

**Tanzim Ibthesam**

Posted on Apr 5, 2021 • Updated on Aug 24, 2021

Making Api CRUD(Create,Read,Update,Delete) with Laravel 8 n API Authentication with sanctum

#laravel #php

Here I am going to mention about how you can make an API with Laravel 8 and Using PostMan. This is inspired from Traversy Medias videos on Laravel APi and Authentication with sanctum [Laravel 8 REST API With Sanctum Authentication](#). SO you can definitely go and watch that.

1.Creating a new Laravel project

```
laravel new laravel-api-sanctum
```

2.Create a model Product along with Controllers,Seeders,Factory

php artisan make:model Product -a it creates model,factory,seeder here its not essential to create Seeders and Factory but I showed this command so that it can help in future development when you need to use seeders and factories.

3.In Product migrations table

```
$table->id();  
$table->string('name');  
$table->string('slug');  
$table->string('description')->nullable();  
$table->decimal('price',5,2);  
$table->timestamps();
```

Then run `php artisan migrate`

4. API testing and routes api.php

Usually in Laravel when we use web.php,when we want to make Single Page Application or mobile apps we need api for that we need to use api.php

in api.php we can write

```
Route::post('/products', function () {

    return Product::create([
        'name'=>'Product One',
        'slug'=>'product-one',
        'description'=>'Details of Product One',
        'price'=>9.99

    ]);

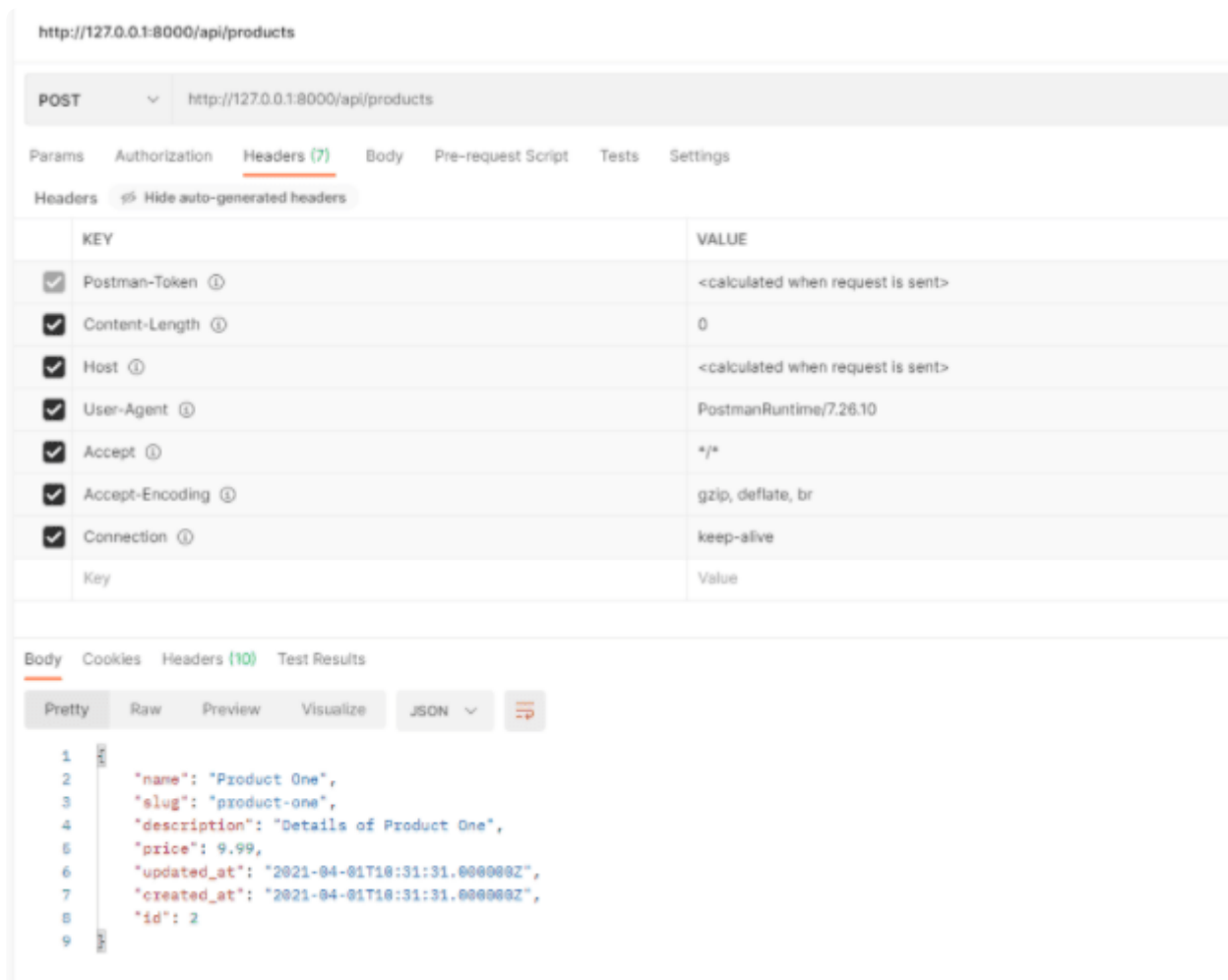
});
```

Then just go to Product.php and for mass assignment. It is a process through which an array of data will be saved to specific model. Here model is Product

In Product.php we need to write protected \$guarded=[];

In Postman while sending requests we get

CREATE



This is how we send post requests in POSTMAN

READ

```
public function index(){
    return Product::all();
}
```

This is how we can get all products data in PostMan

http://127.0.0.1:8000/api/products

GET http://127.0.0.1:8000/api/products

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Headers Hide auto-generated headers

KEY	VALUE
<input checked="" type="checkbox"/> Postman-Token ⓘ	<calculated when request is sent>
<input checked="" type="checkbox"/> Host ⓘ	<calculated when request is sent>
<input checked="" type="checkbox"/> User-Agent ⓘ	PostmanRuntime/7.26.10
<input checked="" type="checkbox"/> Accept ⓘ	*/*
<input checked="" type="checkbox"/> Accept-Encoding ⓘ	gzip, deflate, br
<input checked="" type="checkbox"/> Connection ⓘ	keep-alive
Key	Value

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```

1  [
2    {
3      "id": 1,
4      "name": "Product One",
5      "slug": "product-one",
6      "description": "Details of Product One",
7      "price": "9.99",
8      "created_at": "2021-04-01T10:12:48.000000Z",
9      "updated_at": "2021-04-01T10:12:48.000000Z"
10   }
11 ]
```

To send request through Controller in **ProductController.php**

Now let us try to make requests through Controller. Here its a Resource Controller
In api.php

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

In ProductController.php

```
public function store(Request $request)
{
    //
    return Product::create($request->all());
}
```

If we again send get requests to Postman we will get all 3 products

But if we try to insert data without any field it must give us an error

VALIDATION ERROR

```
public function store(Request $request)
{
    //
    $request->validate([
        'name'=>'required',
        'description'=>'required',
        'slug'=>'required',
        'price'=>'required'

    ]);
    return Product::create($request->all());
}
```

Get a single product from the route

Now instead of writing each and every route we can write it in a Resource Controller

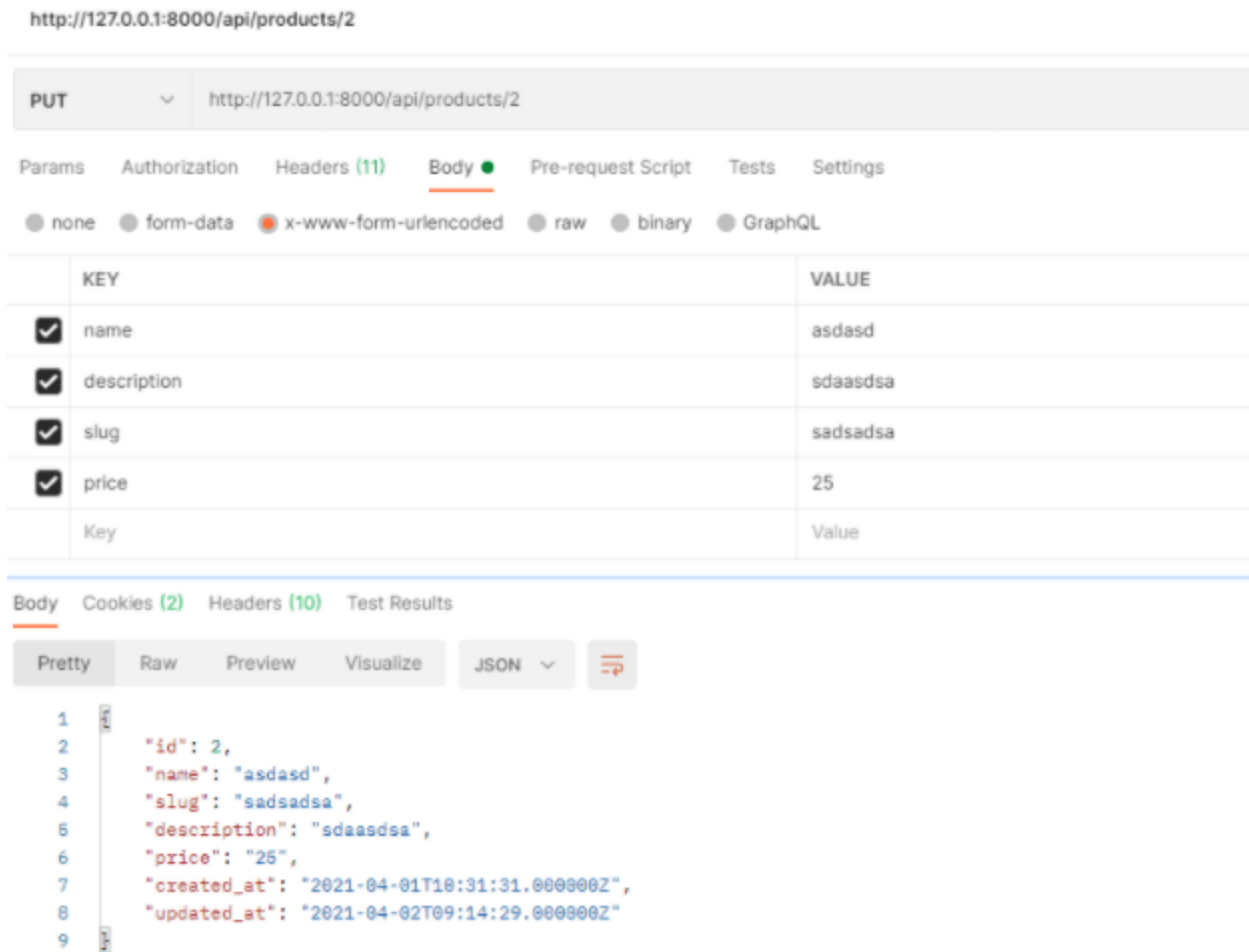
```
Route::resource('products', ProductController::class);
```

Update data

In ProductController.php

```
public function update(Request $request,$id)
{
    //
    $product=Product::find($id);
    $product->update($request->all());
    return $product;
}
```

While we send requests to PostMan for Update this is what it give us



Here we can specify the id we want to update

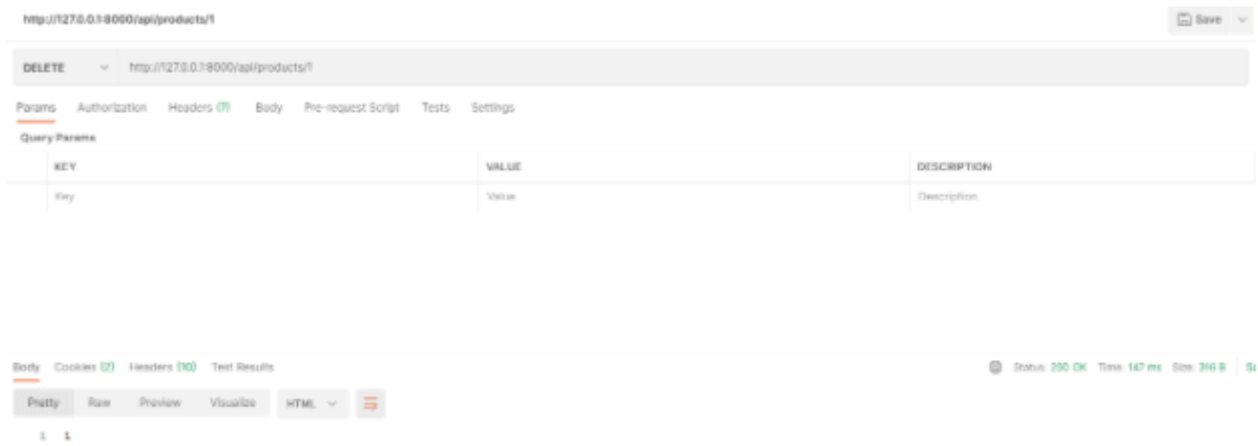
DELETE

Delete Data

In ProductController.php

```
public function destroy($id)
{
    //
    return Product::destroy($id);
}
```

If we send delete request in Postman we get



We are deleting as per id.

For search

In `ProductController.php`

```
public function search($name)
{
    //
    return Product::where('name', 'like', '%'.$name.'%')->get();
}
```

In Postman while sending search requests

http://127.0.0.1:8000/api/products/search/galaxy

GET ⌵ http://127.0.0.1:8000/api/products/search/galaxy

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies (2) Headers (10) Test Results

Pretty Raw Preview Visualize JSON ⌵ ⌵

```
1  [
2    {
3      "id": 4,
4      "name": "Samsung-Galaxy-S21",
5      "slug": "galaxy-s-21",
6      "description": "Flagship phone",
7      "price": "999.99",
8      "created_at": "2021-04-02T09:45:15.000000Z",
9      "updated_at": "2021-04-02T09:45:15.000000Z"
10   }
11 ]
```

API Authentication with Sanctum

Now we are done with API CRUD part we will now enable API authentication with [Sanctum](#)

To install Laravel sanctum

composer require laravel/sanctum

php artisan vendor:publish --provider="Laravel\Sanctum\SanctumServiceProvider"

php artisan migrate

In app/Http/Kernel.php

```
'api' => [
    \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful
    'throttle:api',
```

```
\Illuminate\Routing\Middleware\SubstituteBindings::class,
```

```
],
```

In User.php we need to write use use Laravel\Sanctum\HasApiTokens on top and HasApiTokens below

```
use Laravel\Sanctum\HasApiTokens;
class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;
```

In api.php

Public routes means here anyone can view the products all products or individual products

```
/public routes
Route::get('/products', [ProductController::class, 'index']);
// Route::resource('products', ProductController::class);
Route::get('/products/{id}', [ProductController::class, 'show']);

//Protected routes-Only authenticated users can have access to protected routes
Route::group(['middleware' => ['auth:sanctum']], function () {
    Route::post('/products', [ProductController::class, 'store']);
    Route::put('/products/{id}', [ProductController::class, 'update']);
    Route::delete('/products/{id}', [ProductController::class, 'delete']);

});
```

Auth Controller

Create a new Controller for Authentication

php artisan make:controller AuthController

Register method

```
public function register(Request $request){
    $fields=$request->validate([
        'name'=>'required|string',
        'email'=>'required|string|unique:users,email',
        'password'=>'required|string|confirmed'

    ]);
```



```
$user=User::create([
    'name'=>$fields['name'],
    'email'=>$fields['email'],
    'password'=>bcrypt($fields['password'])
]);

$token=$user->createToken('myapptoken')->plainTextToken;

$response=[
    'user'=>$user,
    'token'=>$token
];

return response($response,201);
}
```

In api.php

```
Route::post('/register',[AuthController::class,'register']);
```

In

api.php

Register

```
Route::post('/register',[AuthController::class,'register']);
```

Testing out in Postman

If we dont give anything it returns us validation errors

http://127.0.0.1:8000/api/register

POST

http://127.0.0.1:8000/api/register

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings
☐ none
 ☐ form-data
 ☒ x-www-form-urlencoded
 ☐ raw
 ☐ binary
 ☐ GraphQL

	KEY	VALUE
<input checked="" type="checkbox"/>	name	Tanzim
<input checked="" type="checkbox"/>	email	tanzim@gmail.com
<input checked="" type="checkbox"/>	password	123456789
<input checked="" type="checkbox"/>	password_confirmation	123456789
	Key	Value

Body Cookies Headers (10) Test Results

Pretty

Raw

Preview

Visualize

JSON



```

1  {
2    "user": {
3      "name": "Tanzim",
4      "email": "tanzim@gmail.com",
5      "updated_at": "2021-04-02T13:37:59.000000Z",
6      "created_at": "2021-04-02T13:37:59.000000Z",
7      "id": 1
8    },
9    "token": "1|uvtH5yVLyURWjEnxfhIn5xfRftwTs6vbx4exhTM9"
10 }

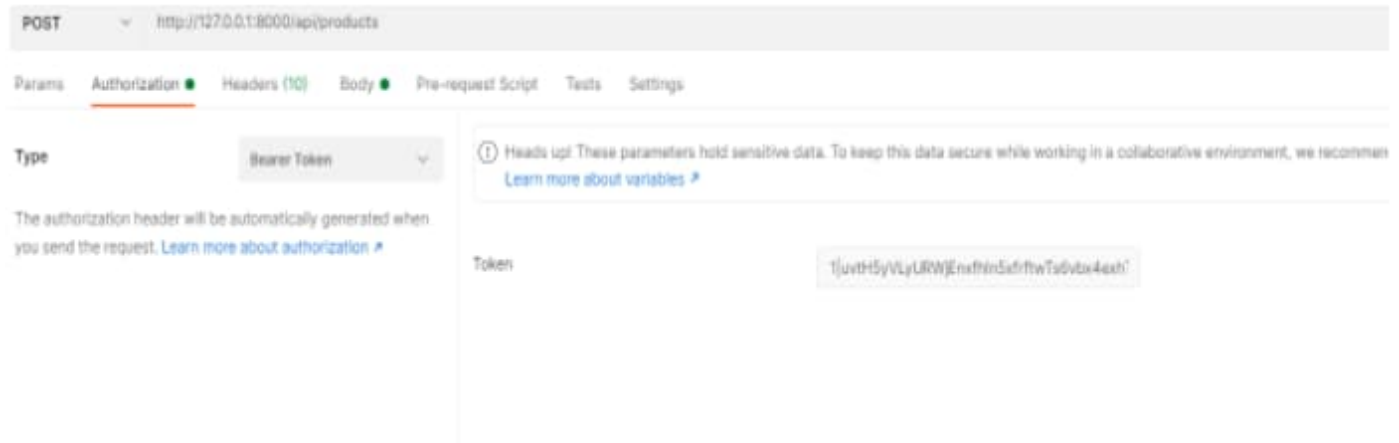
```

So here we see upon registration it gives us a token

Here with help of the token an authenticated user can post,edit and delete

Now we see it gives us a new token

With this token I will be able to gain access to protected routes. If we send post requests we need to use bearer select the bearer token option token and insert the token



While we send post request now we can post data

Now we need to create Logout function in AuthController.php

```
public function logout(Request $request){
    auth()->user()->tokens()->logout();

    return [
        'message'=>'Logged out'
    ];
}
```

In api.php

```
Route::group(['middleware' => ['auth:sanctum']], function () {
    Route::post('/products',[ProductController::class,'store']);
    Route::put('/products/{id}',[ProductController::class,'update']);
    Route::delete('/products/{id}',[ProductController::class,'delete']);
    Route::post('/logout',[AuthController::class,'logout']);

});
```

If we now insert the token in bearer token and try to logout

POST http://127.0.0.1:8000/api/logout

Params Authorization Headers (11) Body Pre-request Script Tests Settings

Type Bearer Token

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collat [Learn more about variables](#)

Token 1|uvtH5yVLyURWjEnxfhIn5xfrftwTs6vbx4e>...

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1
2  "message": "Logged out"
3
```

We see the message as logged out the token is no longer valid

Now we can delete,update and post if we are authenticated using that Token

Top comments (9)



Sanchita Paul • Feb 2 • Edited on



I am getting error during logout. I didn't use any relation ,then why it's showing MorphMany error ?

```
<title>🔥 Call to undefined method
Illuminate\Database\Eloquent\Relations\MorphMany::logout()</title>
```



Tanzim Ibthesam 🌟 • Feb 4



Hello Sanchita can you please check whether you have written the logout method in Controller properly.Thanks



Sanchita Paul • Feb 6





I solved this using delete method. It was showing error using logout.

```
public function logout(Request $request){  
    //auth()->user()->tokens()->logout();  
    auth()->user()->currentAccessToken()->delete();  
    return [  
        'message'=>'You're logged out'  
    ];  
}
```



Tanzim Ibthesam • Feb 6



Thanks



dgloriaweb • Sep 9 '21



Hi, forgot how to add a joined table to the index function pls help!!
This doesn't work

```
return Question::all()  
->with(compact('language'));
```



Tanzim Ibthesam • Sep 10 '21



Can you explain your models and relations



dgloriaweb • Sep 12 '21



Oh, thanks, I've figure it out.

```
return Question::with('language')->get();
```



Kirill • Sep 30 '21



Bad examples about `$request->all(...)`.
Good decision is `$request->validated();` OR `Product::create($request->validate(...))`



Tanzim Ibthesam • Sep 30 '21



Thanks so much for your feedback yes will correct it and such feedback really helps to understand the mistakes

[Code of Conduct](#) • [Report abuse](#)



Life is too short to browse without [dark mode](#)



Tanzim Ibthesam

JOINED

Jan 10, 2021

More from [Tanzim Ibthesam](#)

How to install Vue,Vue-router,Tailwind with Laravel 9,Vite

[#laravel](#) [#vue](#) [#vite](#) [#tailwindcss](#)

Install bootstrap in Laravel 9 with Vite

[#laravel](#) [#php](#) [#vite](#) [#bootstrap](#)

Laravel Api create Users,Posts with Role Permissions using Gates no packages

[#laravel](#) [#php](#) [#role](#) [#permissions](#)