2031927 – CSF304: Web App Development – Lab Class 3

Ashley Eatly

2031927

Table of Contents

CSF304 Web Application Development– Lab Class
Randell Gaya

CSF304 Web App Development – Lab Class 3   05/10/2022

Ü Task 3.1

Create a new migration and model to store People. The table should be named `people` and the model should be named `Person`.

Hint: The artisan command
```
php artisan make:model NameOfModel -m
```
will create both the model `NameOfModel` and the corresponding migration (named correctly).
To see all available options use the command
```
php artisan help make:model
```

Yes, we could use the pre-existing model `User` for this. But, let's avoid using that until we get to authentication and the ability to login to the website.
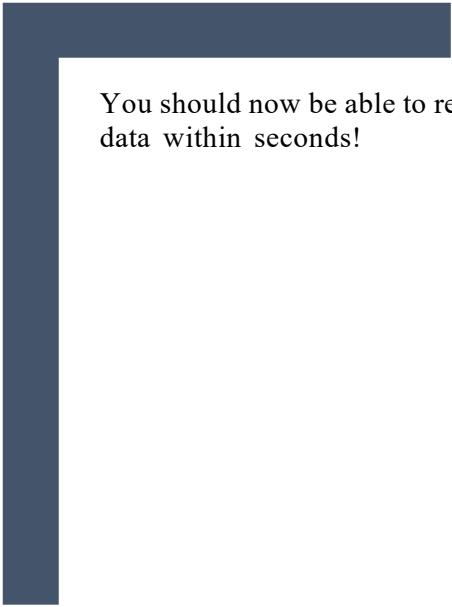
A Person should have:

  ï Title (optional).

  ï First Name.

  ï Surname.

  ï Address.

  ï Two fields of your choice.

Once the above is working, complete the following within Tinker. Take screen-shots to allow for a quick sign-off:

  ï Create some manual models (at least 2).

  ï Find a model by id.

  ï Cause an error by trying to find a non-existent model by id.

  ï Use the `::all()` and `::first()` methods.

  ï Use `::all()` and then extract (`get`) the model at index 1.

  ï Delete a model by calling the `delete` method on it. Show it doesn't exist by trying to find it again.

Ü Task 3.2

Create a seed file for creating a few hard-coded models (at least 2). Once you have this working, create a factory that produces fairly realistic models. Add a call to the factory to your seed file so that you bulk create People when seeding.

You should now be able to reset your database and be back up and running with realistic sample data within seconds!

Task 3.1

	Create a new migration and model to store People.	Completed Y/N
1	A model is equivalent to an object on a table – an instance of the item. In this case the Model is Person and the table will be automatically pluralised to People (usually) A Person should have: •Title (optional). •First Name. •Surname. •Address. •Two fields of your choice.	
2	Depending on the way the environment has been configured the commands for creating an empty model migration (no fields) can be of various forms	
	php artisan make:model Person -m sail artisan make:model Person -m (note -m can be in full --migrate	
	Don't forget run up the containers/services by ./vendor/bin/sail up -d (or if you have created an alias sail up -d (-d to run in background)	
	Also note if you then do sail artisan help make:model.. you will get a large table of help..	

3	sail up -d **sail artisan make:model Person --migration**	

	ashleyeatly@v0019227 lab3a-app % **sail artisan make:model Person -m**
	  INFO Model [app/Models/Person.php] created successfully.
	  INFO Created migration
	[2022_10_17_143808_create_people_table].	

	``php
	Inspection..
	
	2022_10_17_143808_create_people_table
	
	<?php
	
	use Illuminate\Database\Migrations\Migration;
	use Illuminate\Database\Schema\Blueprint;
	use Illuminate\Support\Facades\Schema;
	
	return new class extends Migration
	{
	    /**
	     * Run the migrations.
	     *
	     * @return void
	     */
	    public function up()
	    {
	        Schema::create('people', function (Blueprint $table) {
	            $table->id();
	            $table->timestamps();
	        });
	    }
	
	    /**
	     * Reverse the migrations.
	     *
	     * @return void
	     */
	    public function down()
	    {
	
	        Schema::dropIfExists('people');
	    }
	};
	``	

	``` This creates a bare bones up which is run to create the table and the row definitions and the down to reverse thye migration and delete the table. ```	
	Adding the required row definitions to **up** and adding two other types a date and Boolean Note timestamps() will create two timestamp rows	

	``php
	public function up()
	{
	    Schema::create('people', function (Blueprint $table)
	{
	        $table->id();
	``	

```
        $table->string(column: 'title')->nullable();
        $table->string(column: 'firstname');
        $table->string(column: 'surname');
        $table->text(column: 'address');
        $table->date(column:'date_of_birth')->nullable();
        $table->boolean(column: 'current');
        $table->timestamps();
    });
}
```

sail artisan migrate to run the migration

ashleyeatly@v0019227 lab3a-app % **sail artisan migrate**
  INFO Preparing database.
  Creating migration table
....
116ms DONE
  INFO Running migrations.
  2014_10_12_000000_create_users_table 90ms DONE
  2014_10_12_100000_create_password_resets_table 33ms DONE
  2019_08_19_000000_create_failed_jobs_table 36ms DONE
  2019_12_14_000001_create_personal_access_tokens_table 58ms
DONE
  2022_10_17_143808_create_people_table 19ms DONE

	Note the default app contains 4 default migrations – could have deleted	
3	In PHP Storm database ∨ 〰 lab3a_app@localhost 1 of 4 　∨ 🖧 lab3a_app 　　∨ 📁 tables 6 　　　> ⊞ failed_jobs 　　　> ⊞ migrations 　　　> ⊞ password_resets 　　　∨ ⊞ people 　　　　∨ 📁 columns 9 　　　　　🔑 id bigint unsigned (auto increment) 　　　　　⊞ title varchar(255) 　　　　　⊞ firstname varchar(255) 　　　　　⊞ surname varchar(255) 　　　　　⊞ address text 　　　　　⊞ date_of_birth date 　　　　　⊞ current tinyint(1) 　　　　　⊞ created_at timestamp 　　　　　⊞ updated_at timestamp 　　　　> 📁 keys 1 　　　　> 📁 indexes 1 　　　> ⊞ personal_access_tokens 　　　> ⊞ users 　　> 🖿 Server Objects	

	Start using Tinker	
	Create some manual models (at least 2).	
	ashleyeatly@v0019227 lab3a-app % sail artisan tinker Psy Shell v0.11.8 (PHP 8.1.11 — cli) by Justin Hileman >>>	
	use App\Models\Person; $a = new Person(); $a->title='Mr'; $a->firstname='Ashley'; $a->surname='Eatly'; $a->address='Ferryside Carmarthenshire'; $a->date_of_birth=date('1959-03-31'); $a->current=true; $a->created_at=now(); $a->updated_at=now(); $a->save();	
	$a = new Person(); $a->title='Mr'; $a->firstname='First'; $a->surname='SurName'; $a->address='Ferryside Dyfed'; $a->date_of_birth=date('1970-02-19'); $a->current=true; $a->created_at=now(); $a->updated_at=now(); $a->save();	
	$a = new Person(); $a->title="mr"; $a->firstname='First1'; $a->surname='SurName1'; $a->address='Ferryside Dyfed SA17 5TT'; $a->date_of_birth=date('1989-05-10'); $a->current=true; $a->created_at=now(); $a->updated_at=now(); $a->save();	
	$a = new Person(); $a->title='mr'; $a->firstname='First1'; $a->surname='SurName1'; $a->address='Ferryside Dyfed SA17 5TT'; $a->date_of_birth=date('1989-05-10'); $a->current=true; $a->created_at=now(); $a->updated_at=now(); $a->save();	

	Find a model by id.	Done
| | ```
>>> Person::find(1);
=> App\Models\Person {#4615
    id: 1,
    title: null,
    firstname: "Ashley",
    surname: "Eatly",
    address: "Ferryside Carmarthenshire",
    date_of_birth: "1959-03-31",
    current: 1,
    created_at: "2022-10-17 15:44:28",
    updated_at: "2022-10-17 15:44:28",
  }
``` | |
| | Cause an error by trying to find a non-existent model by id. | Done |
| | ```
>>> Person::find(10);
=> null
``` | |
| | Use the ::all() and ::first() methods. | Done |
| | ::all() | Done part 1 |
| | ```
>>> Person::all();
=> Illuminate\Database\Eloquent\Collection {#4605
    all: [
      App\Models\Person {#4617
        id: 1,
        title: null,
        firstname: "Ashley",
        surname: "Eatly",
        address: "Ferryside Carmarthenshire",
        date_of_birth: "1959-03-31",
        current: 1,
        created_at: "2022-10-17 15:44:28",
        updated_at: "2022-10-17 15:44:28",
      },
      App\Models\Person {#4618
        id: 2,
        title: null,
        firstname: "First",
        surname: "SurName",
        address: "Ferryside Dyfed",
        date_of_birth: "1970-02-19",
``` | |

```
    current: 1,
    created_at: "2022-10-17 15:49:36",
    updated_at: "2022-10-17 15:49:36",
   },
   App\Models\Person {#4619
    id: 3,
    title: "Mr",
    firstname: "First",
    surname: "SurName",
    address: "Ferryside Dyfed",
    date_of_birth: "1970-02-19",
    current: 1,
    created_at: "2022-10-17 15:50:50",
    updated_at: "2022-10-17 15:50:50",
   },
   App\Models\Person {#4613
    id: 4,
    title: null,
    firstname: "First1",
    surname: "SurName1",
    address: "Ferryside Dyfed SA17 5TT",
    date_of_birth: "1989-05-10",
    current: 1,
    created_at: "2022-10-17 15:54:45",
    updated_at: "2022-10-17 15:54:45",
   },
   App\Models\Person {#4607
    id: 5,
    title: null,
    firstname: "First1",
    surname: "SurName1",
    address: "Ferryside Dyfed SA17 5TT",
    date_of_birth: "1989-05-10",
    current: 1,
    created_at: "2022-10-17 15:56:05",
    updated_at: "2022-10-17 15:56:05",
   },
   App\Models\Person {#4353
    id: 6,
    title: "mr",
    firstname: "First1",
    surname: "SurName1",
    address: "Ferryside Dyfed SA17 5TT",
    date_of_birth: "1989-05-10",
    current: 1,
    created_at: "2022-10-17 15:57:24",
    updated_at: "2022-10-17 15:57:24",
```

| ```
    },
   ],
   }
``` | |
| Person::first(); | Done part 2 |
| ```
>>> Person::first();
=> App\Models\Person {#4603
      id: 1,
      title: null,
      firstname: "Ashley",
      surname: "Eatly",
      address: "Ferryside Carmarthenshire",
      date_of_birth: "1959-03-31",
      current: 1,
      created_at: "2022-10-17 15:44:28",
      updated_at: "2022-10-17 15:44:28",
    },
:
``` | |
| Use ::all() and then extract (get) the model at index 1. | Done |
| ```
>>> Person::all()->get(1);
=> App\Models\Person {#4605
     id: 2,
     title: null,
     firstname: "First",
     surname: "SurName",
     address: "Ferryside Dyfed",
     date_of_birth: "1970-02-19",
     current: 1,
     created_at: "2022-10-17 15:49:36",
     updated_at: "2022-10-17 15:49:36",
   }
``` | |
| Note index 1 is not id 1 but the second item in the collection that is returned by ::all() | |
| If you want the first entry then use<br><br>```
>>> Person::all()->get(0);
=> App\Models\Person {#4353
     id: 1,
     title: null,
     firstname: "Ashley",
     surname: "Eatly",
     address: "Ferryside Carmarthenshire",
     date_of_birth: "1959-03-31",
     current: 1,
``` | |

created_at: "2022-10-17 15:44:28",
    updated_at: "2022-10-17 15:44:28",
  }		

•Delete a model by calling the delete method on it. Show it doesn't exist by trying to find it again.	Done	

>>> Person::all()->get(1)->delete();
=> true
 	
>>> Person::find(2);
=> null		

Using database access		
1,,Ashley,Eatly,Ferryside Carmarthenshire,1959-03-31,1,2022-10-17 15:44:28 2,,First,SurName,Ferryside Dyfed,1970-02-19,1,2022-10-17 15:49:36 3,Mr,First,SurName,Ferryside Dyfed,1970-02-19,1,2022-10-17 15:50:50 4,,First1,SurName1,Ferryside Dyfed SA17 5TT,1989-05-10,1,2022-10-17 15:54:45 5,,First1,SurName1,Ferryside Dyfed SA17 5TT,1989-05-10,1,2022-10-17 15:56:05 6,mr,First1,SurName1,Ferryside Dyfed SA17 5TT,1989-05-10,1,2022-10-17 15:57:24 1,,Ashley,Eatly,Ferryside Carmarthenshire,1959-03-31,1,2022-10-17 15:44:28 3,Mr,First,SurName,Ferryside Dyfed,1970-02-19,1,2022-10-17 15:50:50 4,,First1,SurName1,Ferryside Dyfed SA17 5TT,1989-05-10,1,2022-10-17 15:54:45 5,,First1,SurName1,Ferryside Dyfed SA17 5TT,1989-05-10,1,2022-10-17 15:56:05 6,mr,First1,SurName1,Ferryside Dyfed SA17 5TT,1989-05-10,1,2022-10-17 15:57:24		

Task 3.2

Ü Task 3.2

Create a seed file for creating a few hard-coded models (at least 2). Once you have this working, create a factory that produces fairly realistic models. Add a call to the factory to your seed file so that you bulk create People when seeding.

You should now be able to reset your database and be back up and running with realistic sample data within seconds!

	Create a seed file for creating a few hard-coded models (at least 2).	Completed Y/N
1	Sail artisan make:seeder PersonSeeder	Done
| | ```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class PersonSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        //
    }
}
``` | |

Adding the values

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class PersonSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        \App\Models\Person::factory(50) ->create([
            'title' => 'Test Title 1',
            'firstname' => 'First name 1',
            'surname' => 'SurName 1',
            'address' => 'Address 1 where 1',
            'date_of_birth' => date('2010-03-31'),
            'current' => true,
            'created_at' => now(),
            'updated_at' => now(),
        ]);
//          \App\Models\Person::factory(50) ->create();

    }
}
```

`sail artisan db:seed --class=PersonSeeder`	Done

```
ashleyeatly@v0019227 lab3a-app % sail artisan db:seed --
class=PersonSeeder

   INFO  Seeding database.
```

```
I re-ran this as the printout was bad and used

sail artisan migrate:fresh --seed

1,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
2,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
3,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
4,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
5,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
6,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
```

```
7,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
8,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
9,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
10,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
11,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
12,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
13,Test Title 1,First name 1,SurName 1,Address 1 where 1,2010-03-
31,1,2022-10-27 10:22:06,2022-10-27 10:22:06
```

Once you have this working, create a factory that produces fairly realistic models.

```
sail artisan make:factory PersonFactory
```

```php
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;

/**
 * @extends
\Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Model
>
 */
class PersonFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition()
    {
        return [
            //
        ];
    }
}
```

Adding the row definitions

```php
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;

/**
 * @extends
\Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Person>
 */
class PersonFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition()
    {
        return [
            'title' => fake()->title(),
            'firstname' => fake()->firstName(),
            'surname' => fake()->lastName(),
            'address' => fake()->address(),
            'date_of_birth' => fake()->date(),
            'current' => fake()->boolean(50),
            'created_at' => fake()->date(),
            'updated_at' => fake()->date(),
        ];
    }
}
```

Remove the original seeders

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class PersonSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        \App\Models\Person::factory(50) ->create();

    }
}
```

```
ashleyeatly@v0019227 lab3a-app % sail artisan migrate:fresh --seed

  Dropping all tables
.................................................................................
......................................... 409ms DONE
```

```
   INFO  Preparing database.

  Creating migration table ..............................57ms DONE

   INFO  Running migrations.

  2014_10_12_000000_create_users_table ................. 79ms DONE
  2014_10_12_100000_create_password_resets_table
.......................................................32ms DONE
  2019_08_19_000000_create_failed_jobs_table
.......................................................41ms DONE
  2019_12_14_000001_create_personal_access_tokens_table . 55ms DONE
  2022_10_17_143808_create_people_table
.......................................................20ms DONE

   INFO  Seeding database.

  Database\Seeders\PersonSeeder
......................................................RUNNING
  Database\Seeders\PersonSeeder
.......................................................735.16 ms
DONE

ashleyeatly@v0019227 lab3a-app %
```

```
1,Dr.,Sophia,Hintz,"336 Mills Heights
South Icie, TX 49120-1771",1988-10-16,0,1996-10-07 00:00:00,1975-12-
11 00:00:00
2,Prof.,Juliet,Cassin,"77180 Marks Cliff
Kreigerport, TN 62537",1995-01-06,1,1972-01-26 00:00:00,2017-07-05
00:00:00
3,Dr.,Lamont,Witting,"817 Edmond Wall
Alberthastad, GA 37241",1999-12-07,1,2014-07-02 00:00:00,1993-04-07
00:00:00
4,Dr.,Lavinia,Orn,"74139 General Gardens Apt. 548
Homenickbury, LA 75453-3143",1991-09-28,0,2020-05-19 00:00:00,1992-
10-05 00:00:00
5,Dr.,Alycia,Hirthe,"450 Bergstrom Lights
West Arloborough, VT 24251",2002-05-23,0,2006-02-06 00:00:00,2012-12-
10 00:00:00
6,Prof.,Marcelle,Smitham,"20981 Rhianna Hollow
Lutherton, AR 87623",1994-07-05,1,1982-02-10 00:00:00,2010-12-17
00:00:00
7,Mr.,Dortha,Swaniawski,"65478 Schaefer Station Suite 454
Paolochester, AK 07860-7394",2007-01-05,0,2000-09-04 00:00:00,2008-
02-13 00:00:00
8,Mr.,Floy,Homenick,"42668 Elmira Inlet
Bruenland, OK 47434-9296",1985-10-18,0,2011-10-03 00:00:00,1988-09-05
00:00:00
9,Mrs.,Kacie,Klocko,"1944 Zieme Squares Suite 111
Port Skylar, CO 00121-4799",2020-09-02,0,1996-03-27 00:00:00,1990-11-
26 00:00:00
10,Prof.,Dominique,Lang,"2830 Daugherty Throughway
East Mercedesshire, PA 34020-9767",1985-03-17,0,1985-06-15
00:00:00,2003-09-28 00:00:00
11,Dr.,Edgardo,Bashirian,"7196 Zulauf Glen Apt. 524
West Bell, FL 43558-5067",2019-03-12,0,2016-06-29 00:00:00,1980-05-10
00:00:00
12,Dr.,Rickie,Mohr,"63968 Sigurd Ville
Dariusfurt, MD 46400-3051",1978-04-20,1,1994-09-17 00:00:00,2012-08-
16 00:00:00
```

13,Mr.,Meagan,Klein,"6181 Fay Common
Hermistonport, KS 75087-8609",2011-05-21,1,1986-07-18 00:00:00,1988-
10-23 00:00:00