

Model Definitions

Comment:

```
User.php M Comment.php M X DatabaseSeeder.php M Com
app > Models > Comment.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Comment extends Model
9  {
10     use HasFactory;
11     public function post() {
12         return $this->belongsTo(Post::class);
13     }
14
15     public function user() {
16         return $this->belongsTo(User::class);
17     }
18 }
19
```

Post:

```
Tag.php X
app > Models > Tag.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Tag extends Model
9  {
10     use HasFactory;
11     public function posts() {
12         return $this->hasMany(Post::class);
13     }
14 }
15
```

User:

Note: This is the default user model added by breeze, which I have modified to add the posts() and comments() methods.

```
User.php M X
app > Models > User.php
1  <?php
2
3  namespace App\Models;
4
5  // use Illuminate\Contracts\Auth\MustVerifyEmail;
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Illuminate\Notifications\Notifiable;
9  use Laravel\Sanctum\HasApiTokens;
10
11 class User extends Authenticatable
12 {
13     use HasApiTokens, HasFactory, Notifiable;
14     public function posts() {
15         return $this->hasMany(Post::class);
16     }
17
18     public function comments() {
19         return $this->hasMany(Comment::class);
20     }
21     /**
22      * The attributes that are mass assignable.
23      *
24      * @var array<int, string>
25      */
26     protected $fillable = [
27         'name',
28         'email',
29         'password',
30     ];
31     /**
32      * The attributes that should be hidden for serialization.
33      *
34      * @var array<int, string>
35      */
36     protected $hidden = [
37         'password',
38         'remember_token',
39     ];
40     /**
41      * The attributes that should be cast.
42      *
43      * @var array<string, string>
44      */
45     protected $casts = [
46         'email_verified_at' => 'datetime',
47     ];
48 }
```

Migrations:

User:

Note: This is the default users migration supplied by breeze

```
User.php M 2014_10_12_000000_create_users_table.php X
database > migrations > 2014_10_12_000000_create_users_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->id();
18             $table->string('name');
19             $table->string('email')->unique();
20             $table->timestamp('email_verified_at')->nullable();
21             $table->string('password');
22             $table->rememberToken();
23             $table->timestamps();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      *
30      * @return void
31      */
32     public function down()
33     {
34         Schema::dropIfExists('users');
35     }
36 };
37
```

Posts:

```
User.php M 2022_11_05_172446_create_posts_table.php X
database > migrations > 2022_11_05_172446_create_posts_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('posts', function (Blueprint $table) {
17             $table->id();
18             $table->bigInteger('user_id')->references('id')->on('users')
19                 ->onDelete('cascade')->onUpdate('cascade');
20             $table->string('contents');
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      *
28      * @return void
29      */
30     public function down()
31     {
32         Schema::dropIfExists('posts');
33     }
34 };
35
```

Comments:

```
User.php M 2022_11_05_172500_create_comments_table.php X
database > migrations > 2022_11_05_172500_create_comments_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('comments', function (Blueprint $table) {
17             $table->id();
18             $table->bigInteger('post_id')->references('id')->on('posts')
19                 ->onDelete('cascade')->onUpdate('cascade');
20             $table->bigInteger('user_id')->references('id')->on('users')
21                 ->onDelete('cascade')->onUpdate('cascade');
22             $table->string('contents');
23             $table->timestamps();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      *
30      * @return void
31      */
32     public function down()
33     {
34         Schema::dropIfExists('comments');
35     }
36 };
37
```


Tags:

```
User.php M 2022_11_05_172512_create_tags_table.php X
database > migrations > 2022_11_05_172512_create_tags_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('tags', function (Blueprint $table) {
17             $table->id();
18             $table->string('name');
19             $table->timestamps();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      *
26      * @return void
27      */
28     public function down()
29     {
30         Schema::dropIfExists('tags');
31     }
32 };
33
```

Post tag pivot table:

```
User.php M 2022_11_06_180153_create_post_tag_table.php X
database > migrations > 2022_11_06_180153_create_post_tag_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('post_tag', function (Blueprint $table) {
17             $table->primary(['post_id', 'tag_id']);
18             $table->bigInteger('post_id')->unsigned();
19             $table->bigInteger('tag_id')->unsigned();
20             $table->timestamps();
21
22             $table->foreign('post_id')->references('id')->on('posts')
23                 ->onDelete('cascade')->onUpdate('cascade');
24             $table->foreign('tag_id')->references('id')->on('tags')
25                 ->onDelete('cascade')->onUpdate('cascade');
26         });
27     }
28
29     /**
30      * Reverse the migrations.
31      *
32      * @return void
33      */
34     public function down()
35     {
36         Schema::dropIfExists('post_tag');
37     }
38 };
39
```

Seeders

Database seeder:

```
User.php M DatabaseSeeder.php M X
database > seeders > DatabaseSeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
5  // use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7
8  class DatabaseSeeder extends Seeder
9  {
10     /**
11      * Seed the application's database.
12      *
13      * @return void
14      */
15     public function run()
16     {
17         $this->call(UserTableSeeder::class);
18         $this->call(TagTableSeeder::class);
19         $this->call(PostTableSeeder::class);
20         $this->call(CommentTableSeeder::class);
21     }
22 }
23
```


User table seeder:

```
database > seeders > UserTableSeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use App\Models\User;
8  use App\Models\Post;
9  use App\Models\Comment;
10 use App\Models\Tag;
11
12 class UserTableSeeder extends Seeder
13 {
14     /**
15      * Run the database seeds.
16      *
17      * @return void
18      */
19     public function run()
20     {
21         $test = new User;
22         $test->name = "Rachel";
23         $test->email = "Rachel@fake.com";
24         $test->email_verified_at = now();
25         $test->password = "GoodPassword123";
26         $test->save();
27
28         User::factory()->count(10)->create();
29     }
30 }
31
```

Tag table seeder:

```
User.php M DatabaseSeeder.php M TagTableSeeder.php X
database > seeders > TagTableSeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use App\Models\Tag;
8
9  class TagTableSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      *
14      * @return void
15      */
16     public function run()
17     {
18         $test = new Tag;
19         $test->name = "New posts get the new post tag";
20         $test->save();
21
22         Tag::factory()->count(10)->create();
23     }
24 }
25
```

Post table seeder:

```
User.php M DatabaseSeeder.php M PostTableSeeder.php X
database > seeders > PostTableSeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use App\Models\Post;
8  use App\Models\Tag;
9
10 class PostTableSeeder extends Seeder
11 {
12     /**
13      * Run the database seeds.
14      *
15      * @return void
16      */
17     public function run()
18     {
19         $testTag = Tag::find(1);
20         Post::factory()->hasAttached($testTag)->create(
21             ['contents' => "This is a test post!", 'user_id' => 1]);
22
23         // Generate an array of a random number
24         // of existing tags, with no duplicates
25         $randomTags = function() {
26             $tags = [];
27             $usedtags = [];
28             $i = mt_rand(1,5);
29             for ($x = 0; $x < $i; $x++) {
30                 $newId = mt_rand(2, Tag::count());
31                 while (in_array($newId, $usedtags)) {
32                     $newId = mt_rand(2, Tag::count());
33                 }
34                 array_push($tags, Tag::where('id', $newId)->get()[0]);
35                 array_push($usedtags, $newId);
36             }
37             return $tags;
38         };
39
40         // Neet to loop to execute randomTags() again on every iteration
41         for ($x = 0; $x < 50; $x++) {
42             Post::factory()->hasAttached($randomTags())->create();
43         }
44     }
45 }
```

Comment table seeder:

```
database > seeders > CommentTableSeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use App\Models\Comment;
8
9  class CommentTableSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      *
14      * @return void
15      */
16     public function run()
17     {
18         $test = new Comment;
19         $test->contents = "This is a test comment on the test post!";
20         $test->user_id = 1;
21         $test->post_id = 1;
22         $test->save();
23
24         Comment::factory()->count(50)->create();
25     }
26 }
27
```

Factories

User factory:

Note: This is the default user factory provided. I have not modified it.

```
User.php M DatabaseSeeder.php M CommentTableSeeder.php M UserFactory.php X
database > factories > UserFactory.php
1  <?php
2
3  namespace Database\Factories;
4
5  use Illuminate\Database\Eloquent\Factories\Factory;
6  use Illuminate\Support\Str;
7
8  /**
9   * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\User>
10  */
11  class UserFactory extends Factory
12  {
13      /**
14       * Define the model's default state.
15       *
16       * @return array<string, mixed>
17       */
18      public function definition()
19      {
20          return [
21              'name' => fake()->name(),
22              'email' => fake()->unique()->safeEmail(),
23              'email_verified_at' => now(),
24              'password' => '$2y$10$92IXUNpkj00r0Q5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', // password
25              'remember_token' => Str::random(10),
26          ];
27      }
28
29      /**
30       * Indicate that the model's email address should be unverified.
31       *
32       * @return static
33       */
34      public function unverified()
35      {
36          return $this->state(fn (array $attributes) => [
37              'email_verified_at' => null,
38          ]);
39      }
40  }
41
```


Tag factory:

```
User.php M DatabaseSeeder.php M CommentTableSeeder.php M TagFactory.php X
database > factories > TagFactory.php
3 namespace Database\Factories;
4
5 use Illuminate\Database\Eloquent\Factories\Factory;
6
7 /**
8  * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Tag>
9  */
10 class TagFactory extends Factory
11 {
12     /**
13      * Define the model's default state.
14      *
15      * @return array<string, mixed>
16      */
17     public function definition()
18     {
19         return [
20             'name' => fake()->word()
21         ];
22     }
23 }
```

Post factory:

```
database > factories > PostFactory.php
1  <?php
2
3  namespace Database\Factories;
4
5  use Illuminate\Database\Eloquent\Factories\Factory;
6  use App\Models\User;
7
8  /**
9   * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Post>
10  */
11  class PostFactory extends Factory
12  {
13      /**
14       * Define the model's default state.
15       *
16       * @return array<string, mixed>
17       */
18      public function definition()
19      {
20          return [
21              'contents' => fake()->text(),
22              'user_id' => rand(1, User::count())
23          ];
24      }
25  }
26
```

Comment factory:

```
User.php M DatabaseSeeder.php M CommentTableSeeder.php M CommentFactory.php X
database > factories > CommentFactory.php
1  <?php
2
3  namespace Database\Factories;
4
5  use Illuminate\Database\Eloquent\Factories\Factory;
6  use App\Models\Post;
7  use App\Models\User;
8
9  /**
10   * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Comment>
11   */
12  class CommentFactory extends Factory
13  {
14      /**
15       * Define the model's default state.
16       *
17       * @return array<string, mixed>
18       */
19      public function definition()
20      {
21          return [
22              'post_id' => rand(1, Post::count()),
23              'user_id' => rand(1, User::count()),
24              'contents' => fake()->text()
25          ];
26      }
27  }
28
```