

به نام خدا

پروژه درس یکپارچه سازی کاربردهای سازمانی
پیاده سازی وب سرویس - REST API

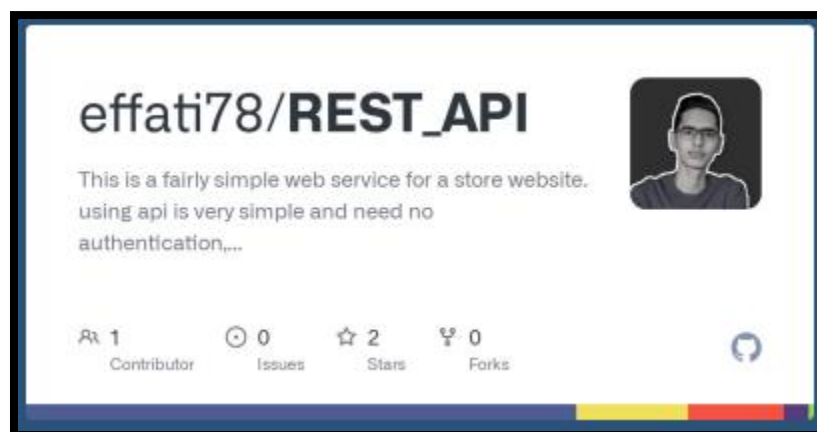
رضا عفتی مقدم

effati78@pm.me

این پروژه، وب سرویسی برای یک وب سایت فروشگاهی است؛ پروژه با زبان PHP (فریمورک Laravel ورژن 8) و دیتابیس MySQL پیاده‌سازی شده است.

سورس کد و دیتابیس پروژه درون لینک زیر است.

https://github.com/effati78/REST_API

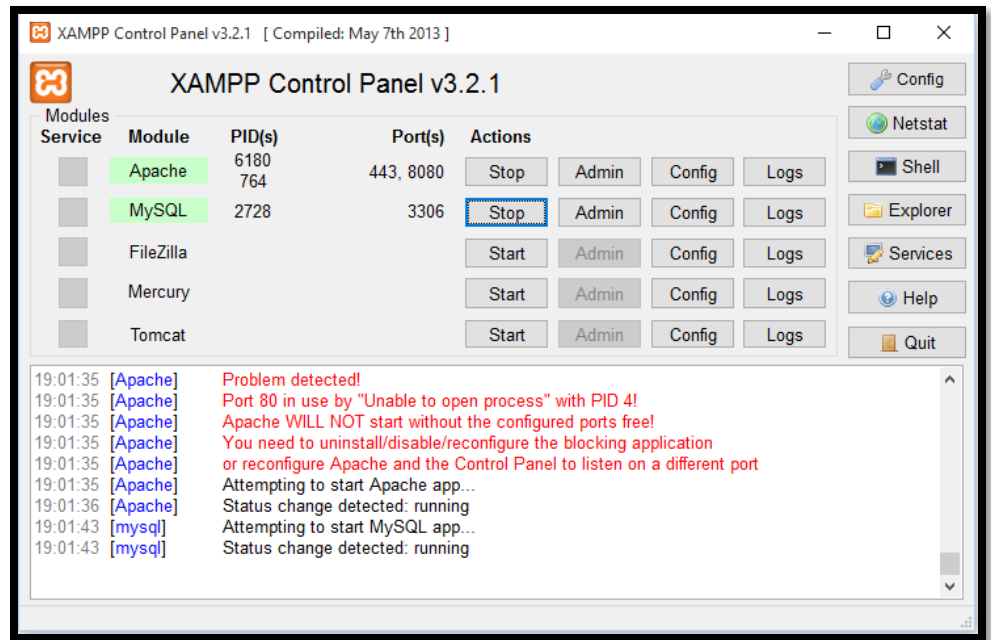


نحوه راه‌اندازی وب سرویس

چون پروژه درون لوکال هاست است، پس نیاز داریم که ابتدا وب سرویس را اجرا کنیم.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS  COMMENTS  [X] php - web-service -
web-service (master) php artisan serv
Starting Laravel development server: http://127.0.0.1:8000
[Wed Jul 6 21:36:15 2022] PHP 7.4.29 Development Server (http://127.0.0.1:8000) started
```

برای اتصال به دیتابیس MySQL و ارسال و دریافت اطلاعات باید نرم‌افزار Xampp را اجرا کنیم.



تحليل دیتابیس وب سرویس

نمای کلی جداول

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> comments	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> migrations	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> personal_access_tokens	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
<input type="checkbox"/> products	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> sponsors	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
5 tables	Sum	15	InnoDB	utf8mb4_general_ci	112.0 KiB	0 B

سه جدول اساسی داریم که وب سرویس بر پایه این جداول پیاده سازی شده است.

- محصولات - Products
- نظرات - Comments
- حامیان مالی - Sponsors

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 content	text	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	4 image_url	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5 price	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	6 score	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	7 isFeatured	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	8 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	9 updated_at	timestamp			Yes	NULL			Change Drop More

- جدول محصولات: شامل (نام محصول، توضیحات محصول، تصویر محصول، قیمت محصول، امتیاز محصول، محصول ویژه است یا نه)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 product_id	int(11)			No	None			Change Drop More
<input type="checkbox"/>	3 sender_name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	4 sender_email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5 sender_avatar	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	6 score	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	7 comment	text	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	8 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	9 updated_at	timestamp			Yes	NULL			Change Drop More

- جدول نظرات: شامل (آیدی محصول مورد نظر، نام فرستنده، تصویر فرستنده، ایمیل فرستنده، امتیاز داده شده به محصول، متن نظر)

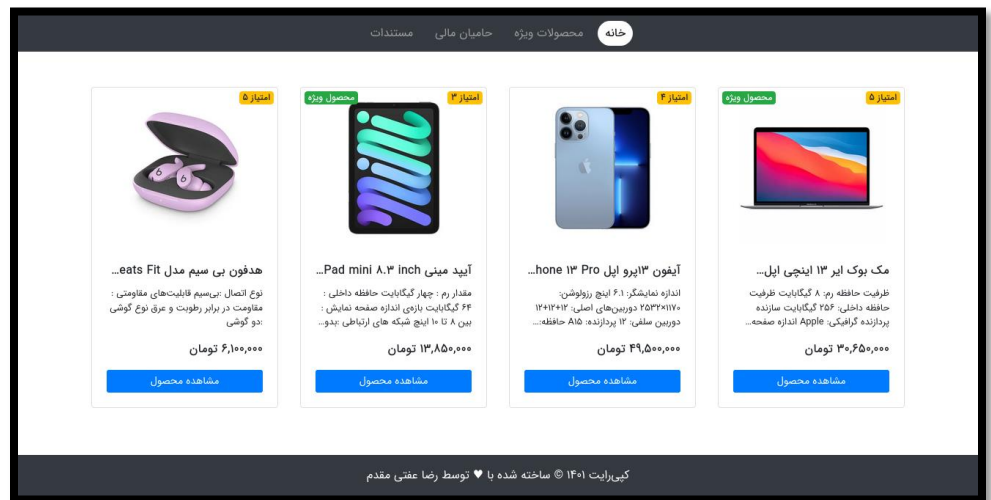
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 url	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	4 img	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	6 updated_at	timestamp			Yes	NULL			Change Drop More

- جدول حامیان مالی: شامل (نام شرکت، آدرس وبسایت، لوگو وبسایت)

رابط کاربری پروژه

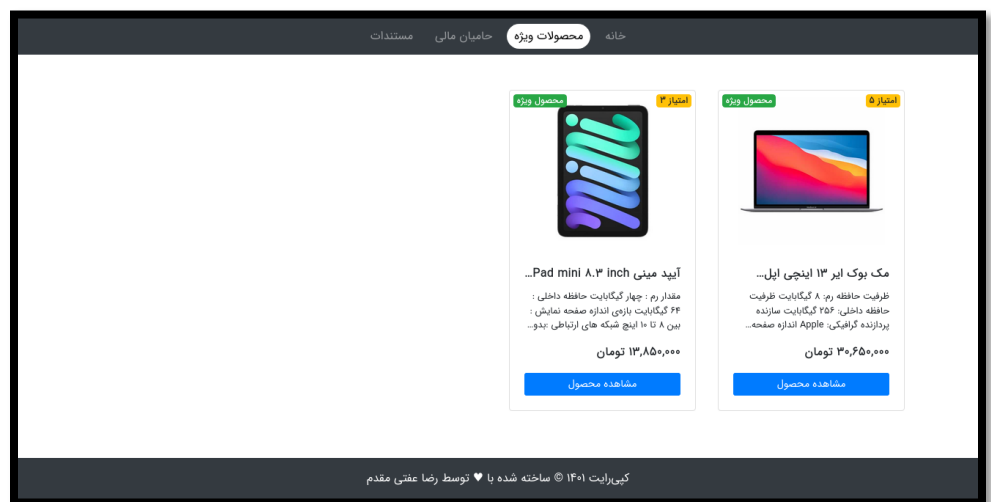
پس از راه اندازی وب سرویس، میتوانیم به صفحات پروژه دسترسی داشته باشیم.

صفحه اصلی (تمامی محصولات) - <http://127.0.0.1:8000>



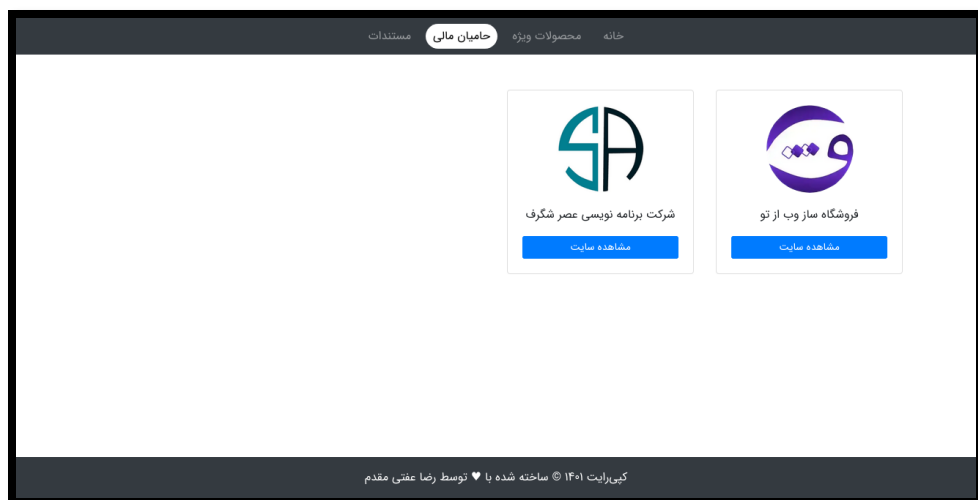
در این صفحه، تمامی محصولات فروشگاه با آیتم های آن لیست میشوند. اگر روی گزینه مشاهده محصول کلیک کنیم میتوانیم محصول رو با جزئیات بیشتری مشاهده کنیم.

صفحه محصولات ویژه - <http://127.0.0.1:8000/featured>



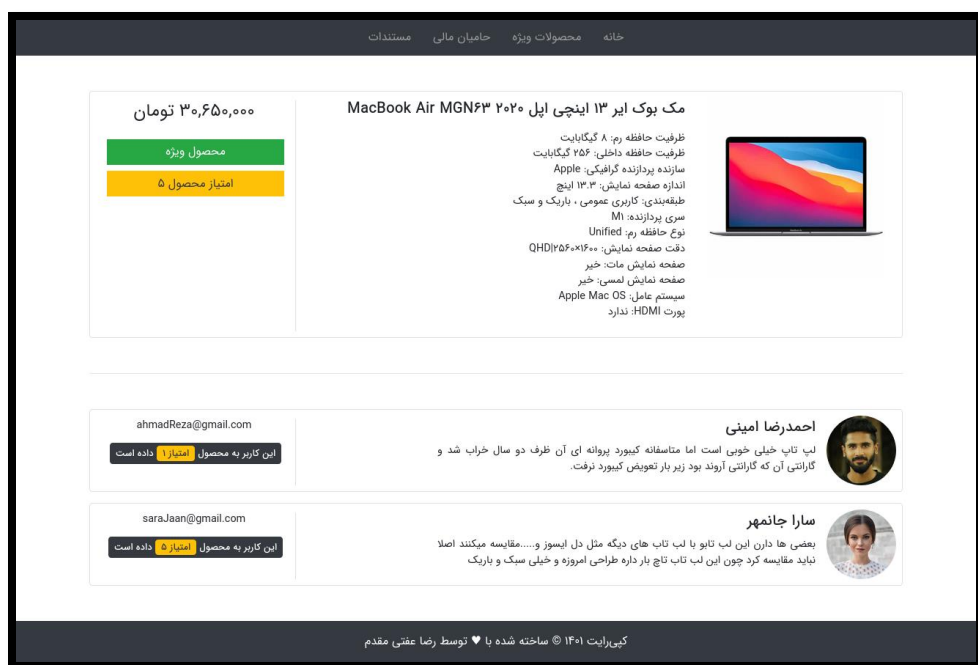
در این صفحه، محصولات ویژه، یعنی محصولاتی که برچسب محصول ویژه دارند، نمایش داده می شود.

صفحه حامیان مالی - <http://127.0.0.1:8000/sponsors>



در این صفحه، تمامی حامیان مالی فروشگاه لیست میشوند.

صفحه محصول - <http://127.0.0.1:8000/products/1>



در این صفحه، یک محصول با جزئیات بیشتری نمایش داده می شود و اگر کاربری برای محصول نظری گذاشته باشد می توانیم مشاهده کنیم.

صفحه مستندات - http://127.0.0.1:8000/documents

خانه

مستندات

تیم

درباره ما

تماس با ما

How to use it?

This is a fairly simple web service for a store website. using api is very simple and need no authentication, this type is the best choice to working with if you start using api recently.

Products

Get all products

```
fetch('http://127.0.0.1:8000/api/products')
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

Get product by ID

```
fetch('http://127.0.0.1:8000/api/products/2')
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

Get featured products

```
fetch('http://127.0.0.1:8000/api/products/featured')
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

Delete product by ID

```
fetch('http://127.0.0.1:8000/api/products/2')
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

GET

DELETE

POST

PUT

خانه

مستندات

تیم

درباره ما

تماس با ما

Update product by ID

```
fetch('http://localhost:8000/api/products/2', {
  method: 'PUT',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    title: 'آیفون 13 پرو 256GB مدل آبی',
    content: 'آیفون 13 پرو 256GB مدل آبی',
    price: '48,500,000 تومان',
    featured: 0,
  })
})
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

Post product

```
fetch('http://localhost:8000/api/products', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    title: 'لپتاپ MacBook Air M1 Pro 16 اینچ',
    content: 'لپتاپ MacBook Air M1 Pro 16 اینچ',
    price: '28,500,000 تومان',
    featured: 1,
  })
})
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

Comments

Get comments by ID

```
fetch('http://127.0.0.1:8000/api/comments/3')
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

Post comment

```
fetch('http://localhost:8000/api/comments/3', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    name: 'علی محمدی',
    email: 'ali.mohamadi@gmail.com',
    avatar: 'https://avatars.githubusercontent.com/u/56348113?v=4',
    comment: 'عالی دایم بود.'
  })
})
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

GET

DELETE

POST

PUT

خانه

مستندات

تیم

درباره ما

تماس با ما

Sponsors

Get all sponsors

```
fetch('http://127.0.0.1:8000/api/sponsors')
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

Post sponsor

```
fetch('http://localhost:8000/api/sponsors', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    name: 'شرکت توسعه وب ایران',
    url: 'https://www.iranweb.com',
    logo: 'https://avatars.githubusercontent.com/u/56348113?v=4'
  })
})
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

Delete sponsor by ID

```
fetch('http://127.0.0.1:8000/api/sponsors/2')
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

Update sponsor by ID

```
fetch('http://localhost:8000/api/sponsors/1', {
  method: 'PUT',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    name: 'شرکت توسعه وب ایران',
    url: 'https://www.iranweb.com',
    logo: 'https://avatars.githubusercontent.com/u/56348113?v=4'
  })
})
  .then(res => res.json())
  .then(json => console.log(json))
```

[Try it](#)

© تمامی حقوق محفوظ است | توسط تیم توسعه وب ایران

GET

DELETE

POST

PUT

در صفحه مستندات، تمامی جزئیات وب سرویس به صورت دقیق و عملی توضیح داده شده است. چگونگی دریافت، ارسال، حذف یا ویرایش محصول با سایر آیتم‌ها (کامنت‌ها، حامیان مالی و ...) به صورت دقیق توضیح داده شده است.

تحلیل سورس کد



```
1 Route::get('/products', [ProductController::class, 'index'])->name('products.get');
2 Route::post('/products', [ProductController::class, 'add_product'])->name('products.add');
3 Route::get('/products/featured', [ProductController::class, 'get_feature'])->name('products.feature.get');
4 Route::get('/products/{id}', [ProductController::class, 'get_product'])->name('products.details.get');
5 Route::delete('/products/{product_id}', [ProductController::class, 'delete_product'])->name('products.delete');
6 Route::put('/products/{product_id}', [ProductController::class, 'edit_product'])->name('products.edit');
7
8 Route::post('/comments/{product_id}', [CommentController::class, 'add_comment'])->name('comments.add');
9 Route::get('/comments/{product_id}', [CommentController::class, 'index'])->name('comments.get');
10
11 Route::get('/sponsors', [SponsorController::class, 'index'])->name('sponsors.get');
12 Route::post('/sponsors', [SponsorController::class, 'add_sponsor'])->name('sponsors.add');
13 Route::delete('/sponsors/{id}', [SponsorController::class, 'delete_sponsor'])->name('sponsors.delete');
14 Route::put('/sponsors/{id}', [SponsorController::class, 'edit_sponsor'])->name('sponsors.edit');
```

لیست مسیرهای API


```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Comment;
6 use App\Models\Product;
7 use Illuminate\Http\Request;
8
9 class ProductController extends Controller
10 {
11     public function index()
12     {
13         $products = Product::all();
14         return response()->json(["status" => 200, "success" => true, "products" => $products], 200);
15     }
16
17     public function get_product($id)
18     {
19         $product = Product::findOrFail($id);
20         return response()->json(["status" => 200, "success" => true, "product" => $product], 200);
21     }
22
23     public function get_feature()
24     {
25         $featured = Product::where('isFeatured', '1')->get();
26         return response()->json(["status" => 200, "success" => true, "featured" => $featured], 200);
27     }
28
29     public function edit_product(Request $request, $id)
30     {
31         $product = Product::findOrFail($id);
32         $product->title = $request->title;
33         $product->content = $request->content;
34         $product->image_url = $request->image;
35         $product->price = $request->price;
36         $product->score = $request->score;
37         $product->isFeatured = $request->isFeatured;
38         $product->save();
39
40         return response()->json(["status" => 200, "success" => true, "message" => "Editing Product Done"], 200);
41     }
42
43     public function add_product(Request $request)
44     {
45         $product = new Product();
46         $product->title = $request->title;
47         $product->content = $request->content;
48         $product->image_url = $request->image;
49         $product->price = $request->price;
50         $product->score = $request->score;
51         $product->isFeatured = $request->isFeatured;
52         $product->save();
53
54         return response()->json(["status" => 201, "success" => true, "message" => "Adding Product Done"], 201);
55     }
56
57     public function delete_product($id)
58     {
59         $product = Product::where('id', $id)->first();
60         $comments = Comment::where('product_id', $id);
61
62         $product->delete();
63         $comments->delete();
64
65         return response()->json(["status" => 202, "success" => true, "message" => "Deleting Product Done"], 202);
66     }
67 }
68

```

توابع مرتبط با حذف، ارسال، دریافت و ویرایش محصول - ProductController

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Sponsor;
6  use Illuminate\Http\Request;
7
8  class SponsorController extends Controller
9  {
10     public function index()
11     {
12         $sponsors = Sponsor::all();
13         return response()->json(["status" => 200, "success" => true, "sponsors" => $sponsors], 200);
14     }
15
16     public function add_sponsor(Request $request)
17     {
18         $sponsor = new Sponsor();
19         $sponsor->name = $request->name;
20         $sponsor->url = $request->url;
21         $sponsor->img = $request->img;
22         $sponsor->save();
23
24         return response()->json(["status" => 201, "success" => true, "message" => "Adding Sponsor Done"], 201);
25     }
26
27     public function delete_sponsor($id)
28     {
29         $sponsor = Sponsor::where('id', $id)->first();
30         $sponsor->delete();
31
32         return response()->json(["status" => 202, "success" => true, "message" => "Deleting Sponsor Done"], 202);
33     }
34
35     public function edit_sponsor(Request $request, $id)
36     {
37         $sponsor = Sponsor::findOrFail($id);
38         $sponsor->name = $request->name;
39         $sponsor->url = $request->url;
40         $sponsor->img = $request->img;
41         $sponsor->save();
42
43         return response()->json(["status" => 200, "success" => true, "message" => "Editing Sponsor Done"], 200);
44     }
45 }
46

```

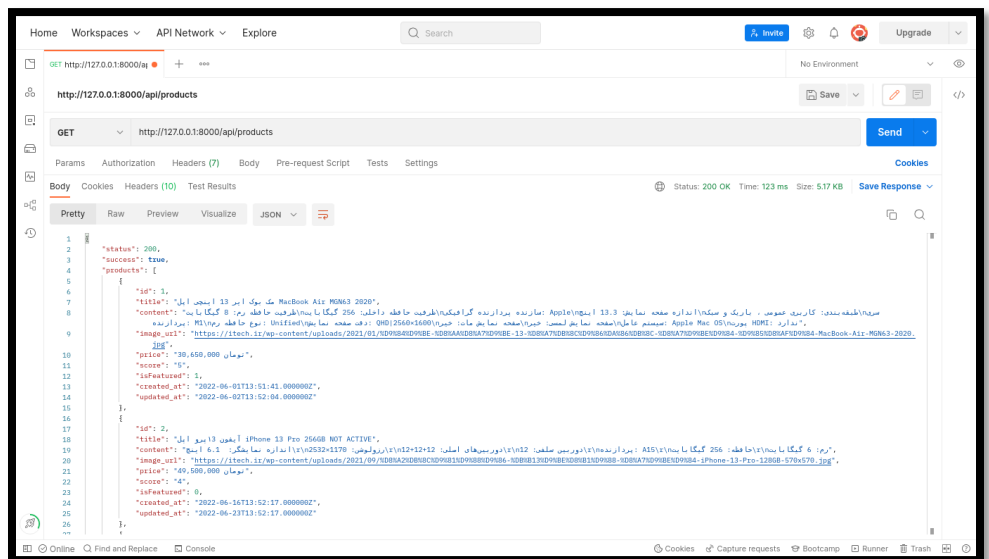
توابع مرتبط با حذف، ارسال، دریافت و ویرایش حامیان مالی - SponsorController

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Comment;
6 use Illuminate\Http\Request;
7
8 class CommentController extends Controller
9 {
10     public function add_comment($id, Request $request)
11     {
12         $comment = new Comment();
13         $comment->product_id = $id;
14         $comment->sender_name = $request->name;
15         $comment->sender_email = $request->email;
16         $comment->sender_avatar = $request->avatar;
17         $comment->score = $request->score;
18         $comment->comment = $request->comment;
19         $comment->save();
20
21         return response()->json(["status" => 201, "success" => true, "message" => "Adding Comment Done"], 201);
22     }
23
24     public function index($id)
25     {
26         $comment = Comment::where('product_id', $id)->get();
27         return response()->json(["status" => 200, "success" => true, "comments" => $comment], 200);
28     }
29 }
30
```

توابع مرتبط با ارسال و دریافت نظرات - CommentController

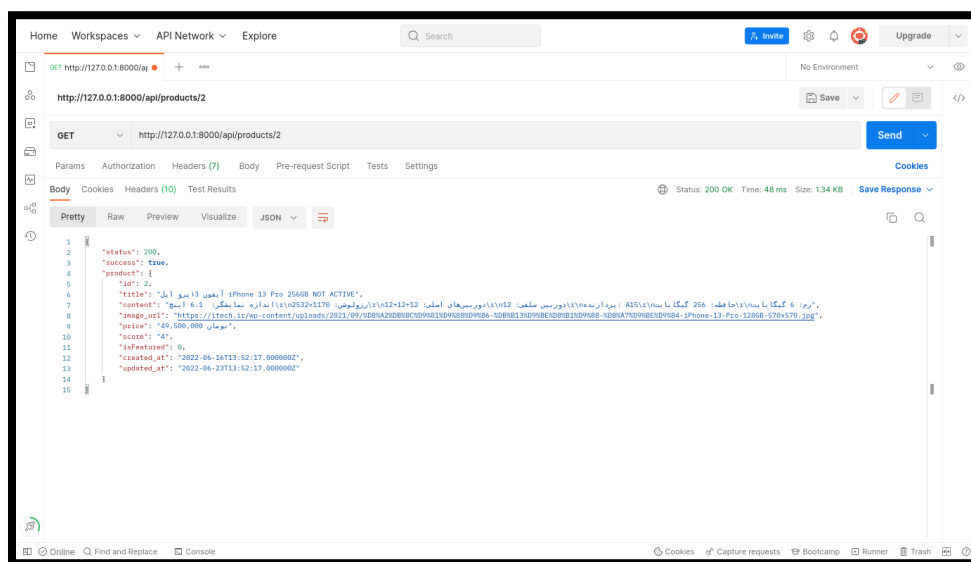
تست وب سرویس با استفاده از نرم افزار Postman

دریافت تمامی محصولات



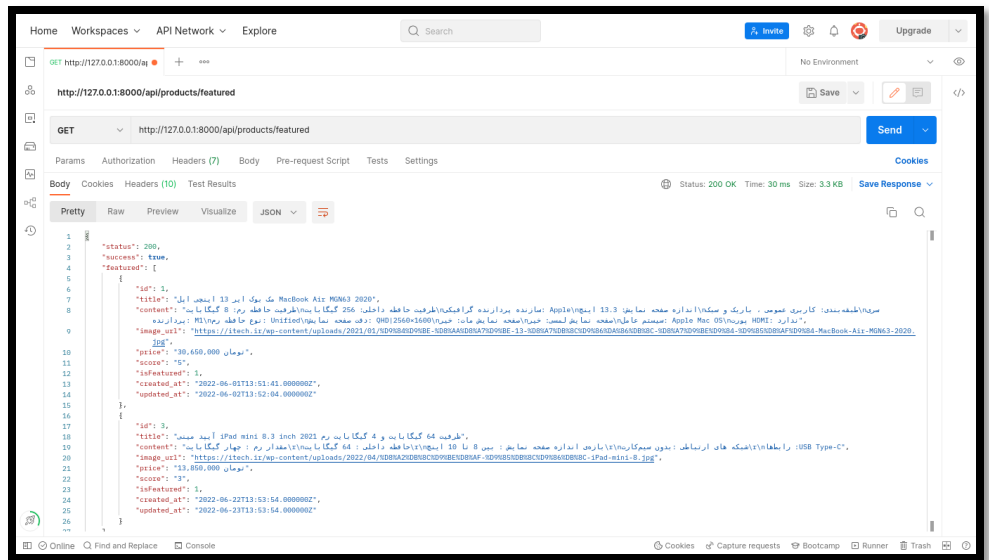
آدرس <http://127.0.0.1:8000/api/products> را با متد GET فراخوانی کردم و خروجی JSON مشاهده شده در تصویر که همان محصولات جدول Products هستند را مشاهده می‌کنیم.

دریافت محصول با آیدی ID



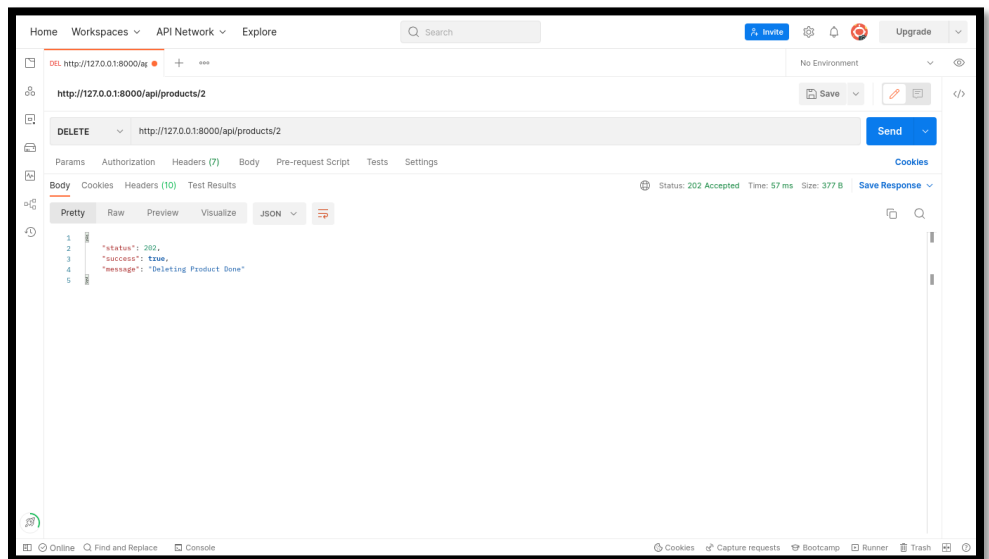
آدرس <http://127.0.0.1:8000/api/products/2> را با متد GET فراخوانی کردم و خروجی JSON مشاهده شده در تصویر که همان محصول با id=2 در جدول Products است را مشاهده می‌کنیم.

دریافت محصولات ویژه



آدرس <http://127.0.0.1:8000/api/products/featured> را با متد **GET** فراخوانی کردم و خروجی JSON مشاهده شده در تصویر که همان محصولاتی که isFeatured=1 در جدول Products هستند را مشاهده می‌کنیم.

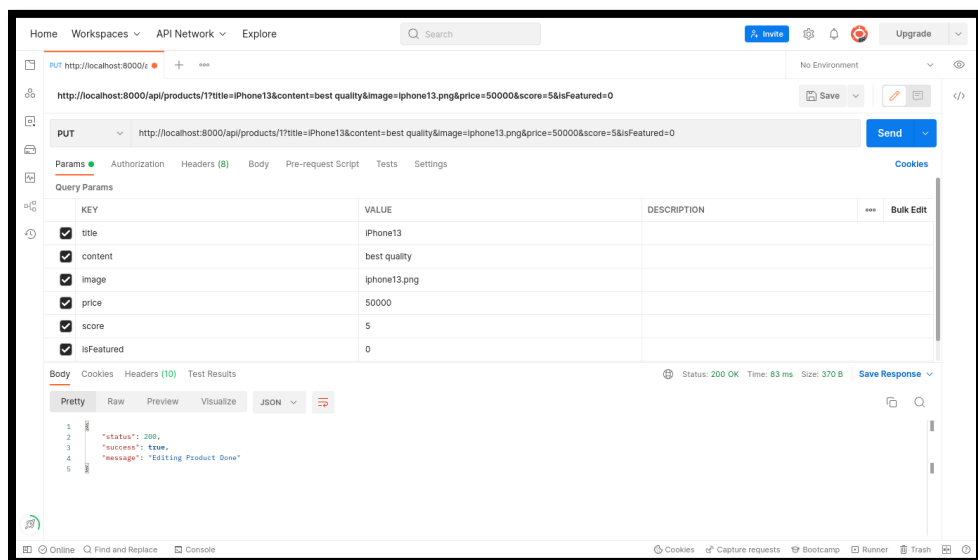
حذف محصول با آیدی ID



آدرس <http://127.0.0.1:8000/api/products/2> را با متد **DELETE** فراخوانی کردم و خروجی JSON مشاهده شده در تصویر نشان می‌دهد که محصول id=2 با موفقیت حذف شد.

توجه کنید، با حذف محصول تمامی نظرات ارسال شده در مورد آن محصول هم حذف می‌شوند.

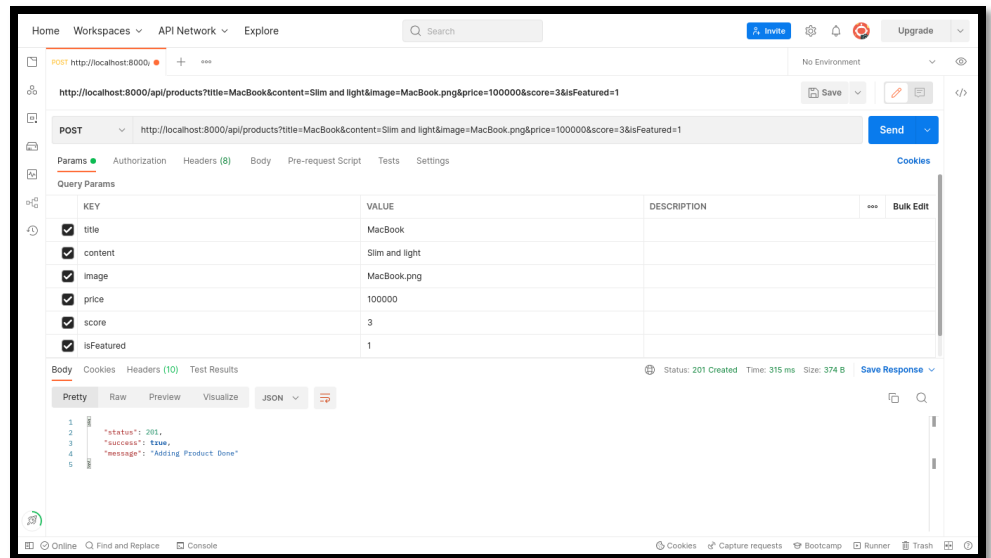
ویرایش محصول با آیدی ID



آدرس <http://127.0.0.1:8000/api/products/1> را با متد **PUT** فراخوانی کردم و خروجی JSON مشاهده شده در تصویر نشان می‌دهد که محصول `id=1` با موفقیت ویرایش شد.

<http://localhost:8000/api/products/1?title=iPhone13&content=best quality&image=iphone13.png&price=50000&score=5&isFeatured=0>

ارسال محصول



آدرس <http://127.0.0.1:8000/api/products> را با متد **POST** فراخوانی کردم و خروجی JSON مشاهده شده در تصویر نشان می‌دهد که محصول با موفقیت افزوده شد.

<http://localhost:8000/api/products?title=MacBook&content=Slim and light&image=MacBook.png&price=100000&score=3&isFeatured=1>

نکته : روند کار برای جداول نظرات و حامیان مالی مشابه جدول محصولات است و تمامی این موارد به صورت دقیق و با جزئیات، درون صفحه مستندات ذکر شده است.

<http://127.0.0.1:8000/documents>