‘lsa — `InvalidArgumentException`.

Imzo: `function sorovlar_soni(int $n, array $ishlatiladi, array $eager): int`

Qaytaradi: `int` — sahifa yuboradigan SQL so‘rovlar soni.

MISOL · CHAQIRUV	MISOL · NATIJA
`sorovlar_soni(50, ['counterparty', 'warehouse'], [])`	`101`

Telefon kamerasi bilan skanerlang

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

03

Indekslar va so‘rov rejasi

B-tree · kompozit indeks va chap prefiks · EXPLAIN ANALYZE · indeks ishlamaydigan holatlar

Indeks | Kompozit | EXPLAIN ANALYZE | PostgreSQL

Indeks — tartiblangan “mundarija”

Indeks bazaga kerakli qatorlarni butun jadvalni o‘qimasdan topish imkonini beradi, lekin har bir yozishga qo‘shimcha narx qo‘shadi.

B-tree

Standart tur: `=`, `in`, `>`, `between`, `order by`, `like 'UN-%'` uchun.

Kompozit

Bir nechta ustun: `(status, created_at)`. Tartib muhim — chap prefiks qoidasi.

Unique

`sku`, `(warehouse_id, product_id)` — cheklov va indeks bir vaqtda.

Narxi

Har bir `insert`/`update` indekslarni ham yangilaydi; disk joyi ketadi.

PostgreSQL tashqi kalit ustunlariga indeksni avtomatik yaratmaydi — `counterparty_id`, `order_id` ga o‘zingiz qo‘shing.

MIGRATSIYA

OmborPro uchun indekslar

```php
return new class extends Migration
{
    public function up(): void
    {
        Schema::table('orders', function (Blueprint $table) {
            $table->index(['status', 'created_at']);
            $table->index('counterparty_id');
        });
        Schema::table('stock_movements', function (Blueprint $t) {
            $t->index(['product_id', 'created_at']);
        });
        Schema::table('invoices', function (Blueprint $table) {
            $table->index(['status', 'due_at']);
        });
    }
};
```

▶ NATIJA

```
INFO  Running migrations.
2026_10_27_100000_add_performance_indexes ........ 184ms DONE

-- yaratilgan indekslar: orders_status_created_at_index,
-- orders_counterparty_id_index, invoices_status_due_at_index ...
```

`['status', 'created_at']`

avval tenglik ustuni, keyin oraliq/tartiblash ustuni

`index('counterparty_id')`

FK ustuni: `join` va eager loading `in (...)` uchun

`['status', 'due_at']`

“muddati o‘tgan to‘lanmagan hisob-fakturalar” hisoboti

Nomlash

Laravel nomi: `jadval_ustun1_ustun2_index`

Katta jadvalda indeks yaratish yozishni bloklashi mumkin; productionda bu masala 14-mavzuda ko‘riladi.

EXPLAIN ANALYZE

Seq Scan → Index Scan

```
EXPLAIN (ANALYZE, COSTS OFF)
SELECT id, number, total
FROM orders
WHERE status = 'confirmed'
  AND created_at >= '2026-10-01'
ORDER BY created_at DESC
LIMIT 50;
```

▶ NATIJA

```
-- indeks yo'q (200 000 qator):
Seq Scan on orders (actual time=0.02..39.5 rows=4210 loops=1)
  Rows Removed by Filter: 195790
Execution Time: 41.9 ms
-- (status, created_at) indeksi bilan:
Index Scan Backward using orders_status_created_at_index on orders
Execution Time: 0.2 ms
```

ANALYZE

so‘rov haqiqatan bajariladi — `update/delete` uchun ehtiyot bo‘ling

Rows Removed by Filter

keraksiz o‘qilgan qatorlar — indeks kerakligining belgisi

Index Scan Backward

`order by created_at desc` ham indeks tartibidan olinadi, `Sort` yo‘q

Execution Time

“oldin/keyin” hisobotiga yoziladigan raqam

Laravel’da: `Order::where(...)->explain()` yoki `->toRawSql()` bilan SQL’ni olib psql’da ishga tushiring.

Indeks bor, lekin ishlamaydi

Holat	Nega ishlamaydi	Yechim
`whereRaw('date(created_at) = ?')`	ustunga funksiya qo‘llangan	`whereBetween('created_at', [$kun, $ertasi])`
`like '%001'`	boshida `%` — tartib foydasiz	prefiks qidiruv `like 'UN-%'` yoki to‘liq matnli qidiruv
Indeks `(status, created_at)`, so‘rov faqat `created_at` bo‘yicha	chap prefiks yo‘q	alohida `created_at` indeksi yoki tartibni o‘zgartirish
`where is_active = true` **(95 % qator)**	selektivlik past — Seq Scan arzonroq	indeks qo‘ymang yoki qisman indeks
`where inn = 123456789` **(ustun `varchar`)**	tur mos emas — almashtirish kerak	qiymatni satr sifatida bering

Qisman indeks: `CREATE INDEX ... ON invoices (due_at) WHERE status <> 'paid'` — migratsiyada `DB::statement()` bilan.

TUSHUNISHNI TEKSHIRISH

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

3-bo‘lim yakuni: Indekslar va so‘rov rejasi

1

`(status, created_at)` indeksi `where created_at >= ?` so‘roviga nega yordam bermaydi?

Savol 1

2

`EXPLAIN ANALYZE` natijasida qaysi qatorlar indeks yetishmasligini ko‘rsatadi?

Savol 2

3

PostgreSQL’da `foreignId()->constrained()` indeks yaratadimi?

Savol 3

MA’RUZADAGI MASHQ

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

2-topshiriq: Kompozit indeksning chap prefiksi

Kompozit B-tree indeksi ustunlari `$indeks` (tartib bilan) va so‘rov shartlari `$shartlar` (`ustun => operator`) berilgan. Indeks ustunlarini chapdan o‘ngga ko‘rib chiqing: ustun shartlarda bo‘lmasa — to‘xtang; operator tenglik (`=`, `in`) bo‘lsa — ustun hisoblanadi va davom etiladi; oraliq (`>`, `<`, `>=`, `<=`, `between`) bo‘lsa — ustun hisoblanadi va to‘xtatiladi. Shartlarda boshqa operator bo‘lsa (masalan, `like`) — `InvalidArgumentException`.

Imzo: `function indeks_ustunlari(array $indeks, array $shartlar): int`

Qaytaradi: `int` — so‘rovda indeksdan foydalaniladigan ustunlar soni (0 dan indeks uzunligigacha).

MISOL · CHAQIRUV

```
indeks_ustunlari(['status', 'created_at'], ['status' => '=',
'created_at' => '>='])
```

MISOL · NATIJA

```
2
```

Telefon kamerasi bilan skanerlang

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

04

Redis bilan keshlash

Cache::remember · TTL va kalitlar · teglar · keshni bekor qilish · stampede

Redis | Cache::remember | Teglar | Invalidation

CACHE-ASIDE

remember(): HIT va MISS

```php
<?php
function remember(array &$c, string $k, int $ttl, int $now,
    callable $fn): array
{
    if (isset($c[$k]) && $c[$k]['exp'] > $now) {
        return [$c[$k]['v'], 'HIT'];
    }
    $c[$k] = ['v' => $fn(), 'exp' => $now + $ttl];
    return [$c[$k]['v'], 'MISS'];
}
$cache = [];
$hisobot = fn () => array_sum([1200000, 850000, 430000]);
foreach ([0, 100, 599, 600, 700] as $t) {
    [$v, $h] = remember($cache, 'report:sales:2026-10', 600, $t,
        $hisobot);
    echo "t={$t}s  {$h}  {$v}\n";
}
```

▶ NATIJA

```
t=0s  MISS  2480000
t=100s  HIT  2480000
t=599s  HIT  2480000
t=600s  MISS  2480000
t=700s  HIT  2480000
```

`isset && exp > $now`

kalit bor va muddati tugamagan — HIT, hisoblash yo‘q

`$now + $ttl`

TTL = 600 s: 600-soniyada kalit eskiradi va qayta hisoblanadi

`Cache::remember`

Laravel’da xuddi shu mantiq bitta chaqiruvda

REPORTSERVICE

Hisobotni teglar bilan keshlash

```php
public function monthlySales(string $month, int $whId): array
{
    $key = "report:sales:v1:{$month}:wh={$whId}";
    $tags = ['reports', "warehouse:{$whId}"];

    return Cache::tags($tags)->remember($key, now()->addMinutes(10),
        fn () => Order::query()
            ->where('warehouse_id', $whId)
            ->where('status', OrderStatus::Shipped)
            ->whereBetween('created_at', $this->monthRange($month))
            ->selectRaw('date(created_at) as day, sum(total) as sum')
            ->groupBy('day')
            ->pluck('sum', 'day')
            ->all());
}
```

▶ NATIJA

```
1-chaqiruv:  MISS → 3 ta SQL, 640 ms
2-chaqiruv:  HIT  → 0 ta SQL, 2 ms (Redis GET)
```

$key

oy, ombor va format versiyasi — natijaga ta’sir qiluvchi hamma narsa

tags([...])

keyin `reports` yoki bitta ombor bo‘yicha tozalash mumkin

now()->addMinutes(10)

TTL — xavfsizlik to‘ri: invalidation unutilsa ham 10 daqiqada yangilanadi

->all()

keshga kolleksiya yoki model emas, oddiy massiv saqlanadi

Teglar `file` va `database` drayverlarida ishlamaydi: `.env` da `CACHE_STORE=redis`.

INVALIDATION

Ma’lumot o‘zgarsa — keshni tozalash

```php
// app/Listeners/FlushReportCache.php
class FlushReportCache
{
    public function handle(OrderConfirmed $event): void
    {
        $whId = $event->order->warehouse_id;
        Cache::tags(["warehouse:{$whId}"])->flush();
    }
}

// stampede himoyasi: bir vaqtda faqat bitta jarayon hisoblaydi
$data = Cache::lock("lock:report:{$month}", 10)
    ->block(5, fn () => $reports->monthlySales($month, 1));

// .env
// CACHE_STORE=redis  REDIS_HOST=redis  REDIS_CLIENT=phpredis
```

▶ NATIJA

```
OrderConfirmed(BY-2026-000123, warehouse_id=1)
→ Cache::tags(['warehouse:1'])->flush()
→ keyingi hisobot so'rovi: MISS, yangi raqamlar
```

OrderConfirmed

10-mavzudagi hodisa: listener uni avtomatik eshitadi

`tags([...])->flush()`

faqat shu omborning hisobotlari tozalanadi, boshqalari keshda qoladi

`Cache::lock()->block(5, ...)`

qulf 10 s; boshqa jarayonlar 5 s gacha kutadi

Qoida: har bir keshlangan qiymat uchun “uni nima eskirtiradi?” degan savolga javob yozilgan bo‘lsin.

▶ INTERAKTIV

Keshlash kerakmi?

Har bir ma’lumotni to‘g‘ri qutiga joylashtiring. (Interaktiv slaydlarda bosib bajariladi.)

Oylik savdo hisoboti (10 daqiqa eskirishi mumkin)

Markaziy bank USD kursi bugungi sana uchun

Kategoriyalar daraxti menyusi

Rezerv qilishdan oldin mavjud qoldiq

Hisob-faktura to‘lov holati (webhookdan keyin)

Dashboard: bugungi buyurtmalar soni

Foydalanuvchi rollari va ruxsatlari (kalitda user_id yo‘q)

Login paytidagi parol tekshiruvi

Keshlash mumkin

Keshlamang — har doim bazadan

TUSHUNISHNI TEKSHIRISH

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

4-bo‘lim yakuni: Redis bilan keshlash

1

Nega kesh kaliti barcha parametrlarni o‘z ichiga olishi shart?

Savol 1

2

Laravel kesh teglari qaysi drayverlarda ishlamaydi?

Savol 2

3

Cache stampede nima va undan qanday himoyalanamiz?

Savol 3

MA’RUZADAGI MASHQ

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

3-topshiriq: Hisobot uchun kesh kaliti

Hisobot nomi `$hisobot` va parametrlar `$params` dan barqaror kesh kalitini yasang: `omborpro:report:<hisobot>:<qism>`. Parametrlar kalit bo‘yicha alifbo tartibida (`ksort`, `SORT_STRING`), `null` qiymatlilar tashlab yuboriladi, `true`/`false` → `1`/`0`, massiv qiymat o‘sish tartibida saralanib vergul bilan qo‘shiladi; juftliklar `kalit=qiymat` ko‘rinishida `&` bilan birlashtiriladi. Parametr qolmasa, qism — `all`. Hisobot nomi `^[a-z][a-z_]*$` ga mos kelmasa — `InvalidArgumentException`.

Imzo: `function kesh_kaliti(string $hisobot, array $params): string`

Qaytaradi: `string` — masalan, `omborpro:report:sales:from=2026-10-01&to=2026-10-31&warehouse_id=3`.

MISOL · CHAQIRUV

```
kesh_kaliti('sales', ['to' => '2026-10-31', 'from' => '2026-10-01', 'warehouse_id' => 3])
```

MISOL · NATIJA

```
"omborpro:report:sales:from=2026-10-01&to=2026-10-31&warehouse_id=3"
```

Telefon kamerasi bilan skanerlang

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

05

Strukturali loglash

Log darajalari · kontekst · JSON kanal · request_id · maxfiy ma’lumotlar

Log | JSON | Context | request_id

KONTEKST

Log + kontekst va request_id

```php
// app/Http/Middleware/AssignRequestId.php
public function handle(Request $request, Closure $next): Response
{
    $id = $request->header('X-Request-Id') ?: (string) Str::uuid();
    Context::add('request_id', $id);
    $response = $next($request);
    $response->headers->set('X-Request-Id', $id);
    return $response;
}

// app/Services/OrderService.php
public function logConfirmed(Order $order): void
{
    Log::info('order.confirmed', [
        'order_id' => $order->id, 'number' => $order->number,
        'total' => $order->total, 'user_id' => auth()->id(),
    ]);
}
```

▶ NATIJA

```
[2026-10-27 10:15:02] production.INFO: order.confirmed
{"order_id":123,"number":"BY-2026-000123","total":"1500000.00",
"user_id":7} {"request_id":"9b2f6c1e-..."}
```

'order.confirmed'

o‘zgarmas xabar — sanash va qidirish oson

[...] kontekst

o‘zgaruvchan qiymatlar alohida maydonlarda, matn ichida emas

Context::add

Laravel 12: shu so‘rovdagi barcha loglarga va job’larga qo‘shiladi

X-Request-Id

javob sarlavhasida qaytariladi — foydalanuvchi xabar berganda shu ID bilan qidiramiz

JSON KANAL

config/logging.php: JSON formatdagi kanal

```php
use Monolog\Formatter\JsonFormatter;
use Monolog\Handler\StreamHandler;

return [
    'default' => env('LOG_CHANNEL', 'stack'),
    'channels' => [
        'stack' => ['driver' => 'stack',
            'channels' => explode(',', env('LOG_STACK', 'single'))],
        'json' => [
            'driver' => 'monolog',
            'level' => env('LOG_LEVEL', 'info'),
            'handler' => StreamHandler::class,
            'handler_with' => ['stream' => 'php://stderr'],
            'formatter' => JsonFormatter::class,
        ],
        // single, daily, stderr ... standart kanallar
    ],
];
```

▶ NATIJA

```
{"message":"order.confirmed","context":{"order_id":123,
"total":"1500000.00"},"level":200,"level_name":"INFO",
"channel":"production","datetime":"2026-10-
27T10:15:02.123456+05:00",
"extra":{"request_id":"9b2f6c1e-..."}}
```

`'driver' => 'monolog'`

istalgan Monolog handler va formatter bilan kanal

JsonFormatter

har bir yozuv — bitta qator JSON: Loki/ELK uchun tayyor

`php://stderr`

Docker’da loglar konteyner chiqishiga yoziladi

`LOG_STACK=daily,json`

bir vaqtda kunlik fayl va JSON oqimi

`.env`: `LOG_CHANNEL=stack`, `LOG_STACK=daily,json`, `LOG_LEVEL=info`. Natijadagi `level: 200` — Monolog’da INFO.

Logga yozishdan oldin niqoblash

```php
<?php
function logLine(string $level, string $msg, array $ctx): string
{
    $secret = ['password', 'token', 'card_number'];
    foreach ($ctx as $k => $v) {
        if (in_array($k, $secret, true)) {
            $ctx[$k] = '***';
        } elseif ($k === 'phone') {
            $ctx[$k] = substr($v, 0, 6) . '*****' . substr($v, -2);
        }
    }
    return json_encode(['level' => strtoupper($level),
        'message' => $msg, 'context' => $ctx]);
}
echo logLine('warning', 'payment.failed', [
    'invoice_id' => 45, 'method' => 'payme',
    'token' => 'tok_abc123', 'phone' => '+998901234567',
]), "\n";
```

▶ NATIJA

```
{"level":"WARNING","message":"payment.failed","context":{"invoice_
id":45,"method":"payme","token":"*","phone":"+99890***67"}}
```

`$secret`

parol va tokenlar to‘liq yashiriladi

`phone`

shaxsiy ma’lumot qisman niqoblanadi — qidirish uchun yetarli, oshkor emas

`json_encode`

bitta qator JSON — log tizimi maydonlarni ajratib oladi

Laravel’da buni Monolog processor yoki kanalning `tap` sozlamasi orqali barcha loglarga qo‘llash mumkin.

TUSHUNISHNI TEKSHIRISH

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

5-bo‘lim yakuni: Strukturali loglash

1

Strukturali logning oddiy matnli logdan afzalligi nimada?

Savol 1

2

Productionda `LOG_LEVEL=debug` qo‘yish nega yomon?

Savol 2

3

Logga yozilmasligi kerak bo‘lgan 3 ta ma’lumotni ayting.

Savol 3

06

Health check va Sentry

/up marshruti · DiagnosingHealth · kengaytirilgan tekshiruvlar · Sentry bilan xatolar monitoringi

/up | Health check | Sentry | Monitoring

LARAVEL 11

/up marshruti va o‘z tekshiruvlarimiz

```php
// bootstrap/app.php
return Application::configure(basePath: dirname(__DIR__))
    ->withRouting(
        web: __DIR__.'/../routes/web.php',
        api: __DIR__.'/../routes/api.php',
        commands: __DIR__.'/../routes/console.php',
        health: '/up',
    )
    // ...
    ->create();

// AppServiceProvider::boot()
Event::listen(function (DiagnosingHealth $event) {
    DB::select('select 1');
    Cache::store('redis')->put('health:ping', now()->timestamp, 60);
});
```

▶ NATIJA

```
$ curl -s -o /dev/null -w "%{http_code}\n"
https://omborpro.test/up
200
# PostgreSQL to'xtatilganda:
500
```

`health: '/up'`

tayyor marshrut: ilova yuklansa 200, istisno bo‘lsa 500

DiagnosingHealth

`/up` chaqirilganda yuboriladigan hodisa

`Event::listen(fn (DiagnosingHealth $e))`

turi bo‘yicha hodisani aniqlaydi; istisno → 500

`/up` — Docker `healthcheck` va uptime monitor uchun yengil tekshiruv; og‘ir so‘rovlar qo‘ymang.

KENGAYTIRILGAN

/health: JSON javob va 503

```php
Route::get('/health', function () {
    $check = function (callable $fn): array {
        $t = microtime(true);
        $ok = rescue(fn () => $fn() !== false, false, false);
        $ms = (int) round((microtime(true) - $t) * 1000);
        return ['ok' => $ok, 'ms' => $ms];
    };
    $checks = [
        'database' => $check(fn () => DB::select('select 1')),
        'redis' => $check(fn () => Redis::ping()),
        'queue' => $check(fn () => Queue::size('default') < 1000),
    ];
    $fail = ! $checks['database']['ok'] || ! $checks['redis']['ok'];
    $all = ! in_array(false, array_column($checks, 'ok'), true);
    $status = $fail ? 'fail' : ($all ? 'ok' : 'degraded');
    $code = $fail ? 503 : 200;
    return response()->json(compact('status', 'checks'), $code);
});
```

▶ NATIJA

```
HTTP/1.1 200 OK
{"status":"degraded","checks":{"database":{"ok":true,"ms":3},
"redis":{"ok":true,"ms":1},"queue":{"ok":false,"ms":2}}}
```

`rescue($fn, false, false)`

istisnoni ushlab `false` qaytaradi, `report` qilmaydi

database, redis

muhim tekshiruvlar: ishlamasa — `fail` va 503

queue

navbatda 1000+ vazifa — `degraded`: ishlayapti, lekin ogohlantirish

Marshrutga `->middleware('throttle:30,1')` qo‘shib, uni ortiqcha chaqiruvlardan himoyalang.

HOLATNI HISOBLASH

ok · degraded · fail

```php
<?php
$checks = [
    'database' => ['ok' => true, 'critical' => true],
    'redis' => ['ok' => true, 'critical' => true],
    'queue' => ['ok' => false, 'critical' => false],
];
$failed = array_keys(array_filter($checks, fn ($c) => !$c['ok']));
$critical = array_filter($failed, fn ($n) => $checks[$n]['critical']);
$status = match (true) {
    $critical !== [] => 'fail',
    $failed !== [] => 'degraded',
    default => 'ok',
};
$http = $status === 'fail' ? 503 : 200;
echo "HTTP $http\n";
echo json_encode(['status' => $status, 'failed' => $failed]), "\n";
```

▶ NATIJA

```
HTTP 200
{"status":"degraded","failed":["queue"]}
```

critical

qaysi qism ishlamasa butun tizim “ishlamaydi” deb hisoblanadi

match (true)

eng jiddiy holat birinchi tekshiriladi

503 / 200

`degraded` da trafik to‘xtatilmaydi, faqat ogohlantirish beriladi

SENTRY

Xatolar monitoringi: o‘rnatish va sozlash

```php
// composer require sentry/sentry-laravel
// php artisan sentry:publish --dsn=https://KEY@sentry.example.uz/1

// bootstrap/app.php
use Illuminate\Foundation\Configuration\Exceptions;
use Sentry\Laravel\Integration;

return Application::configure(basePath: dirname(__DIR__))
    // ->withRouting(...)->withMiddleware(...)
    ->withExceptions(function (Exceptions $exceptions) {
        Integration::handles($exceptions);
    })
    ->create();

// .env
// SENTRY_LARAVEL_DSN=https://KEY@sentry.example.uz/1
// SENTRY_TRACES_SAMPLE_RATE=0.2
// SENTRY_SEND_DEFAULT_PII=false
```

▶ NATIJA

```
$ php artisan sentry:test
[Sentry] DSN discovered!
[Sentry] Sending test event...
[Sentry] Event sent with ID: 3f1c...e9
```

`Integration::handles`

Laravel 12: ushlanmagan istisnolar Sentry’ga yuboriladi

`TRACES_SAMPLE_RATE=0.2`

so‘rovlarning 20 % i uchun unumdorlik treysi

`SEND_DEFAULT_PII=false`

IP, cookie, foydalanuvchi ma’lumotlari yuborilmaydi

`sentry.example.uz`

self-hosted Sentry yoki GlitchTip — ma’lumot mamlakat ichida qoladi

DSN — maxfiy sozlama: faqat `.env` da, repozitoriyga yozilmaydi.

TUSHUNISHNI TEKSHIRISH

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

6-bo‘lim yakuni: Health check va Sentry

1

Liveness va readiness tekshiruvlarining farqi nimada?

Savol 1

2

Laravel 12 dagi `/up` marshrutiga o‘z tekshiruvingizni qanday qo‘shasiz?

Savol 2

3

Sentry’ni moliya tizimiga ulashda qaysi sozlamalarga e’tibor berasiz?

Savol 3

XULOSA

Asosiy xulosalar

1 Avval o‘lcha

Optimallashtirish p95, so‘rovlar soni va “oldin/keyin” o‘lchovlariga asoslanadi, taxminga emas.

2 N+1 → eager loading

`with()`, `load()`, `withCount()`; dev muhitda `Model::preventLazyLoading()`.

3 Indeks — so‘rov uchun

Kompozit indeksda tartib muhim (chap prefiks); natijani `EXPLAIN ANALYZE` bilan tekshiring.

4 Kesh + invalidation

To‘liq kalit, TTL, teglar va hodisa bo‘yicha `flush()`; teglar uchun Redis kerak.

5 Strukturali log

O‘zgarmas xabar + kontekst, JSON kanal, `request_id`; maxfiy ma’lumotlar yo‘q.

6 Health check va Sentry

`/up` + `DiagnosingHealth`, JSON `/health` (503), Sentry bilan xatolar monitoringi.

TAKRORLASH

Yakunlashdan oldin sakkizta savol

1. 50 ta buyurtma va ularning kontragenti `with()` siz chiqarilsa, nechta so‘rov ketadi?

2. `withCount('items')` qaysi atributni qo‘shadi?

3. `Model::preventLazyLoading()` qachon istisno tashlaydi?

4. `(status, created_at)` indeksi qaysi so‘rovga mos: `where status = ?` yoki `where created_at > ?`?

5. `EXPLAIN ANALYZE` dagi `Seq Scan` nimani anglatadi?

6. `Cache::tags()` ishlashi uchun qaysi drayver kerak?

7. Nima uchun log xabarini o‘zgarmas qilib, qiymatlarni kontekstga yozamiz?

8. Health endpoint baza ishlamaganda qaysi HTTP kodni qaytarishi kerak?

AMALIY MASHG‘ULOT

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

4 **Apdex: foydalanuvchi qoniqishi**
Oson · 10 ball

5 **Strukturali log yozuvi va niqoblash**
O‘rta · 20 ball

6 **Health check holatini hisoblash**
O‘rta · 20 ball

7 **Teglar va TTL bilan kesh**
Qiyin · 30 ball

8 **Takroriy so‘rovlarni aniqlash (N+1 detektori)**
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

O‘z loyihangizdagi eng sekin sahifa qaysi va uni o‘lchash uchun qaysi vositadan foydalanasiz?

2

`Cache::remember('report', 600, ...)` kalitida qanday xato bor va uni qanday tuzatasiz?

3

Productionda xato yuz berdi: Sentry, loglar va `request_id` yordamida sababni qanday topasiz?