



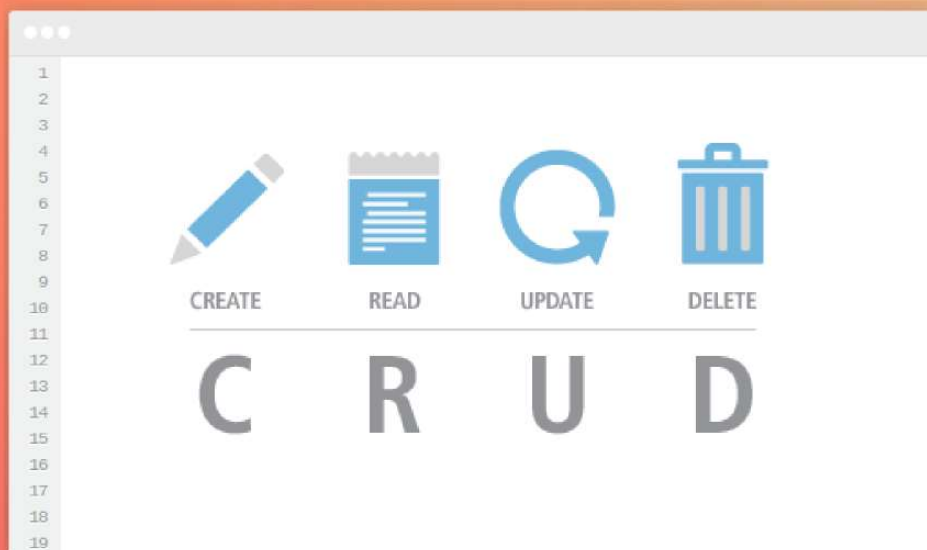
Ahmad Rosid

[Follow](#)<https://ahmadrosid.com> The more I learn, the more I realize how much I don't know.

Sep 19, 2016 · 4 min read

## Membangun web API Dengan Lumen 5.3 Part 2

Lumen.



### Apa itu CRUD?

Seperti biasa saya selalu seka membuka pertanyaan untuk materi yang akan kita bahas. OK berselancar di wikipedia yuk!!!

*Crud are the four basic functions of persistent storage. Alternate words are sometimes used when defining the four basic functions of CRUD, retrieve instead of read, modify instead of update, or destroy instead of delete*

Jadi CRUD itu singkatan dari create, read, update and delete. Untuk mengolah data CRUD ini fungsi yang sering kita gunakan, dimana kita

bisa membuat, membaca, merubah dan menghapus sebuah data. CRUD ini juga adalah keterampilan penting bagi seorang software developer, karna pada dasarnya hal utama dalam software development itu sendiri adalah transforming data. Jadi keterampilan dalam operasi CRUD ini sangat penting untuk kita asah. Begitu juga dalam membuat sebuah API kita tidak akan jauh—jauh dari operasi CRUD ini.

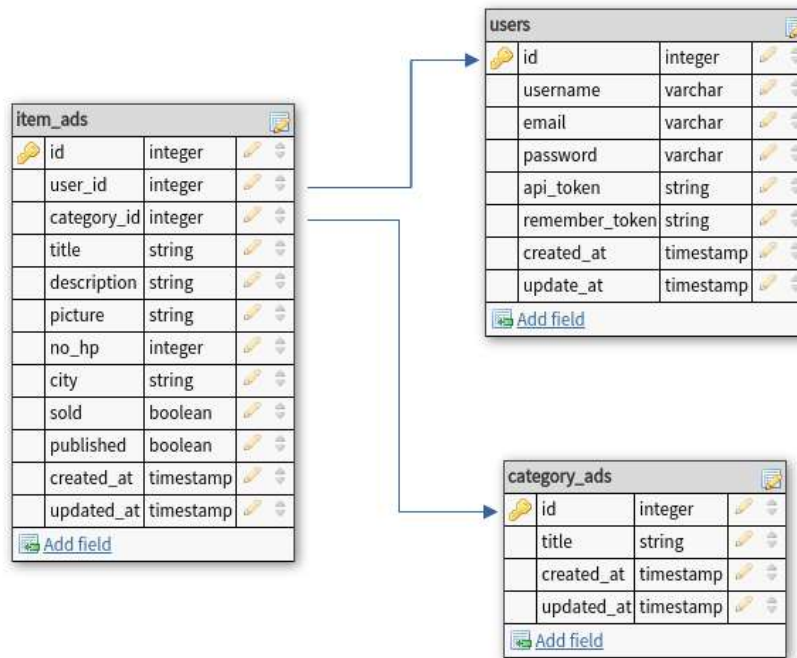
## Konsep Aplikasi

Pada study case kali ini kita akan belajar bagaimana melakukan operasi CRUD ini. Disini kita akan membuat aplikasi untuk menjual barang bekas. Disini kita coba membuat versi sederhana dari olx.

Ok disini saya akan bicara sedikit konsep dari aplikasi yang akan kita buat. Jadi pada aplikasi yang akan kita buat ini nanti user akan melakukan register dan bisa memasang iklan untuk barang yang akan mereka jual. Selain itu user bisa merubah, menambah, dan menghapus iklan yang mereka buat. Untuk itu disini kita akan mengimplemntasikan konsep CRUD ini.

## Database

Sebelum masuk development kita perlu membuat desain database dari aplikasi yang akan kita buat ini. Pada kasus aplikasi yang akan kita buat kali ini kita akan membuat 2 database dengan nama **item\_ads** dan **category\_ads**. Nah berikut ini schema databasenya :



Nah untuk belajar mendesain database googling ya guys karna mungkin akan terlalu panjang kalau saya bahas disini.

#### cara mendesain database - Penelusuran Google

12 Apr 2016 ... Di dalam pembahasan ini, Akan diperlihatkan bagaimana cara mendesain dan...  
[www.google.co.id](http://www.google.co.id)

Ok sekarang setelah mendesain database kita buat schema database menggunakan artisan, jalankan perintah berikut ini :

```
php artisan make:migration create_item_ads_table --
create=item_ads
```

Dan untuk schema databse migrationnya seperti ini :

```
Schema::create('item_ads', function (Blueprint $table) {  
    $table->increments('id');  
    $table->integer('user_id');  
    $table->integer('category_id');  
    $table->string('title');  
    $table->string('description');  
    $table->string('picture');  
    $table->integer('no_hp');  
    $table->string('city');  
    $table->boolean('sold');  
    $table->boolean('published');  
    $table->timestamps();  
});
```

Lanjut lagi kita akan buat database untuk category :

```
php artisan make:migration create_category_ads_table --  
create=category_ads
```

Dan ini untuk migration category\_ads nya

```
Schema::create('category_ads', function (Blueprint $table) {  
    $table->increments('id');  
    $table->string('name');  
    $table->timestamps();  
});
```

Sekarang kita buat model untuk database **item\_ads** dan **category\_ads** jalankan perintah berikut ini. Dalam kasus ini saya memberi contoh membuat file model lewat terminal :

```
touch app/ItemAds.php  
touch app/CategoryAds.php
```

dan berikut ini untuk isi dari file ItemAds.php

```
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  /**
8   * Model item ads
9   */
10 class ItemAds extends Model
11 {
12
13     /**
14      * Table database
15      */
16     protected $table = 'item_ads';
17
18     /**
19      * The attributes that are mass assignable.
20      *
21      * @var array
22      */
```

Dan selanjutnya untuk model CategoryAds.php seperti berikut ini

```
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  /**
8   * Model item ads
9   */
10 class CategoryAds extends Model
11 {
12
13     /**
14      * Table database
15      */
16     protected $table = 'category_ads';
17
18     /**
```

Sudah buat migrationnya sudah juga buat modelnya sekarang kita lanjut migrate databasenya :

```
php artisan migrate
```

## Db Seeder

Apa itu db seed? Db seed ini adalah salah satu dari lumen untuk membuat sample data dari database yang kita buat. Nah fungsinya untuk apa? Fungsinya untuk testing terkadang ketika kita ingin melakukan test menampilkan data kita perlu membuat sample datanya nah fitur db seed ini adalah salah satu alternative yang mudah untuk melakukan sampling data yang akan kita test.

Sudah cukup paham dengan database seeder? Kalo masih belum puas baca dari sumbernya [disini](#). Ok disini kita akan membuat 2 seeder untuk category dan item ads. Untuk file dari fungsi seed ini ada dalam folder **lumen-web-api/database/seeds**.

Sekarang kita buat database seed untuk category. Buat file **CategoryAdsSeeder.php** pada folder seeds seperti berikut ini :

```
1  <?php
2
3  use Illuminate\Database\Seeder;
4  use Illuminate\Database\Eloquent\Model;
5
6  class CategoryAdsSeeder extends Seeder {
7
8      public function run()
9      {
10         DB::table('category_ads')->delete();
11         $category = app()->make('App\CategoryAds');
```

Sekarang lanjut kita buat seed untuk item ads nya dengan membuat file **ItemAdsSeeder.php** seperti berikut ini :

```
1  <?php
2
3  use Illuminate\Database\Seeder;
4  use Illuminate\Database\Eloquent\Model;
5
6  class ItemAdsSeeder extends Seeder {
7
8      public function run()
9      {
10         DB::table('item_ads')->delete();
11         $item = app()->make('App\ItemAds');
12         $item->fill([
13             'user_id' => 1,
14             'category_id' => 1,
15             'title' => 'Macbook Pro 14 in',
16             'price' => 12000000,
17             'description' => 'Di jual macbook pro masih mulus
18             'picture' => 'macbook-pro-2012.php',
```

Yang terakhir kita akan buat seeder untuk user dengan file seeder karna disini nanti kita akan menggunakannya untuk sampling data user.

Beri nama **UserSeeder.php** :

```
1  <?php
2
3  use Illuminate\Database\Seeder;
4  use Illuminate\Database\Eloquent\Model;
5
6  class UserSeeder extends Seeder {
7
8      public function run()
9      {
10         DB::table('users')->delete();
11         $item = app()->make('App\User');
12         $hasher = app()->make('hash');
13         $password = $hasher->make('password');
14         $api_token = sha1(time());
15         $item->fill([
16             'username'=>'Ahmad Rosid',
```

Sekarang jalankan perintah berikut ini untuk melakukan seeder database ini :

```
composer dump-autoload

php artisan migrate:refresh --seed
```

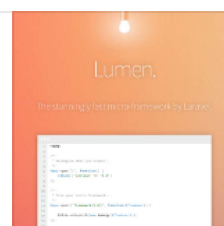
## Controller CRUD

DI postingan pertama tutorial ini saya sudah jelaskan tentang controller so buat kamu yang belum baca yang postingan pertama silahkan buka postingan nya dibawah ini :

Membangun Web API dengan Lumen 5.3 part 1

Apa itu Lumen?

medium.com





Disini kita akan membuat 2 controller untuk database item ads dan category ads. Ok yang pertama kita buat untuk **CategoryAdsController.php** seperti berikut ini.

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\CategoryAds;
7
8  class CategoryAdsController extends Controller
9  {
10
11     /**
12      * Create a new auth instance.
13      *
14      * @return void
15      */
16     public function __construct()
17     {
18         $this->middleware('auth');
19     }
20
21     /**
22      * Get all data from category
23      */
24     public function index(Request $request)
25     {
26         $category = new CategoryAds;
27
28         $res['success'] = true;
29         $res['result'] = $category->all();
30
31         return response($res);
32     }
33
34     /**
35      * Insert database for CategoryAds
36      * Url : /category
37      */
38     public function create(Request $request)
39     {
40         $category = new CategoryAds;
41         $category->fill(['name' => $request->input('name')])

```

```

42         1+($category->save()){
43             $res['success'] = true;
44             $res['result'] = 'Success add new category!';
45
46             return response($res);
47         }
48
49     }
50
51     /**
52      * Get one data CategoryAds by id
53      * Url : /category/{id}
54      */
55     public function read(Request $request, $id)
56     {
57         $category = CategoryAds::where('id',$id)->first();
58         if ($category !== null) {
59             $res['success'] = true;
60             $res['result'] = $category;
61
62             return response($res);
63         }else{
64             $res['success'] = false;
65             $res['result'] = 'Category not found!';
66
67             return response($res);
68         }

```

Dan berikut ini untuk routenya :

```

1
2  /*
3   | -----
4   |  CRUD Routes using auth middleware
5   | -----
6   */
7   $app->get('/category', 'CategoryAdsController@index');
8   $app->get('/category/{id}', 'CategoryAdsController@read');

```

Sekarang untuk controller kedua kita akan membuat crud controller untuk table item\_ads dengan nama **ItemAdsController.php**

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\ItemAds;
7
8  class ItemAdsController extends Controller
9  {
10
11     /**
12      * Create a new auth instance.
13      *
14      * @return void
15      */
16     public function __construct()
17     {
18         $this->middleware('auth');
19     }
20
21     /**
22      * Get all data from item_ads
23      */
24     public function index(Request $request)
25     {
26         $item_ads = ItemAds::where('published', true)->get();
27         if (count($item_ads) !== 0) {
28             $res['success'] = true;
29             $res['result'] = $item_ads;
30
31             return response($res);
32         }else{
33             $res['success'] = true;
34             $res['result'] = 'No ads have been published!';
35
36             return response($res);
37         }
38     }
39 }
40
41 /**

```

```

42      * Insert database for ItemAds
43      * Url : /item_ads
44      */
45      public function create(Request $request)
46      {
47          $item_ads = new ItemAds;
48          $item_ads->fill([
49              'user_id' => $request->input('user_id'),
50              'category_id' => $request->input('category_id'),
51              'title' => $request->input('title'),
52              'price' => $request->input('price'),
53              'description' => $request->input('description'),
54              'picture' => $request->input('picture'),
55              'no_hp' => $request->input('no_hp'),
56              'city' => $request->input('city'),
57              'sold' => false,
58              'published' => true
59          ]);
60          if($item_ads->save()){
61              $res['success'] = true;
62              $res['result'] = 'Success add new item_ads!';
63
64              return response($res);
65          }
66      }
67
68      /**
69       * Get one data ItemAds by id
70       * Url : /item_ads/{id}
71       */
72      public function read(Request $request, $id)
73      {
74          $item_ads = ItemAds::where('id',$id)->first();
75          if ($item_ads !== null) {
76              $res['success'] = true;
77              $res['result'] = $item_ads;
78
79              return response($res);

```

Dan tabahkan route nya seperti ini :

```
1
2      /* Route item ads */
3      $app->get('/item_ads', 'ItemAdsController@index');
4      $app->get('/item_ads/{id}', 'ItemAdsController@read');
5      $app->get('/item_ads/delete/{id}', 'ItemAdsController@delete');
6      $app->post('/item_ads/create', 'ItemAdsController@create');
```

Ok di postingan awal saya sudah memberikan contoh testing api nya silahkan coba di praktikan. Berikut ini daftar url yang bisa di test :

## Category Ads Url Test

```
GET : /category
http://localhost/category/?
api_token=7e18e4ded86362d5e237d06e5bfeebfed9e6fd4b

GET : /category/{id}
http://localhost/category/1/?
api_token=7e18e4ded86362d5e237d06e5bfeebfed9e6fd4b

POST : /category/update/{id}
input = name
http://localhost/category/1/?
api_token=7e18e4ded86362d5e237d06e5bfeebfed9e6fd4b

GET : /category/delete/{id}
http://localhost/category/delete/1/?
api_token=7e18e4ded86362d5e237d06e5bfeebfed9e6fd4b
```

## Item Ads Url Test

```
GET : /item_ads
http://localhost/item_ads/?
api_token=7e18e4ded86362d5e237d06e5bfeebfed9e6fd4b

GET : /item_ads/{id}
http://localhost/item_ads/1/?
api_token=7e18e4ded86362d5e237d06e5bfeebfed9e6fd4b

POST : /item_ads/update/{id}
http://localhost/item_ads/1/?
api_token=7e18e4ded86362d5e237d06e5bfeebfed9e6fd4b
```

```
GET :  
http://localhost/item_ads/delete/1/?  
api_token=7e18e4ded86362d5e237d06e5bfeebfed9e6fd4b
```

Nah disini aplikasi yang kita buat adalah memberikan fitur untuk user jadi disini user bisa menambahkan kategory dan juga menambahkan item dari iklan yang ingin mereka buat. Nah disini dasarnya untuk aplikasi mini olx ini silahkan temen—temen kembangkan tambah kan fitur untuk menampilkan iklan kepada pembeli nya atau yang lainnya. Jangan takut untuk ber eksplorasi kalau ada error silah kan bertanya. Tapi biasakan sebelum bertanya kamu cari dulu error nya inget ada google dan stackoverflow dimana disini kita bisa dapet sample error yang mungkin juga kita alami.

Ok semangat ya belajar buat API nya tunggu next artikel dari saya semoga bisa bermanfaat. Untuk source code dari aplikasi ini bisa akses di bawah sini :

**lumen web api CRUD · ar-android/Lumen-Web-API@effc1d4**

Lumen-Web-API - Simple web API using Lumen 5.3 with Simple Auth System

[github.com](#)







