

# Multitenancy with Single Database

Step 1 :- Create a Model For handle tenants(like Stores)

Ex: the tenants of the project is Stores

```
Schema::create('stores', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('domain')->unique(); // like elabd.store.com
    $table->timestamps();
});
```

Step 3 :- create a Middleware to get the host name of the request and then select only the store ids from each table

create a middleware called TenantMiddleware

```
class TenantMiddleware
{
    // Handle an incoming request.
    // Parameters: Closure($IlluminateHttpRequest): (Response) $next
    public function handle(Request $request, Closure $next): Response
    {
        $domain = $request->getHost(); // Like a.localhost
        $tenant = \App\Models\Store::where('domain', $domain)->firstOrFail(); // Select * from stores where domain = a.localhost
        if (!$tenant) { // If tenant not found Like the main tenant (localhost)
            return $next($request);
        }
        app()->instance('store.active', $tenant); // add tenant to Service container
        return $next($request);
    }
}
```

Step 5 : add this Trait to any Model you want to apply for it the Tenancy

Add this trait to Product Model

```
class Product extends Model
{
    use HasFactory;
    use \App\Traits\StoreScope;
}
```

Ex: if i have a specific Products for each store

```
Schema::create('products', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('description');
    $table->foreignId('category_id')->constrained('categories')->cascadeOnDelete()->cascadeOnUpdate();
    $table->foreignId('store_id')->constrained('stores')->cascadeOnDelete()->cascadeOnUpdate();
    $table->float('price');
    $table->timestamps();
});
```

Step 2 :- for each Model for your project if you want to make it to be special for each tenant , you should add the foreign id column of the store

add global scope for Product Model to get only the products of this tenancy

Step 1 :- create a trait called StoreScope

Step 2 :- add this code for the trait :-

```
trait StoreScope
{
    no usages
    public static function bootStoreScope()
    {
        static::addGlobalScope('store', function (Builder $builder) {
            $store = app()->make('store.active');
            if ($store) {
                $builder->where('store_id', $store->id);
            }
        });
    }

    public function store()
    {
        return $this->belongsTo('related: Store::class');
    }
}
```

Step 4 :- add a global Scope for each model you want to apply the tenancy on it

```
< > http://localhost:3000
1 // 20240910034950
2 // http://a.localhost:8080/
3
4 {
5   "id": 13,
6   "name": "Tatyana Yillman",
7   "description": "Porro quia quia ut nihil est animi a expedita. Dolor quo sit delectus ut. Laudantium optio aliquam occaecati qu",
8   "category_id": 20,
9   "store_id": 61,
10  "price": 11429,
11  "created_at": "2024-09-09T20:56:17.000000Z",
12  "updated_at": "2024-09-09T20:56:17.000000Z"
13 },
14 {
15   "id": 15,
16   "name": "Mina Wansch",
17   "description": "Ture quos sed reprehenderit. Eligendi commodi facilis eveniet voluptates consequatur assumenda eveniet laborum.",
18   "category_id": 23,
19   "store_id": 61,
20   "price": 1684,
21   "created_at": "2024-09-09T20:56:17.000000Z",
22   "updated_at": "2024-09-09T20:56:17.000000Z"
23 }
24 ]
```

```
< > http://localhost:3000
1 // 20240910034700
2 // http://d.localhost:8080/
3
4 {
5   "id": 10,
6   "name": "Betsy Bashirian",
7   "description": "Dolores repudiandae non accusantium magni dolorum. Nemo reiciendis enim et praesentium. Reiciendis quia aut et eius.",
8   "category_id": 15,
9   "store_id": 64,
10  "price": 12230,
11  "created_at": "2024-09-09T20:56:17.000000Z",
12  "updated_at": "2024-09-09T20:56:17.000000Z"
13 }
14 ]
```

The result