

PAVEDIMO ATLIKIMAS

Pavedimui naudojam **Jobs**

1. Paleidžiam komandą PhpStorm'e: **php artisan queue:table** (sukurs darbų/užduočių eilės (queue) stulpelių aprašymą lentelei kurti).
2. Paleidžiam komandą lentelės sukūrimui: **php artisan migrate** (sukurs lentelę duomenų bazėje, kurioje saugosime užduotis, kurias paleis pagal nurodytą laiką)
3. Į **.env** failą įkeliame eilutę **QUEUE_DRIVER=database** (nurodom, koks draiveris bus naudojamas darbų eilėms vykdyti)
4. Paleidžiam komandą **php artisan queue:listen**, kuri seks, ar paleistas koks nors metodas **job** darbui atlikti. Jei pastebės, darbą įkels į lentelę ir jį atliks per nurodytą laiką. Dar naudojama komanda **php artisan queue:work** bet man ji nelabai pastebi metodą, prašantį darbą statyti į eilę.

Plačiau apie queue ir jobs

1. <https://www.larashout.com/laravel-queues-step-by-step-guide>
2. <https://upnrunn.com/blog/2018/09/how-to-set-up-job-queue-in-laravel/>
3. <https://www.nicesnippets.com/blog/laravel-76-queue-example-tutorial>

Tinklapio puslapio kelio nurodymas web.php

```
4 use App\Http\Controllers\HomeController;
5 use App\Http\Controllers\TransferController;
6
7 /*
8 |-----|
9 | Web Routes |
10 |-----|
11 | Here is where you can register web routes for your application. These
12 | routes are loaded by the RouteServiceProvider within a group which
13 | contains the "web" middleware group. Now create something great!
14 |
15 |*/
16 Route::resource('name: '/', controller: HomeController::class);
17 Route::get( uri: '/search', [HomeController::class, 'search']);
18 Route::get( uri: '/transfer', [HomeController::class, 'transfer']);
19 Route::post( uri: '/list', [HomeController::class, 'list']);
20 Route::post( uri: '/store', [TransferController::class, 'store']);
21 Route::get( uri: '/error', [TransferController::class, 'error']);
22 Route::get( uri: '/cancel/{id}', [TransferController::class, 'cancel']);
23
```

Nurodom, kokius kontrolerius naudosime šitam faile

Jei naršyklėje atsiranda nuoroda /transfer keliaujam į HomeController failą ir jame paleidžiame metodą transfer, kuris atidaro puslapio vaizdą (pavedimo formą)

Pirma sukuriu tinklapio puslapį (**transfer.blade.php**) pavedimams atlikti (pilna nuoroda pas mane <http://bank.test/transfer>).

Formą sukurti pagal mokytojo pateiktą šabloną. Išsikopinti ir susidėlioti laukėlius.

Jau turime puslapio vaizdą ir nuorodą route (vaizdas iš failo web.php)

```
Route::get(uri: '/transfer', [HomeController::class, 'transfer']);
```

Toliau kuriame kontrolerį, kuris paleis metodą transfer(). Šis metodas atidarys puslapio pavedimo formą (view).

php artisan make:controller HomeController

Valio! Turime kontrolerį (vaizdas iš failo web.php)

```
Route::get(uri: '/transfer', [HomeController::class, 'transfer']);
```

Route nurodome, kad nereiks perduoti jokių duomenų, o tik gauti vaizdą su transfer metodo pagalba (get).

```
Route::get(uri: '/transfer', [HomeController::class, 'transfer']);
```

Einame į failą HomeController ir jame parašysime metodą transfer() kuris ir parodys formą.

```
public function transfer(){  
    return view('pages.transfer');  
}
```

Turime metodą, kuris pateikia puslapio vaizdą naršyklei (vaizdas iš failo web.php)

```
Route::get(uri: '/transfer', [HomeController::class, 'transfer']);
```

Apibendrinimas: suvedame naršyklėje tinklapio pavadinimą su pabaiga /transfer. Galima padaryti meniu mygtukui nuorodą ``. Paspaudžiam ir kontrolieris su metodu transfer atidaro formos vaizdą (view).

Bankas

Profilis

Atlikti paiešką

Atlikti pavedimą

Atsijungti

Lėšų pavedimas

Pavedimo duomenys

Pasirinkite iš kurios sąskaitos atliksite pavedimą

Pasirinkite į kurią sąskaitą atliksite pavedimą

Suma

Pasirinkite...

Gavėjo vardas

Gavėjo pavardė

Pavedimo paskirtis

Vykdyti pavedimą

Copyright © 2021

Dabar reikia priversti formą įterpti duomenis į duomenų bazę, paspaudus mygtuką Vykdyti pavedimą.

`<form action="/store" method="post" >`

Paspaudus mygtuką atidarys tinklapio puslapį su nuoroda/store.

Naudosime metodą post, nes reikia perduoti formoje surašytus duomenis

Vėl einame į failą `web.php` ir nurodysime, koks metodas bus paleistas naršyklėje atsiradus nuorodai `/store`




```
Route::post('/store', [TransferController::class, 'store']);
```

Route nurodysime, kad formos duomenis perduodame, tad **post**

Sukuriu naują kontrolerį **TransferController**, nes operacija susijusi su pavedimu, tai patogiau bus rašyti kodą. Visi metodai, susiję su pavedimais, gali būti šiame kontroleryje.


php artisan make:controller TransferController

Turime kontrolerį, jame įrašysime, metodą **store**.



```
Route::post('/store', [TransferController::class, 'store']);
```

Kontroleryje įrašysime, metodą **store**. Jį paleis kontroleris paspaudus mygtuką formoje.



```
Route::post('/store', [TransferController::class, 'store']);
```

Einame į failą **TransferController** ir įrašome metodą **store()**

Šis metodas pasiims formoje suvestus duomenis per kintamąjį `$request`

Formoje suvesti duomenys perduoti su **post** metodu

```
public function store(Request $request) {  
  
    $ac = Account::where('account_no', '=', request('faccountto'))->first();  
    if(isset($ac)) {  
        $us = User::where('id', '=', $ac['user_id'])->first();  
    }else{  
        throw ValidationException::withMessages(['Įvesta klaidinga gavėjo sąskaita']);  
    }  
    $ac1 = Account::where('account_no', '=', request('faccountfrom'))->first();  
  
    //dd($na['surname']);  
  
    $validateData = $request->validate([
```

Per modelį Account pasiimu pirmą įrašą pagal formoje nurodytą sąskaitos numerį į kurį turi ateiti pinigai

jei modelis grąžino įrašą - nustatau, kam priklauso sąskaita. Juos paduosiu vėliau metodui SendMoney

Jei modelis nieko negrąžino, reiškia tokios sąskaitos nėra ir mesti pranešimą

Laukų validavimo taisyklės

Validuojant galima naudoti Rule::
Šiuo atveju patikrins ar accounts lentelėje, stulpelyje account_no yra sąskaitos numeris, kuris buvo įrašytas pavedimo formoje laukelyje vardu faccountto

Klaidų išvertimas į lietuvių kalbą

```
],  
[  
    'faccountfrom.required'=> 'Pasirinkite sąskaita',  
    'faccountfrom.string'=> 'Klaidinga sąskaita',  
    'faccountfrom.min'=> 'Klaida. Klaidinga sąskaita',  
    'faccountfrom.max'=> 'Klaida. Klaidinga sąskaita',  
    'faccountto.exists'=> 'Klaida. Tokia sąskaita neegzistuoja',  
    'faccountto.different'=> 'Klaida. Gavėjo sąskaita sutampa su pavedimo sąskaita',  
    'faccountto.required'=> 'Įveskite gavėjo sąskaitos numerį',  
    'surname.required' => 'Įveskite pavardę',
```

```

'surname.exists' => 'Neteisinga pavardė. Šiam vartotojui nepriklauso nurodyta sąskaita',
'name.required' => 'Įveskite vardą',
'name.exists' => 'Neteisingas vardas. Šiam vartotojui nepriklauso nurodyta sąskaita',
'purpose.required' => 'Įveskite pavedimo paskirtį',
'purpose.max' => 'Sutrumpinkite pavedimo paskirties tekstą',
'amount.required' => 'Įveskite pavedimo sumą'

]
);

if($ac1['balance']-$ac1['reserved']>=request('amount')) {
    Account::where('account_no', request('faccountfrom'))->update(['reserved' =>
    $ac1['reserved']+request('amount')]);
} else {
    throw ValidationException::withMessages(['Saskaitoje nepakanka lėšų']);
    // return redirect()->back()->withInput();
}

$tid=Transfer::create([
    'account_id_from' => self::filterAccNumber(request('faccountfrom'))->id, //
    'account_id_to'=> self::filterAccNumber(request('faccountto'))->id,
    'purpose'=>request('purpose'),
    'status'=>1,
    'amount'=> request('amount'),
    'date'=>now()->format('Y-m-d')
]);

$data=SendMoney::dispatch($ac, $ac1, request('faccountfrom'), request('faccountto'),
request('amount'), $tid->id )
->delay(120);

//dd($request);
// dd($ats->id);prepareForValidation
//return Redirect::back('/transfer')->withStatus('Mokejimas atliktas');
return redirect('/transfer')->with('message','Mokejimas atliktas');
}

```

Patikrinu ar sąskaitoje esanti suma atėmus norimą rezervuoti sumą yra pakankama. Jei pinigų neužtenka – metu klaidą

Jei validacija ok – formoje pateiktus duomenis įterpiu per modelį Transfer į duomenų bazę. T.y. pvedimas atliekamas, rezervuojama suma, pavedimo statusas 1, t.y. galima atšaukti

Supakuoju kintamuosius ir užklausų duomenis, kurie reikalingi įterpti įrašams į duomenų bazės lenteles. Užklausos bus vykdomos per **Job** klasę **SendMoney**. SendMoney nustatau laiko limitą 120 sekundžių. Ši klasė t.y. jos metodas handle() įterpiamas į jobs lentelę su **dispatch** ir bus įvykdytas po 120 sekundžių (užlaikymas - delay).

Anksčiau buvome paleidę komandą **php artisan queue:listen**. Ji pagauna dispatch ir stato į eilę metodą handle iš klasės SendMoney. Įvykčius SendMoney šis metodas iš jobs lentelės ištrinamas automatiškai, kaip įvykdytas.

Jei viskas ok, nukreips atgal į pavedimo formą ir perduos žinutę, kad pavedimas atliktas

Apibendrinimas:

1. Iš pavidimo formos per kintamąjį \$request pasiimu laukų reikšmes.
2. Atlieku laukų validaciją. Jei validacija bloga, formos viršuje išmeta klaidos pranešimus ir kodas toliau nevykdomas.
3. Jei validacija ok, tada įterpiu į pavidimų lentelę transfers naujo pavidimo duomenis: nustatau, kad status 1, kad matyčiau, jog pavidimą galima atšaukti. Įvykdžius SendMoney statusas bus nustatytas į 2. Statusas man reikalingas, jei gaunantis pinigų žmogus filtruos pavidimus tai nematys, jam skirto pavidimo, kol neįvykdytas SendMoney. O kas perveda matys ir galės atšaukti. Padariau pavidimo atšaukimą ir pavidimas ištrinamas, bet galima padaryti statusą 3. Taip pat įterpiu į accounts lentelės lauką rezervuojamą sumą pasirinktai sąskaitai. Jei bus įvykdytas sėkmingai SendMoney, rezervuota suma atimama iš stulpelio reserved, papildoma sąskaita pervedama pinigų suma.
4. Praėjus 120 sek. Jobs lentelėje saugomas SendMoney bus įvykdytas ir nebus galimybės atšaukti pavidimo.

Pavidimo atšaukimas

Atlieku pavidimų paiešką ir sąraše matau pavidimą kurį galiu atšaukti

Mygtukui turi būti nustatyta nuoroda `/cancel/{id}` Šio puslapio nekuriu (view), tiesiog bus įvykdomas metodas. id – tai pavidimo numeris, kad žinočiau kurį pavidimą atšaukti. Kontroleryje `Cancel()` metodo gale įrašysiu nukreipimą

```
return redirect('/');
```

į pagrindinį puslapį. Jei nepadarysiu nukreipimo į pagrindinį psl., tada bus tuščias langas. Galima padaryti nukreipimą atgal į pavidimų paiešką ir pan.

Tik paimu duomenis, tad metodus get

Kontroleris, kuris paleis metoą cancel

```
Route::get('/cancel/{id}', [TransferController::class, 'cancel']);
```

Tiklapio nuoroda, be vaizdo. Tik perduodamas pavidimo id

Pasiimu pavidimo {id} iš nuorodos, tik pavadinu account

Sukuriu cancel metodą.

```
public function cancel($account)
{
```

```
$jobs = Job::get();
```

Gaunu visus jobs iš jobs lentelės, kuriems nustatytas atlikimo laiko tarpas ir dar nėra įvykdyti

```
foreach ($jobs as $job){
    $aw = json_decode($job->payload)->data->command;
    $cm = unserialize($aw);
```

Jobs turi stulpelį payload ir jame saugomi dispatch perduoti duomenys. Tame tarpe ir mano sukurtas metodashandle iš SendMoney. Dekoduojau, kad pasiimti informaciją

```
if($cm->tid == $account){
    try { //jei įvyks klaida kurioj nors iš 4-ių užklausų, nebu
        DB::beginTransaction();
        DB::table('jobs')->whereId($job->id)->delete();
        DB::table('transfers')->whereId($account)->delete();
        $acc_reserved = DB::table('accounts')->where('account_no', $cm->faccountfrom)->first()-
```

Tid saugo perduotą pavidimo numerį ir sukdamas ciklą lyginu su atsineštu pavidimo numeriu, surandu jobs sąrašę reikiamą pavidimo numerį, jei jis yra, tada ištrinu

Try gaudo klaidas, jei kurioje užklausoje bus klaida, nė viena užklausa nevykdoma

```
>reserved;
        Account::where('account_no', $cm->faccountfrom)->update(['reserved' => $acc_reserved-
        $cm->amount]);
        DB::commit();
    } catch (\Exception $e) {
        DB::rollback();
    }
    //dar reik siust zinute pavyko
} else {
```

Rollback atšaukia kitas užklausas, jei nors vienoje įvyko klaida

Commit įvykdo užklausas, jei nė vienoje nebuvo klaidos

```

        //dar reik siust zinute nepavyko
    }
}
return redirect('/');
//https://stackoverflow.com/questions/40139208/how-do-i-nicely-decode-laravel-failed-jobs-json
// dd($cm->tid);
}

```

Jei viskas ok, nukreipiu į pagrindinį puslapį.

Apibendrinimas: jei jobs sąrašas rastas pavedimo id, vykdomos užklauso: trinamas įrašas iš jobs lentelės, kad nebūtų įvykdyta SendMoney metodas handle. Taip pat trinamas pavedimas iš transfers lentelės pagal jo id. Iš lentelės accounts atimama reserved suma.

Sukuriamas Jobs

```
php artisan make:job SendMoney
```

atidarome failą ir surasome klasei SendMoney kodą

```

SendMoney implements ShouldQueue
{
    use Dispatchable, InteractsWithQueue, Queueable, SerializesModels;
    public $timeout = 10;

    public $ac;
    public $acl;
    public $faccountfrom;
    public $faccountto;
    public $amount;
    public $tid;
    /**
     * Create a new job instance.
     *
     * @return void
     */
}

```

Konstruktoriui nurodau kokius duomenis perimu

```

public function __construct($ac, $acl, $faccountfrom, $faccountto, $amount, $tid)
{
    $this->ac=$ac;
    $this->acl=$acl;
    $this->faccountfrom=$faccountfrom;
    $this->faccountto=$faccountto;
    $this->amount=$amount;
    $this->tid=$tid;
}
/**
 * Execute the job.
 *
 * @return void
 */
public function handle()
{
    // $this->job->getJobId();
    Account::where('account_no', $this->faccountfrom)->update(['reserved' => $this->acl['reserved']-
    $this->amount]);
    Account::where('account_no', $this->faccountfrom)->update(['balance' => $this->acl['balance']-
    $this->amount]);
    Account::where('account_no', $this->faccountto)->update(['balance' => $this->ac['balance']+$this-
    >amount]);
    Transfer::where('id', $this->tid)->update(['status' => 2]);
    //Account::where('account_no', 'LT655597745445546503')->update(['balance' => $this->h]);
}
}

```

Atlieku trynimo ir update
komandas

Dekoduoju json decode

<https://stackoverflow.com/questions/40139208/how-do-i-nicely-decode-laravel-failed-jobs-json>

Komandų vykdymas po tam tikro skaitliuko rezultato

<https://divinglaravel.com/properties-of-a-scheduled-job-in-laravel>

Job komandos

<https://stackoverflow.com/questions/37067934/how-to-execute-laravel-job-queue>