**URL: https://www.codecheef.org/article/laravel-vue-js-search-example-with-lodash-debounce**

**Laravel Vue Js Search Example with Lodash (Debounce)**

**Posted** Mahedi Hasan **Category** Laravel 7.x **Published** March 9, 2021

Hello artisan, coming with a brand new tutorial that is Laravel vue js search example with debounce. In this laravel vuejs ajax search example you will learn how to search data from database in Laravel and vue js using lodash.

To make a ajax request request to fetch vue js search data i will use axios. You know that axios is a javascript promise based http popular request. So i will use this axios request to fetch our vue js laravel search data with pagination.

laravel vuejs search with pagination example i will use a controller and fetch data from this controller. From this laravel vue search tutorial you will also learn lodash and how to import debounce from lodash vuejs.

**Step 1: Install Laravel**

In this vue lodash debounce tutorial you have to first download a fresh laravel project to learn laravel vue js instant search data with pagination.

composer create-project --prefer-dist laravel/laravel blog

PHP

**Step 2:  Create Model**

We need units table. So paste this below code to units table

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateUnitsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('units', function (Blueprint $table) {

$table->increments('id');

$table->string('name');

$table->tinyInteger('is\_active')->default(1);

$table->integer('created\_by')->nullable();

$table->integer('updated\_by')->nullable();

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('units');

}

}

PHP

Now run php artisan migrate command to migrate units table.

php artisan migrate

PHP

We need to setup our Unit model. so paste this below code to avoid mass assignment error before inserting data to units table.

**app/Unit.php**

namespace App;

use Illuminate\Database\Eloquent\Model;

class Unit extends Model

{

protected $guarded = [];

}

PHP

**Step 3:  Setup Route**

We have to first fetch the data then we have to create search data option for vue js. So we need two routes like below. So update your web.php file as like below.

**routes/web.php**

Route::get('/unit', 'UnitController@view\_unit')->name('view');

Route::get('/search\_unit', 'UnitController@search\_unit\_by\_key');

PHP

And update api.php

**routes/api.php**

Route::apiResources(

[

'unit' => 'API\UnitController'

]

);

PHP

**Step 4:  Create Controller**

Now we have to create controller to create those avobe method. so create contoller and paste this below code.

php artisan make:controller API/UnitController --api

PHP

Now paste below code in UnitController.php

**app/Https/Controller/UnitController.php**

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;

use App\Unit;

use Illuminate\Http\Request;

use Symfony\Component\HttpFoundation\Response;

class UnitController extends Controller

{

public function view\_unit()

{

return view('unit');

}

public function search\_unit\_by\_key()

{

$key = \Request::get('q');

$unit = Unit::withCount('products')

->where('name','LIKE',"%{$key}%")

->orWhere('is\_active','LIKE',"%{$key}%")

->get();

return response()->json([ 'unit' => $unit ],Response::HTTP\_OK);

}

}

PHP

[](https://go.ezodn.com/ads/charity/proxy?p_id=afead638-40cf-4351-7d75-6d18abe823e7&d_id=267458&imp_id=1707339784932063&c_id=1076&l_id=10016&url=https%3A%2F%2Fsecure.info-komen.org%2Fsite%2FDonation2%3F23777.donation%3Dform1%26amp%3Bidb%3D653252013%26amp%3Bdf_id%3D23777%26amp%3BFR_ID%3D8793%26amp%3Bmfc_pref%3DT%26amp%3BPROXY_ID%3D8793%26amp%3BPROXY_TYPE%3D21&ffid=1&co=BD)

And we have to also update our API Unit controller to create search with vue js in laravel.

**app/Https/Controller/API/UnitController.php**

namespace App\Http\Controllers\API;

use App\Http\Controllers\Controller;

use App\Unit;

use Illuminate\Http\Request;

use Symfony\Component\HttpFoundation\Response;

class UnitController extends Controller

{

public function index( Request $request )

{

return $request->ajax() ? Unit::paginate(5) : abort(404);

}

}

PHP

**Step 5:   Configure Vue**

If you want to install vue in laravel 8 application then install following laravel ui composer package to get it. So run below command

composer require laravel/ui

PHP

**Install Vue**

php artisan ui vue

PHP

Now run this command to install laravel vue pagination package

npm install laravel-vue-pagination

PHP

Now our setup is completed. Now we just need to change our app.js file and our component file.

**resources/assets/js/components/UnitComponent.vue**

<template>

<div class="card">

<div class="card-header" id="hide">

<div class="row">

<div class="col-md-2">

<input type="text" @keyup="searchUnit" placeholder="Search" v-model="search" class="form-control form-control-sm">

</div>

</div>

</div>

<div class="card-body" id="hide\_again">

<table id="" class="table table-bordered table-striped">

<thead>

<tr>

<th>Name</th>

<th>Status</th>

</tr>

</thead>

<tbody>

<tr v-for="unit in units.data" :key="unit.id">

<td>{{ unit.name }}</td>

<td>{{ unit.is\_active == 1 ? 'Active' : 'InActive' }}</td>

</tr>

</tbody>

<pagination :data="units" @pagination-change-page="getResults"></pagination>

</table>

</div>

</div>

</template>

<script>

import \_ from "lodash";

export default {

data() {

return {

units: {},

search: '',

}

},

methods: {

fetchingAllUnit() {

axios.get("api/unit").then( data => (this.units = data.data));

},

getResults(page = 1) {

axios.get('api/unit?page=' + page)

.then(response => {

this.customers = response.data;

});

},

searchUnit:\_.debounce(function(){

axios.get('/search\_unit?q='+this.search)

.then((response)=>{

this.units.data = response.data.unit

})

}),

},

created(){

this.fetchingAllUnit();

},

}

</script>

HTML

now update the app.js file like belew

**resources/js/app.js**

require('./bootstrap');

window.Vue = require('vue');

Vue.component('pagination', require('laravel-vue-pagination'));

Vue.component('unit-component', require('./components/UnitComponent.vue').default);

const app = new Vue({

el: '#app',

});

JavaScript

**Step 6:   Configure blade file**

Now almost all are set to go. open resources/layouts/app.php and paste this following code.

**resources/views/layouts/app.blade.php**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="csrf-token" content="{{ csrf\_token() }}">

<title>{{ config('app.name', 'Laravel') }}</title>

<link href="{{ mix('css/app.css') }}" rel="stylesheet">

<title>Document</title>

</head>

<body>

<div id="app">

<div class="py-4">

@yield('content')

</div>

</div>

<script src="{{ mix('js/app.js') }}" defer></script>

</body>

</html>

HTML

Now open resources/views/post/create.blade.php file and paste this code.

**resources/views/unit.blade.php**

@extends('layouts.app')

@section('content')

<unit-component></unit-component>

@endsection

HTML

Now everything is set to go. Now need to compile our all javascript file. so run below command.

npm run dev

JavaScript

**Read also :**[**Real Time Chat App with Laravel 7.x Vue Js and Pusher**](https://www.codecheef.org/article/real-time-chat-app-with-laravel-6-vue-js-and-pusher)

Now visit the following url to see Laravel vue js instant search example without page refresh with pagination using lodash debouce.