

PEMBUATAN API SEDERAHANA MENGUNAKAN SERVICE REPOSITORY PATTERN

**Muflihun Abdul Firmansyah
Aji Karuniadi Irawan**

Install Package laravel-mongodb

Buka atau buat project Laravel, kemudian jalankan Terminal Visual Studio Code lalu ketik perintah

composer require jenssegers/mongodb versi-laravel

Untuk melihat versi MongoDB yang compatible dengan versi Laravel yang digunakan dapat diperiksa melalui: <https://github.com/jenssegers/laravel-mongodb>

Sebagai contoh Laravel menggunakan versi 8 maka MongoDB yang diinstal versi 3.8.x, sehingga perintah yang dijalankan pada terminal

composer require jenssegers/mongodb 3.8

```
(base) → inosoft-bootcamp-2 git:(part4_todo) ✖ composer require jenssegers/mongodb
```

Mengubah file config/app.php

Setelah itu buka file **./config/app.php** dan tambahkan baris code seperti yang ada pada gambar dibawah ini

```
<?php

return [
    ....
    'providers' => [
        ....
        Jenssegers\Mongodb\MongodbServiceProvider::class,
    ]
    ....
]
```

Mengubah file config/database.php

Lalu Kita tambahkan baris code dibawah ini didalam

```
'mongodb' => [  
    'driver'    => 'mongodb',  
    'host'      => env('MONGO_DB_HOST', 'localhost'),  
    'port'      => env('MONGO_DB_PORT', 27017),  
    'database'  => env('MONGO_DB_DATABASE'),  
    'username'  => env('MONGO_DB_USERNAME'),  
    'password'  => env('MONGO_DB_PASSWORD'),  
    'options'   => []  
],
```

Mengubah file .env

Lalu Kita tambahkan baris code dibawah di file .env untuk nama database bisa disesuaikan

```
MONGO_DB_HOST=127.0.0.1  
MONGO_DB_PORT=27017  
MONGO_DB_DATABASE=hddatabase  
MONGO_DB_USERNAME=  
MONGO_DB_PASSWORD=
```

Membuat Model Todo

Setelah konfigurasi selesai langkah selanjutnya adalah membuat sebuah model Todo nya, dengan memasukkan perintah

```
php artisan make:model nama-model
```

Sebagai contoh

```
php artisan make:model Todo
```

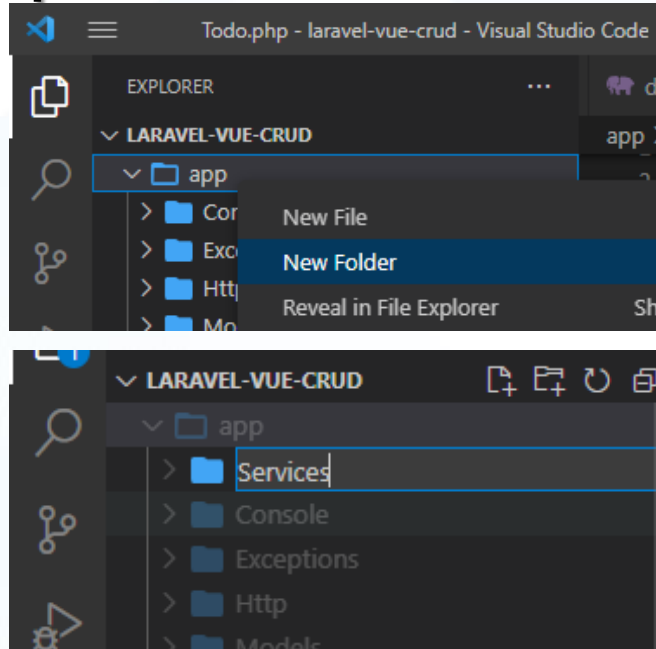
Kemudian menyesuaikan import package karena project ini akan menggunakan mongodb sebagai database

Edit file model Todo.php

Menambahkan sintaks ke file model yang telah dibuat
(./app/models/Todo.php)
seperti pada gambar disamping.

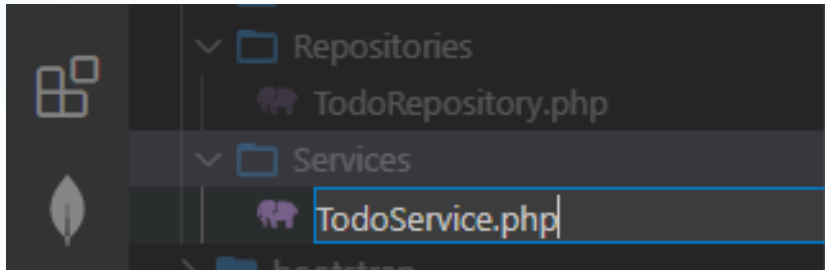
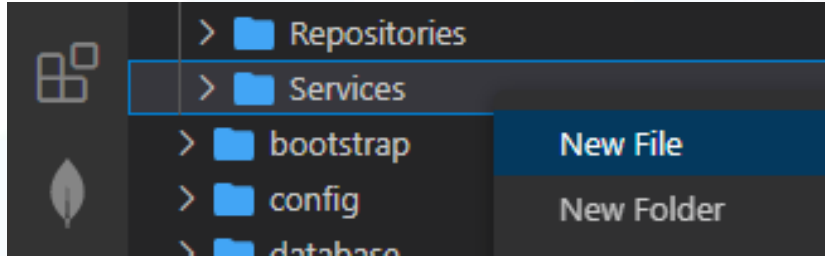
```
1  <?php
2
3  namespace App\Models;
4
5  use Jenssegers\Mongodb\Eloquent\Model;
6
7  class Todo extends Model
8  {
9      protected $connection = 'mongodb';
10     protected $collection = 'todo';
11
12
13     /**
14      * The attributes that are mass assignable.
15      *
16      * @var array
17      */
18     protected $fillable = [ 'title' ];
19 }
20
```

Membuat Folder Services dan Repositories



Menambahkan 2 folder baru pada folder app, Bernama **Services** dan **Repositories**. Klik kanan pada folder app kemudian pilih **New Folder**, lalu ketikkan nama folder baru yang akan dibuat.

Membuat File pada Folder Services dan Repositories



Menambahkan file baru pada folder masing-masing folder **Services** dan **Repositories**. Klik kanan pada folder yang akan ditambahkan file baru kemudian pilih **New File**, lalu ketikkan nama folder baru yang akan dibuat.

Pada folder **Services** beri nama file baru dengan nama **TodoService.php** dan folder **Repositories** beri nama file baru dengan nama **TodoRepository.php**

Edit file TodoRepository.php

Menambahkan sintaks ke file
TodoRepository.php seperti
pada gambar disamping.

```
1  <?php
2
3  namespace App\Repositories;
4
5  use App\Models\Todo;
6
7  class TodoRepository
8  {
9      protected $todo;
10     public function __construct(Todo $todo)
11     {
12         $this->todo = $todo;
13     }
14     public function getAll() : Object
15     {
16         $todo = $this->todo->get();
17         return $todo;
18     }
19     public function store($data) : Object
20     {
21         $dataBaru = new $this->todo;
22         $dataBaru->titile = $data['title'];
23         $dataBaru->save();
24         return $dataBaru->fresh();
25     }
26 }
```

Edit file TodoService.php

Menambahkan sintaks ke file
TodoService.php seperti
pada gambar disamping.

```
1  <?php
2
3  namespace App\Services;
4
5  use App\Repositories\TodoRepository;
6  use MongoDB\Exception\InvalidArgumentException;
7  use Illuminate\Support\Facades\Validator;
8
9  class TodoService {
10
11     protected $todoRepository;
12
13     public function __construct(TodoRepository $todoRepository)
14     {
15         $this->todoRepository = $todoRepository;
16     }
17     public function getAll()
18     {
19         $todo = $this->todoRepository->getAll();
20
21         return $todo;
22     }
23     public function store($data) : Object
24     {
25         $validator = Validator::make($data, [
26             'title' => 'required',
27
28         ]);
29
30         if ($validator->fails()) {
31             throw new InvalidArgumentException($validator->errors()->first());
32         }
33
34         $result = $this->todoRepository->store($data);
35         return $result;
36     }
37 }
38
39 }
```

```

1  class TodoController extends Controller
2  {
3
4      protected $todoService;
5
6      public function __construct(TodoService $todoService)
7      {
8          $this->todoService = $todoService;
9      }
10
11     public function getTodoList() {
12         try {
13             $result = $this->todoService->getAll();
14         } catch (Exception $e) {
15             $result = [
16                 'status' => 500,
17                 'error' => $e->getMessage()
18             ];
19         }
20
21         return response()->json($result);
22     }
23
24     public function addTodo(Request $request) {
25         $data = $request->only(['title']);
26
27         $result = ['status' => 201];
28
29         try {
30             $result['data'] = $this->todoService->store($data);
31         } catch (Exception $e) {
32             $result = [
33                 'status' => '422',
34                 'error' => $e->getMessage(),
35             ];
36         }
37
38         return response()->json($result,$result['status']);
39     }
40 }

```

TodoController

Membuat controller baru dengan nama TodoController dengan memasukkan perintah

**php artisan make:controller
TodoController**

Kemudian menambahkan sintaks ke file **TodoController.php** seperti pada gambar disamping.

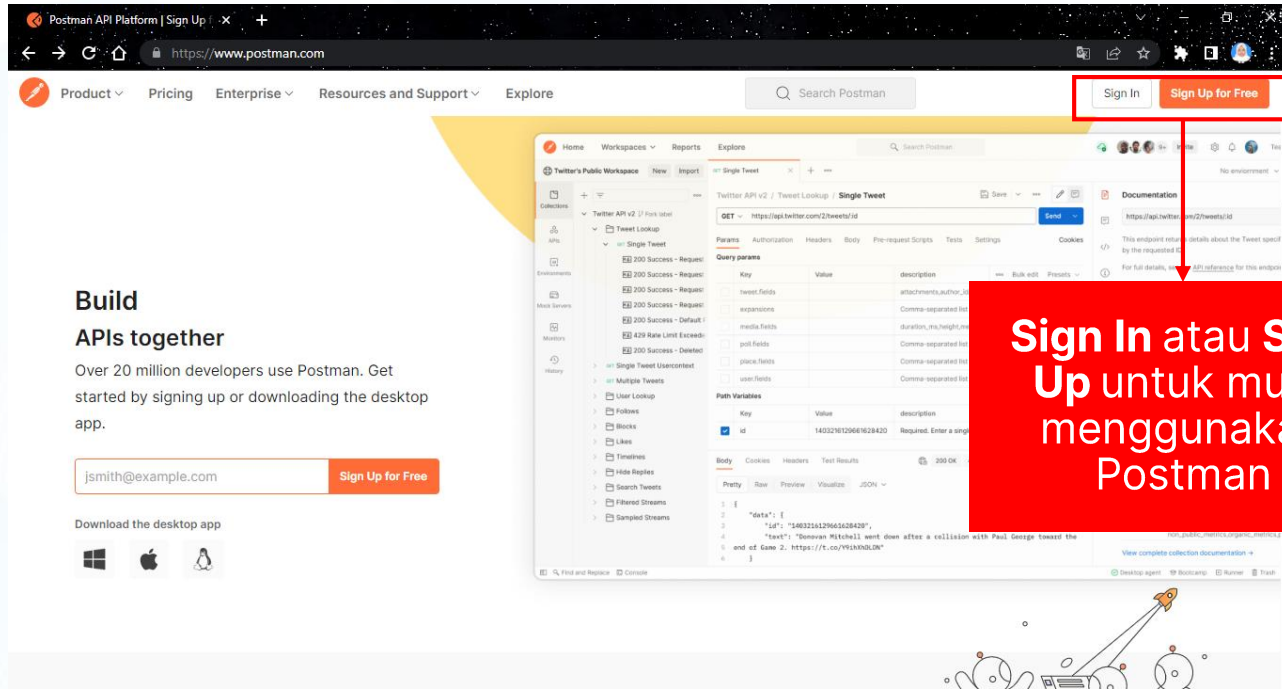
Route

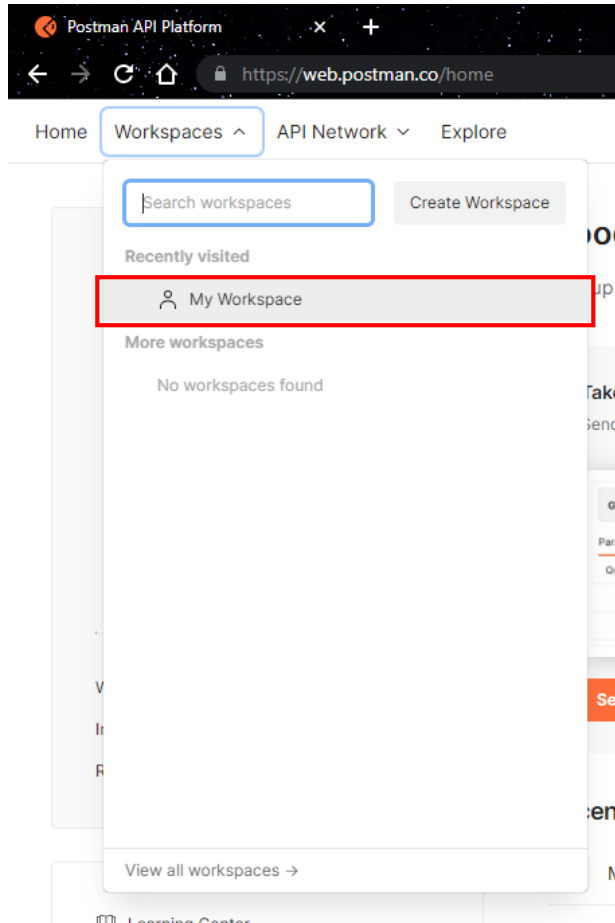
Menambahkan router pada file **./route/api.php**

```
21 Route::get('get_todo', 'App\Http\Controller\TodoController@getTodoList');  
22 Route::post('add_todo', 'App\Http\Controller\TodoController@addTodo');
```

Testing Menggunakan Postman

Membuka aplikasi Postman yang bisa didownload pada <https://www.postman.com/>. Pada link tersebut bisa juga menggunakan Postman secara online tanpa harus menginstal pada perangkat.





Setelah Sign In atau Sign Up maka akan masuk ke halaman **Home**. Pada menu bar pilih menu **Workspace** lalu pilih sub menu **My Workspace** untuk mulai menggunakan Postman.

POST

http://localhost:8000/api/add_todo

Send

ParamsAuthorizationHeaders (10)BodyPre-request ScriptTestsSettingsCookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	title	Testing Data			
<input type="checkbox"/>	detail	Hai			
	Key	Value	Description		

BodyCookies (2)Headers (10)Test Results

Status: 201 CreatedTime: 109 msSize: 479 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "status": 201,
3   "data": {
4     "_id": "6264e08a72006485ff5d3e82",
5     "title": "Testing Data",
6     "updated_at": "2022-04-24T05:30:50.725000Z",
7     "created_at": "2022-04-24T05:30:50.725000Z"
8   }
9 }
```

GET

http://localhost:8000/api/get_todo

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input type="checkbox"/>	title	Testing Data			
<input type="checkbox"/>	detail	Hai			
	Key	Value	Description		

Body

Cookies (2)

Headers (10)

Test Results

Status: 200 OK

Time: 58 ms

Size: 725 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  [
2    {
3      "_id": "6264da4018744f611d21e675",
4      "title": "Tes",
5      "updated_at": "2022-04-24T05:04:00.407000Z",
6      "created_at": "2022-04-24T05:04:00.407000Z"
7    },
8    {
9      "_id": "6264de5618744f611d21e676",
10     "titile": "Tes",
11     "updated_at": "2022-04-24T05:21:26.810000Z",
12     "created_at": "2022-04-24T05:21:26.810000Z"
13   },
14   ]
```

Terima **Kasih**