

# Curso de Laravel 5.3

## CODECASTS (Fábio Vedovelli)

[https://www.youtube.com/watch?v=25JnwUAH99U&list=PLy5T05I\\_eQYM4-B6W374JkAuh\\_wOrs3Pi](https://www.youtube.com/watch?v=25JnwUAH99U&list=PLy5T05I_eQYM4-B6W374JkAuh_wOrs3Pi)

Resumo do curso feito por Roberto Pinheiro

## Aula 01 - Introdução

### Framework Laravel

É uma caixa de ferramentas que serve para trabalhar com PHP.

O Laravel permite criar de uma forma muito rápida um CRUD protegido por senha

### O que será visto no curso

- Artisan
- Database
  - Migrations, Factories, Seeds
  - Eloquent, Query Builder e Collection
  - Relationship
- Helpers
- Routes e Middleware
- Requests, Responses e Flash Messages
- Views, Layouts e Assets
- Forms e Validation
- Debur
- Controllers e Repositories
- Cache
- Elixir

## Aula 02 - Criando o projeto

A instalação é feita através de uma ferramenta chamada **Composer**, que é um gerenciador de dependências para o PHP.

Instale o **Composer** em seu computador.

Há duas formas de instalar o Laravel:

### Via Laravel Installer

O comando a seguir instala, de forma global, o Laravel no computador.

```
composer global require laravel/installer
```

O comando a seguir cria o projeto:

```
laravel new <nome_projeto>
```

### Via Composer Create-Project

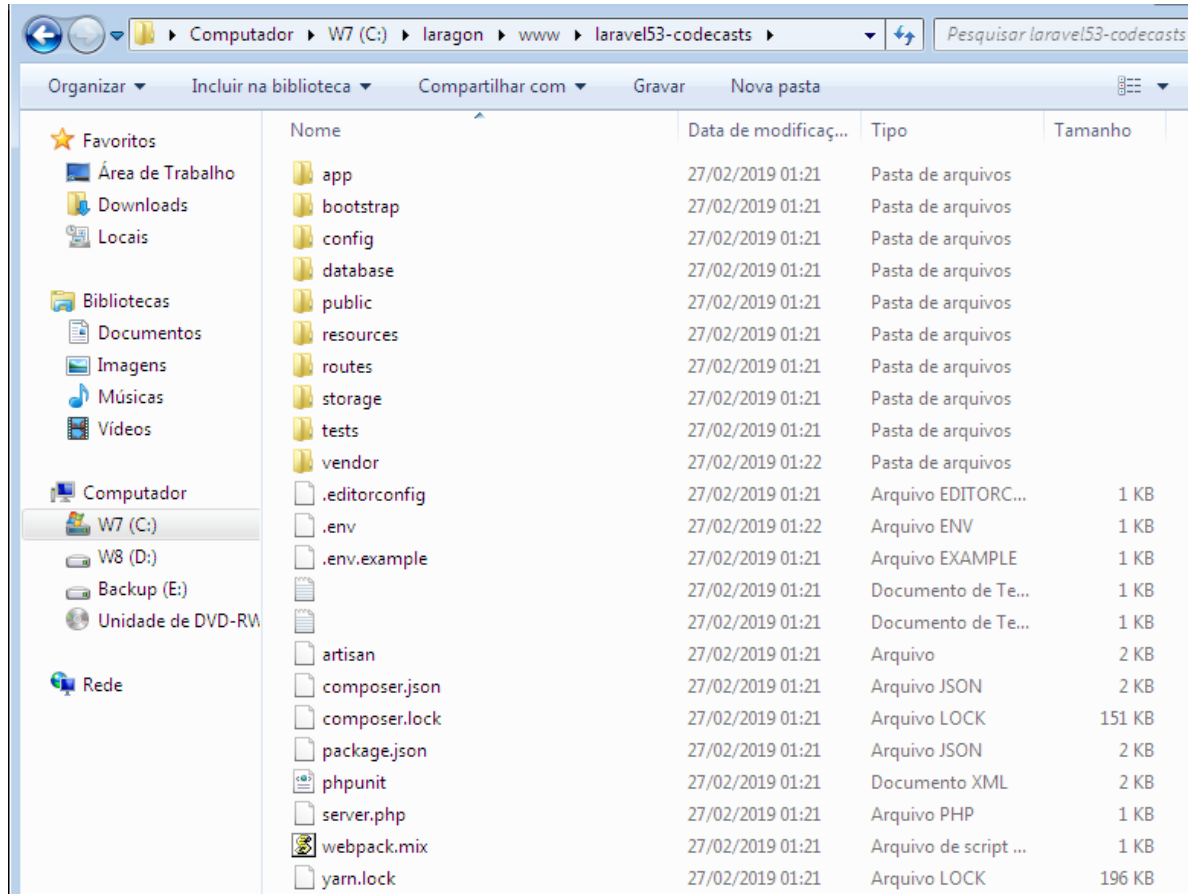
O comando abaixo instala o Laravel e cria o projeto na pasta ativa:

```
composer create-project --prefer-dist laravel/laravel <nome_projeto>
```

## Instalação do Laravel e criação do projeto

composer global require laravel/installer

laravel new laravel53-codecasts



## Aula 05 - Planejamento

Será desenvolvido uma aplicação para controle de clientes e suas compras.

A aplicação precisa de um nome!

O que é necessário criar para que a funcionalidade dos clientes seja completa:

### CLIENTES (Customers)

- Migrations
- Model
- Factory
- Seed
- Routes
- Controller
- Views
- LaravelCollective/  
HTML
- Validator
- Repositório
- Serviço

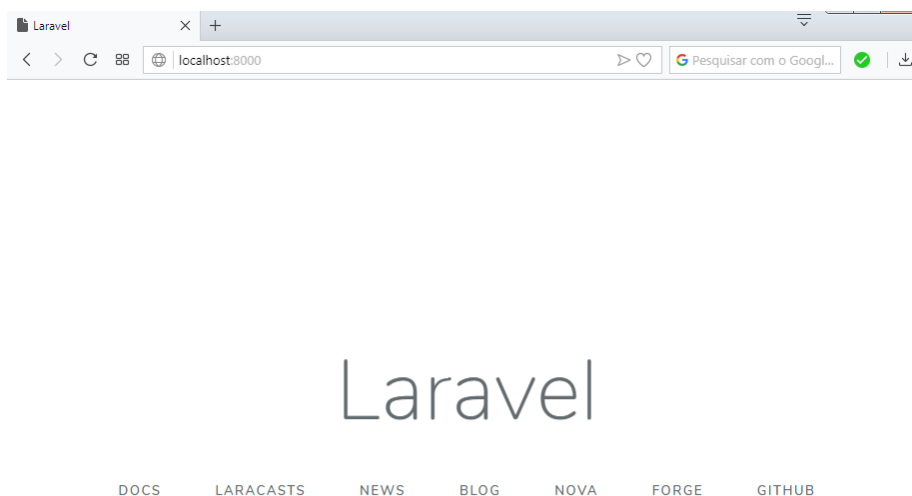
## Aula 06 - Nomear a aplicação

### Iniciando o servidor

php artisan serve

No browser:

localhost:8000



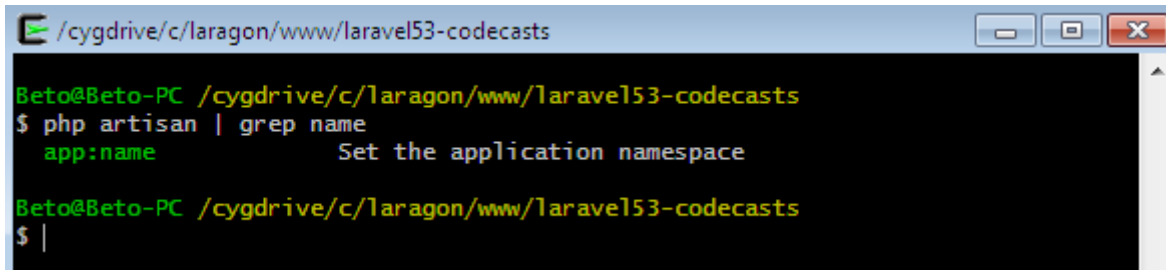
### Terminal Bash no Windows

Para a permitir que várias versões do Microsoft Windows possam, de certa forma, agir como um sistema Unix você pode experimentar o **CygWin**, que é uma coleção de ferramentas de software livre originalmente desenvolvidas por **Cygnus Solutions**. Sua principal intenção é portar softwares que rodam em sistemas **POSIX** (como sistemas Linux, sistemas **BSD**, e sistemas Unix) para que rodem em Windows com pouco mais do que uma recompilação. O Cygwin é atualmente mantido por funcionários da Red Hat e outras pessoas.



### Usando o Bash para filtrar comandos do artisan

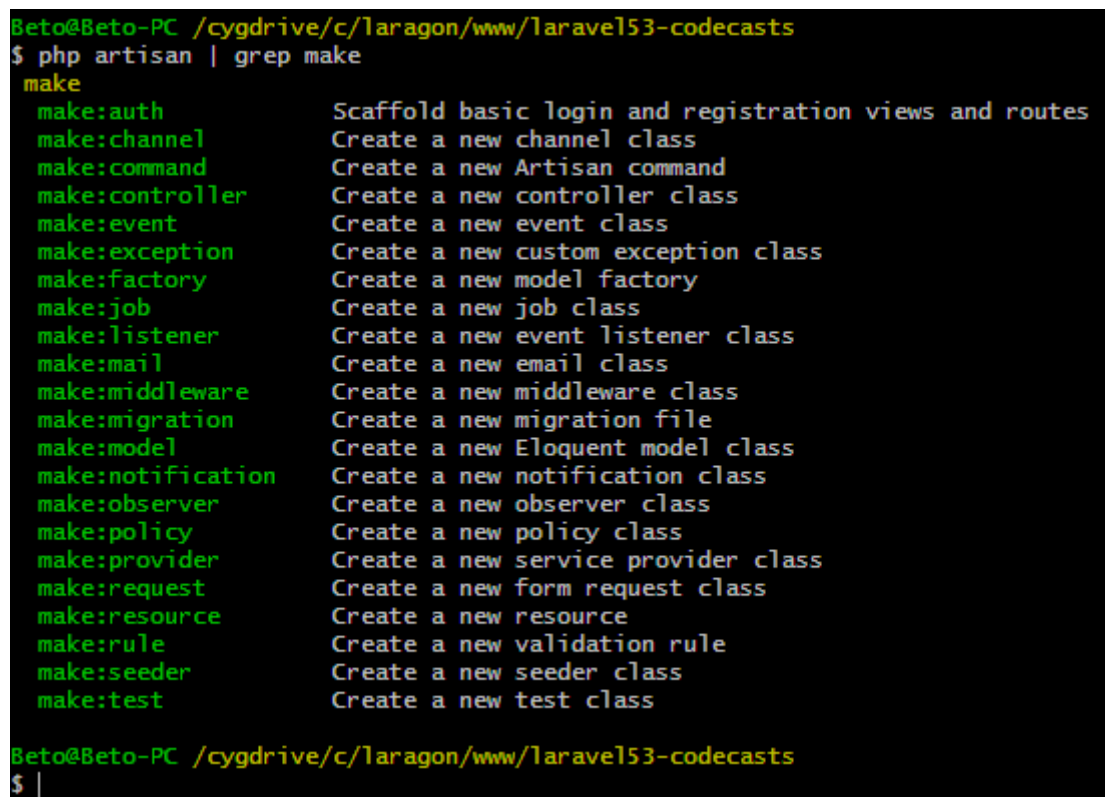
php artisan | grep name



```
/cygdrive/c/laragon/www/laravel53-codecasts
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan | grep name
  app:name          Set the application namespace

Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ |
```

php artisan | grep make



```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan | grep make
make
make:auth          Scaffold basic login and registration views and routes
make:channel        Create a new channel class
make:command        Create a new Artisan command
make:controller     Create a new controller class
make:event          Create a new event class
make:exception      Create a new custom exception class
make:factory        Create a new model factory
make:job            Create a new job class
make:listener       Create a new event listener class
make:mail           Create a new email class
make:middleware     Create a new middleware class
make:migration       Create a new migration file
make:model          Create a new Eloquent model class
make:notification   Create a new notification class
make:observer       Create a new observer class
make:policy         Create a new policy class
make:provider       Create a new service provider class
make:request        Create a new form request class
make:resource       Create a new resource
make:rule           Create a new validation rule
make:seeder         Create a new seeder class
make:test          Create a new test class

Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ |
```

## Aula 07 - Migrations - Overview

### Criando o banco de dados

Acesse o phpmyadmin e crie o banco de dados com o nome **laravel53-codecasts**

### Configurando o acesso ao banco de dados

#### Arquivo .env

```
DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=laravel53-codecasts
DB_USERNAME=root
DB_PASSWORD=
```

### Migrations padrões

O Laravel já vem com duas migrations padrões:

```
2014_10_12_000000_create_users_table.php
2014_10_12_100000_create_password_resets_table.php
```

**database/migrations/2014\_10\_12\_000000\_create\_users\_table.php**

```
<?php
```

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```



## Criando as tabelas

Para criar as tabelas:

```
php artisan migrate
```

```
|>> php artisan migrate  
Migration table created successfully.  
Migrated: 2014_10_12_000000_create_users_table  
Migrated: 2014_10_12_100000_create_password_resets_table
```

## Apagando as tabelas

Para voltar atrás e remover as tabelas:

```
php artisan migrate:rollback
```

```
/Users/vedovelli/Codecasts/basico  
>> php artisan migrate:rollback  
Rolled back: 2014_10_12_100000_create_password_resets_table  
Rolled back: 2014_10_12_000000_create_users_table
```

## Tipos de campos - Schema

A classe schema contém uma variedade de tipos de campos que podem ser especificados na construção das tabelas:

Command	Description
<code>\$table-&gt;bigIncrements('id');</code>	Auto-incrementing UNSIGNED BIGINT (primary key) equivalent column.
<code>\$table-&gt;bigInteger('votes');</code>	BIGINT equivalent column.
<code>\$table-&gt;binary('data');</code>	BLOB equivalent column.
<code>\$table-&gt;boolean('confirmed');</code>	BOOLEAN equivalent column.
<code>\$table-&gt;char('name', 100);</code>	CHAR equivalent column with an optional length.
<code>\$table-&gt;date('created_at');</code>	DATE equivalent column.
<code>\$table-&gt;dateTime('created_at');</code>	DATETIME equivalent column.
<code>\$table-&gt;dateTimeTz('created_at');</code>	DATETIME (with timezone) equivalent column.
<code>\$table-&gt;decimal('amount', 8, 2);</code>	DECIMAL equivalent column with a precision (total digits) and scale (decimal digits).
<code>\$table-&gt;double('amount', 8, 2);</code>	DOUBLE equivalent column with a precision (total digits) and scale (decimal digits).
<code>\$table-&gt;enum('level', ['easy', 'hard']);</code>	ENUM equivalent column.
<code>\$table-&gt;float('amount', 8, 2);</code>	FLOAT equivalent column with a precision (total digits) and scale (decimal digits).
<code>\$table-&gt;geometry('positions');</code>	GEOMETRY equivalent column.
<code>\$table-&gt;geometryCollection('positions');</code>	GEOMETRYCOLLECTION equivalent column.
<code>\$table-&gt;increments('id');</code>	Auto-incrementing UNSIGNED INTEGER (primary key) equivalent column.
<code>\$table-&gt;integer('votes');</code>	INTEGER equivalent column.
<code>\$table-&gt;ipAddress('visitor');</code>	IP address equivalent column.
<code>\$table-&gt;json('options');</code>	JSON equivalent column.

Command	Description
<code>\$table-&gt;jsonb('options');</code>	JSONB equivalent column.
<code>\$table-&gt;lineString('positions');</code>	LINESTRING equivalent column.
<code>\$table-&gt;longText('description');</code>	LONGTEXT equivalent column.
<code>\$table-&gt;macAddress('device');</code>	MAC address equivalent column.
<code>\$table-&gt;mediumIncrements('id');</code>	Auto-incrementing UNSIGNED MEDIUMINT (primary key) equivalent column.
<code>\$table-&gt;mediumInteger('votes');</code>	MEDIUMINT equivalent column.
<code>\$table-&gt;mediumText('description');</code>	MEDIUMTEXT equivalent column.
<code>\$table-&gt;morphs('taggable');</code>	Adds taggable_id UNSIGNED BIGINT and taggable_type VARCHAR equivalent columns.
<code>\$table-&gt;multiLineString('positions');</code>	MULTILINESTRING equivalent column.
<code>\$table-&gt;multiPoint('positions');</code>	MULTIPOINT equivalent column.
<code>\$table-&gt;multiPolygon('positions');</code>	MULTIPOLYGON equivalent column.
<code>\$table-&gt;nullableMorphs('taggable');</code>	Adds nullable versions of morphs() columns.
<code>\$table-&gt;nullableTimestamps();</code>	Alias of timestamps() method.
<code>\$table-&gt;point('position');</code>	POINT equivalent column.
<code>\$table-&gt;polygon('positions');</code>	POLYGON equivalent column.
<code>\$table-&gt;rememberToken();</code>	Adds a nullable remember_token VARCHAR(100) equivalent column.
<code>\$table-&gt;smallIncrements('id');</code>	Auto-incrementing UNSIGNED SMALLINT (primary key) equivalent column.
<code>\$table-&gt;smallInteger('votes');</code>	SMALLINT equivalent column.
<code>\$table-&gt;softDeletes();</code>	Adds a nullable deleted_at TIMESTAMP equivalent column for soft deletes.
<code>\$table-&gt;softDeletesTz();</code>	Adds a nullable deleted_at TIMESTAMP (with timezone) equivalent column for soft deletes.
<code>\$table-&gt;string('name', 100);</code>	VARCHAR equivalent column with a optional length.

Command	Description
<code>\$table-&gt;text('description');</code>	TEXT equivalent column.
<code>\$table-&gt;time('sunrise');</code>	TIME equivalent column.
<code>\$table-&gt;timeTz('sunrise');</code>	TIME (with timezone) equivalent column.
<code>\$table-&gt;timestamp('added_on');</code>	TIMESTAMP equivalent column.
<code>\$table-&gt;timestampTz('added_on');</code>	TIMESTAMP (with timezone) equivalent column.
<code>\$table-&gt;timestamps();</code>	Adds nullable created_at and updated_at TIMESTAMP equivalent columns.
<code>\$table-&gt;timestampsTz();</code>	Adds nullable created_at and updated_at TIMESTAMP (with timezone) equivalent columns.
<code>\$table-&gt;tinyIncrements('id');</code>	Auto-incrementing UNSIGNED TINYINT (primary key) equivalent column.
<code>\$table-&gt;tinyInteger('votes');</code>	TINYINT equivalent column.
<code>\$table-&gt;unsignedBigInteger('votes');</code>	UNSIGNED BIGINT equivalent column.
<code>\$table-&gt;unsignedDecimal('amount', 8, 2);</code>	UNSIGNED DECIMAL equivalent column with a precision (total digits) and scale (decimal digits).
<code>\$table-&gt;unsignedInteger('votes');</code>	UNSIGNED INTEGER equivalent column.
<code>\$table-&gt;unsignedMediumInteger('votes');</code>	UNSIGNED MEDIUMINT equivalent column.
<code>\$table-&gt;unsignedSmallInteger('votes');</code>	UNSIGNED SMALLINT equivalent column.
<code>\$table-&gt;unsignedTinyInteger('votes');</code>	UNSIGNED TINYINT equivalent column.
<code>\$table-&gt;uuid('id');</code>	UUID equivalent column.
<code>\$table-&gt;year('birth_year');</code>	YEAR equivalent column.

## Aula 08 - Migrations específicas

### Filtrando os comandos migrate

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan | grep migrate
  migrate                Run the database migrations
migrate
  migrate:fresh           Drop all tables and re-run all migrations
  migrate:install         Create the migration repository
  migrate:refresh         Reset and re-run all migrations
  migrate:reset           Rollback all database migrations
  migrate:rollback        Rollback the last database migration
  migrate:status          Show the status of each migration
```

### Criando as migrations

php artisan make:migration create\_customers\_table --create=customers

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan make:migration create_customers_table --create=customers
Created Migration: 2019_02_27_211109_create_customers_table

Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ |
```

php artisan make:migration create\_purchases\_table --create=purchases

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan make:migration create_purchases_table --create=purchases
Created Migration: 2019_02_27_212327_create_purchases_table

Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ |
```

**database/migrations/2019\_02\_27\_211109\_create\_customers\_table.php**

```
<?php
```

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreateCustomersTable extends Migration
{
    public function up()
    {
        Schema::create('customers', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name', 150);
            $table->string('city', 50);
            $table->string('state', 2);
            $table->datetime('birthdate');
            $table->boolean('special_customer');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('customers');
    }
}
```

database/migrations/2019\_02\_27\_212327\_create\_purchases\_table.php

<?php

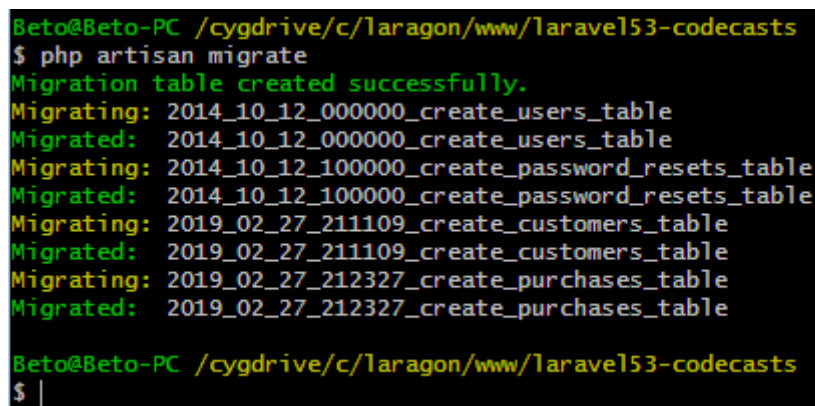
```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreatePurchasesTable extends Migration
{
    public function up()
    {
        Schema::create('purchases', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('customer_id')->unsigned();
            $table->string('amount', 10);
            $table->string('description');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('purchases');
    }
}
```

## Criando as tabelas

php artisan migrate



```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2019_02_27_211109_create_customers_table
Migrated: 2019_02_27_211109_create_customers_table
Migrating: 2019_02_27_212327_create_purchases_table
Migrated: 2019_02_27_212327_create_purchases_table

Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ |
```

## Aula 09 - Model Factories

### Criando as models

php make:model Customer

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan make:model Customer
Model created successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ |
```

php make:model Purchase

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan make:model Purchase
Model created successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ |
```

### Inserindo um registro com o tinker

php artisan tinker

```
C:\laragon\www\laravel53-codecasts>php artisan tinker
Psy Shell v0.9.9 (PHP 7.3.1 64-bit) by Justin Hileman
>>> $c = new \App\Customer;
=> App\Customer {#2921}
>>> $c->name = 'vedovelli';
=> "vedovelli"
>>> $c->city = 'Santo Andre';
=> "Santo Andre"
>>> $c->state='SP';
=> "SP"
>>> $c->special_customer = true;
=> true
>>> $c->birthdate = '1974-08-07 00:00:00';
=> "1974-08-07 00:00:00"
>>> $c->save();
=> true
>>>
```



Showing rows 0 - 0 (1 total, Query took 0.0000 seconds.)

SELECT \* FROM `customers`

Profiling [Edit inline] [Edit] [Explain SQL] [C]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

|   | id | name      | city        | state | birthdate           | special_customer | created_at          | updated_at          |
|---|----|-----------|-------------|-------|---------------------|------------------|---------------------|---------------------|
| <input type="checkbox"/> Edit Copy Delete | 1  | vedovelli | Santo Andre | SP    | 1974-08-07 00:00:00 | 1                | 2019-02-28 02:36:08 | 2019-02-28 02:36:08 |

↑ ☐ Check all With selected: Edit Copy Delete Export

## Criando uma factory

php artisan make:factory CustomerFactory

## Usando a factory para criar registros aleatórios

database/Factory/CustomerFactory.php

```
<?php
```

```
use Faker\Generator as Faker;
```

```
$factory->define(App\Customer::class, function (Faker $faker) {
    return [
        'name' => $faker->name,
        'city' => $faker->city,
        'state' => $faker->stateAbbr,
        'birthdate' => $faker->datetime(),
        'special_customer' => $faker->boolean
    ];
});
```

No terminal:

```
factory(App\Customer::class, 50)->create();
```

```

Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan tinker
Psy Shell v0.9.9 (PHP 7.3.1 - cli) by Justin Hileman
factory(App\Customer::class, 50)->create();
=> Illuminate\Database\Eloquent\Collection {#2948
  all: [
    App\Customer {#2944
      name: "Orville Klocko II",
      city: "Krajcikport",
      state: "NM",
      birthdate: DateTime @592338941 {#2942
        date: 1988-10-08 18:35:41.0 UTC (+00:00),
      },
      special_customer: true,
      updated_at: "2019-02-28 04:08:35",
      created_at: "2019-02-28 04:08:35",
      id: 2,
    },
    App\Customer {#2940
      name: "Geovany Mann",
      city: "Tillmanview",
      state: "UT",
      birthdate: DateTime @1295895589 {#2939
        date: 2011-01-24 18:59:49.0 UTC (+00:00),
      },
      special_customer: true,
      updated_at: "2019-02-28 04:08:35",
      created_at: "2019-02-28 04:08:35",
      id: 3,
    },
    App\Customer {#2941
      name: "Elissa Reilly",
      city: "Lake Everettberg",
      state: "TN",
      birthdate: DateTime @1318948427 {#2943
        date: 2011-10-18 14:33:47.0 UTC (+00:00),
      },
      special_customer: false,
      updated_at: "2019-02-28 04:08:35",
      created_at: "2019-02-28 04:08:35",
      id: 4,
    },
    App\Customer {#2945
      name: "Prof. Madelynn McClure III",
      city: "Skilesland",
      state: "MN",
      birthdate: DateTime @1425004376 {#2947
        date: 2015-02-27 02:32:56.0 UTC (+00:00),
      },
    },
  ],
}

```

Showing rows 0 - 24 (51 total, Query took 0.0030 seconds.)

SELECT \* FROM `customers`

Profiling

[Edit inline]

[Edit]

[Explain SQL]

[Create PHP code]

[Refresh]

1

>

>>

Show all

Number of rows: 25

Filter rows: 

Search this table

Sort by key: None

+ Options

id

name

city

state

birthdate

special\_customer

created\_at

updated\_at

Edit

Copy

Delete

1

vedovelli

Santo Andre

SP

1974-08-07 00:00:00

1

2019-02-28 00:52:07

2019-02-28 00:52:07

Edit

Copy

Delete

2

Orville Klocko II

Krajcikport

NM

1988-10-08 18:35:41

1

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

3

Geovany Mann

Tillmanview

UT

2011-01-24 18:59:49

1

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

4

Elissa Reilly

Lake Everettberg

TN

2011-10-18 14:33:47

0

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

5

Prof. Madelynn McClure III

Skilesland

MN

2015-02-27 02:32:56

1

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

6

Prof. Bailee McLaughlin

Dibbertport

KY

1970-10-31 22:28:44

1

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

7

Mrs. Lorine Hill

Lake Josueton

NM

1970-10-12 21:11:49

0

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

8

Leonor Schmitt

East Christyside

VA

1976-08-08 21:01:01

1

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

9

Salvatore Little

Littletown

LA

1989-08-04 00:34:28

0

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

10

Terrance Green MD

Cletuston

OK

1993-08-07 10:13:41

0

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

11

Tony Dibbert

West Gussiestad

NY

2010-09-27 08:17:29

1

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

12

Emery Barton

Turnerside

FL

2010-09-20 23:27:26

0

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

13

Grayce Grant

Keelingberg

CA

2009-10-18 23:42:49

1

2019-02-28 04:08:35

2019-02-28 04:08:35

Edit

Copy

Delete

14

Bria Braun I

Mollyfurt

DC

2001-06-12 15:35:49

1

2019-02-28 04:08:35

2019-02-28 04:08:35

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan make:factory PurchaseFactory
Factory created successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ |
```

Acesse a URL:

<https://github.com/fzaninotto/Faker>

## Formatters

Each of the generator properties (like name, address, and lorem) are called "formatters". A faker generator has many of them, packaged in "providers". Here is a list of the bundled formatters in the default locale.

### Faker\Provider\Base

```
randomDigit           // 7
randomDigitNotNull    // 5
randomNumber($nbDigits = NULL, $strict = false) // 79907610
randomFloat($nbMaxDecimals = NULL, $min = 0, $max = NULL) // 48.8932
numberBetween($min = 1000, $max = 9000) // 8567
randomLetter          // 'b'

// returns randomly ordered subsequence of a provided array

randomElements($array = array ('a','b','c'), $count = 1) // array('c')
randomElement($array = array ('a','b','c')) // 'b'
shuffle('hello, world') // 'rlo,h eoldlw'
shuffle(array(1, 2, 3)) // array(2, 1, 3)
numerify('Hello ###') // 'Hello 609'
lexify('Hello ???') // 'Hello wgt'
bothify('Hello ##??') // 'Hello 42jz'
asciify('Hello ***') // 'Hello R6+'
regexify('[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}'); // sm0@y8k96a.ej
```

### Faker\Provider\Lorem

```
word                  // 'aut'
words($nb = 3, $asText = false) // array('porro', 'sed', 'magni')
sentence($nbWords = 6, $variableNbWords = true) // 'Sit vitae voluptas sint non voluptates.'
sentences($nb = 3, $asText = false) // array('Optio quos qui illo error.', 'Laborum vero a officia id corporis.', 'Saepe provident esse hic eligendi.')
paragraph($nbSentences = 3, $variableNbSentences = true) // 'Ut ab voluptas sed a nam. Sint autem inventore aut officia aut aut blanditiis. Ducimus eos odit amet et est ut eum.'
paragraphs($nb = 3, $asText = false) // array('Quidem ut sunt et quidem est accusamus aut. Fuga est placeat rerum ut. Enim ex eveniet facere sunt.', 'Aut nam et eum architecto fugit repellendus illo. Qui ex esse
```

```
veritatis.', 'Possimus omnis aut incidunt sunt. Asperiores incidunt iure sequi cum culpa rem. Rerum exercitationem est rem.')
text($maxNbChars = 200) // 'Fuga totam reiciendis qui architecto fugiat nemo. Consequatur recusandae qui cupiditate eos quod.'
```

#### Faker\Provider\en\_US\Person

```
title($gender = null|'male'|'female') // 'Ms.'
titleMale // 'Mr.'
titleFemale // 'Ms.'
suffix // 'Jr.'
name($gender = null|'male'|'female') // 'Dr. Zane Stroman'
firstName($gender = null|'male'|'female') // 'Maynard'
firstNameMale // 'Maynard'
firstNameFemale // 'Rachel'
lastName // 'Zulauf'
```

#### Faker\Provider\en\_US\Address

```
cityPrefix // 'Lake'
secondaryAddress // 'Suite 961'
state // 'NewMexico'
stateAbbr // 'OH'
citySuffix // 'borough'
streetSuffix // 'Keys'
buildingNumber // '484'
city // 'West Judge'
streetName // 'Keegan Trail'
streetAddress // '439 Karley Loaf Suite 897'
postcode // '17916'
address // '8888 Cummings Vista Apt. 101, Susanbury, NY 95473'
country // 'Falkland Islands (Malvinas)'
latitude($min = -90, $max = 90) // 77.147489
longitude($min = -180, $max = 180) // 86.211205
```

#### Faker\Provider\en\_US\PhoneNumber

```
phoneNumber // '201-886-0269 x3767'
tollFreePhoneNumber // '(888) 937-7238'
e164PhoneNumber // '+27113456789'
```

#### Faker\Provider\en\_US\Company

```
catchPhrase // 'Monitored regional contingency'
bs // 'e-enable robust architectures'
company // 'Bogan-Treutel'
companySuffix // 'and Sons'
jobTitle // 'Cashier'
```

#### Faker\Provider\en\_US\Text

```
realText($maxNbChars = 200, $indexSize = 2) // "And yet I wish you could manage it?) 'And what are they made of?' Alice asked in a shrill, passionate voice. 'Would YOU like cats if you were never even spoke to Time!' 'Perhaps not,' Alice replied."
```

## Faker\Provider\DateTime

```
unixTime($max = 'now') // 58781813
dateTime($max = 'now', $timezone = null) // DateTime('2008-04-25 08:37:17', 'UTC')
dateTimeAD($max = 'now', $timezone = null) // DateTime('1800-04-29 20:38:49',
'Europe/Paris')
iso8601($max = 'now') // '1978-12-09T10:10:29+0000'
date($format = 'Y-m-d', $max = 'now') // '1979-06-09'
time($format = 'H:i:s', $max = 'now') // '20:49:42'
dateTimeBetween($startDate = '-30 years', $endDate = 'now', $timezone = null) //
DateTime('2003-03-15 02:00:49', 'Africa/Lagos')
dateTimeInInterval($startDate = '-30 years', $interval = '+ 5 days', $timezone =
null) // DateTime('2003-03-15 02:00:49', 'Antartica/Vostok')
dateTimeThisCentury($max = 'now', $timezone = null) // DateTime('1915-05-30
19:28:21', 'UTC')
dateTimeThisDecade($max = 'now', $timezone = null) // DateTime('2007-05-29
22:30:48', 'Europe/Paris')
dateTimeThisYear($max = 'now', $timezone = null) // DateTime('2011-02-27
20:52:14', 'Africa/Lagos')
dateTimeThisMonth($max = 'now', $timezone = null) // DateTime('2011-10-23
13:46:23', 'Antarctica/Vostok')
amPm($max = 'now') // 'pm'
dayOfMonth($max = 'now') // '04'
dayOfWeek($max = 'now') // 'Friday'
month($max = 'now') // '06'
monthName($max = 'now') // 'January'
year($max = 'now') // '1993'
century // 'VI'
timezone // 'Europe/Paris'
```

Methods accepting a `$timezone` argument default to `date_default_timezone_get()`. You can pass a custom timezone string to each method, or define a custom timezone for all time methods at once using `$faker::setDefaultTimezone($timezone)`.

## Faker\Provider\Internet

```
email // 'tkshlerin@collins.com'
safeEmail // 'king.alford@example.org'
freeEmail // 'bradley72@gmail.com'
companyEmail // 'russel.durward@mcdermott.org'
freeEmailDomain // 'yahoo.com'
safeEmailDomain // 'example.org'
userName // 'wade55'
password // 'k&|X+a45*2['
domainName // 'wolffdeckow.net'
domainWord // 'feeney'
tld // 'biz'
url // 'http://www.skilesdonnelly.biz/aut-accusantium-ut-
architecto-sit-et.html'
slug // 'aut-repellat-commodi-vel-itaque-nihil-id-saepe-
nostrum'
ipv4 // '109.133.32.252'
localIpv4 // '10.242.58.8'
ipv6 // '8e65:933d:22ee:a232:f1c1:2741:1f10:117c'
macAddress // '43:85:B7:08:10:CA'
```

## Faker\Provider\UserAgent

```
userAgent          // 'Mozilla/5.0 (Windows CE) AppleWebKit/5350 (KHTML, like
Gecko) Chrome/13.0.888.0 Safari/5350'
chrome             // 'Mozilla/5.0 (Macintosh; PPC Mac OS X 10_6_5)
AppleWebKit/5312 (KHTML, like Gecko) Chrome/14.0.894.0 Safari/5312'
firefox            // 'Mozilla/5.0 (X11; Linux i686; rv:7.0) Gecko/20101231
Firefox/3.6'
safari             // 'Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_7_1 rv:3.0;
en-US) AppleWebKit/534.11.3 (KHTML, like Gecko) Version/4.0 Safari/534.11.3'
opera              // 'Opera/8.25 (Windows NT 5.1; en-US) Presto/2.9.188
Version/10.00'
internetExplorer   // 'Mozilla/5.0 (compatible; MSIE 7.0; Windows 98; Win 9x
4.90; Trident/3.0)'
```

## Faker\Provider\Payment

```
creditCardType      // 'MasterCard'
creditCardNumber     // '4485480221084675'
creditCardExpirationDate // 04/13
creditCardExpirationDateString // '04/13'
creditCardDetails    // array('MasterCard', '4485480221084675', 'Aleksander
Nowak', '04/13')
// Generates a random IBAN. Set $countryCode to null for a random country
iban($countryCode)   // 'IT31A8497112740YZ575DJ28BP4'
swiftBicNumber        // 'RZTIAT22263'
```

## Faker\Provider\Color

```
hexcolor           // '#fa3cc2'
rgbcolor            // '0,255,122'
rgbColorAsArray     // array(0,255,122)
rgbCssColor         // 'rgb(0,255,122)'
safeColorName       // 'fuchsia'
colorName           // 'Gainsbor'
```

## Faker\Provider\File

```
fileExtension       // 'avi'
mimeType            // 'video/x-msvideo'
// Copy a random file from the source to the target directory and returns the
fullpath or filename
file($sourceDir = '/tmp', $targetDir = '/tmp') //
'/path/to/targetDir/13b73edae8443990be1aa8f1a483bc27.jpg'
file($sourceDir, $targetDir, false) // '13b73edae8443990be1aa8f1a483bc27.jpg'
```

## Faker\Provider\Image

```
// Image generation provided by LoremPixel (http://loempixel.com/)
imageUrl($width = 640, $height = 480) // 'http://loempixel.com/640/480/'
imageUrl($width, $height, 'cats')      // 'http://loempixel.com/800/600/cats/'
imageUrl($width, $height, 'cats', true, 'Faker') //
'http://loempixel.com/800/400/cats/Faker'
imageUrl($width, $height, 'cats', true, 'Faker', true) //
'http://loempixel.com/grey/800/400/cats/Faker/' Monochrome image
image($dir = '/tmp', $width = 640, $height = 480) //
'/tmp/13b73edae8443990be1aa8f1a483bc27.jpg'
image($dir, $width, $height, 'cats') //
'tmp/13b73edae8443990be1aa8f1a483bc27.jpg' it's a cat!
image($dir, $width, $height, 'cats', false) //
'13b73edae8443990be1aa8f1a483bc27.jpg' it's a filename without path
image($dir, $width, $height, 'cats', true, false) // it's a no randomize images
(default: `true`)
image($dir, $width, $height, 'cats', true, true, 'Faker') //
'tmp/13b73edae8443990be1aa8f1a483bc27.jpg' it's a cat with 'Faker' text. Default,
`null`.
```

## Faker\Provider\Uuid

```
uuid // '7e57d004-2b97-0e7a-b45f-5387367791cd'
```

## Faker\Provider\Barcode

```
ean13 // '4006381333931'
ean8   // '73513537'
isbn13 // '9790404436093'
isbn10 // '4881416324'
```

## Faker\Provider\Miscellaneous

```
boolean // false
boolean($chanceOfGettingTrue = 50) // true
md5      // 'de99a620c50f2990e87144735cd357e7'
sha1     // 'f08e7f04ca1a413807ebc47551a40a20a0b4de5c'
sha256   //
'0061e4c60dac5c1d82db0135a42e00c89ae3a333e7c26485321f24348c7e98a5'
locale   // en_UK
countryCode // UK
languageCode // en
currencyCode // EUR
emoji    // 😊
```

## Faker\Provider\Biased

```
// get a random number between 10 and 20,
// with more chances to be close to 20
biasedNumberBetween($min = 10, $max = 20, $function = 'sqrt')
```

## Faker\Provider\HtmlLorem

```
//Generate HTML document which is no more than 2 levels deep, and no more than 3
elements wide at any level.
randomHtml(2,3) // <html><head><title>Aut illo dolore et accusantium
eum.</title></head><body><form action="example.co
```

## Modifiers

---

Faker provides three special providers, `unique()`, `optional()`, and `valid()`, to be called before any provider.

```
// unique() forces providers to return unique values

$values = array();
for ($i=0; $i < 10; $i++) {
    // get a random digit, but always a new one, to avoid duplicates
    $values []= $faker->unique()->randomDigit;
}
print_r($values); // [4, 1, 8, 5, 0, 2, 6, 9, 7, 3]

// providers with a limited range will throw an exception when no new unique value
// can be generated

$values = array();
try {
    for ($i=0; $i < 10; $i++) {
        $values []= $faker->unique()->randomDigitNotNull;
    }
} catch (\OverflowException $e) {
    echo "There are only 9 unique digits not null, Faker can't generate 10 of
them!";
}

// you can reset the unique modifier for all providers by passing true as first
// argument

$faker->unique($reset = true)->randomDigitNotNull; // will not throw
// OverflowException since unique() was reset

// tip: unique() keeps one array of values per provider

// optional() sometimes bypasses the provider to return a default value instead
// (which defaults to NULL)

$values = array();
for ($i=0; $i < 10; $i++) {
    // get a random digit, but also null sometimes
    $values []= $faker->optional()->randomDigit;
}
print_r($values); // [1, 4, null, 9, 5, null, null, 4, 6, null]

// optional() accepts a weight argument to specify the probability of receiving
// the default value.

// 0 will always return the default value; 1 will always return the provider.
// Default weight is 0.5 (50% chance).

$faker->optional($weight = 0.1)->randomDigit; // 90% chance of NULL
$faker->optional($weight = 0.9)->randomDigit; // 10% chance of NULL

// optional() accepts a default argument to specify the default value to return.
```



```

// Defaults to NULL.

$faker->optional($weight = 0.5, $default = false)->randomDigit; // 50% chance of
FALSE
$faker->optional($weight = 0.9, $default = 'abc')->word; // 10% chance of 'abc'

// valid() only accepts valid values according to the passed validator functions
$values = array();
$evenValidator = function($digit) {
    return $digit % 2 === 0;
};
for ($i=0; $i < 10; $i++) {
    $values []= $faker->valid($evenValidator)->randomDigit;
}
print_r($values); // [0, 4, 8, 4, 2, 6, 0, 8, 8, 6]

// just like unique(), valid() throws an overflow exception when it can't generate
a valid value

$values = array();
try {
    $faker->valid($evenValidator)->randomElement(1, 3, 5, 7, 9);
} catch (\OverflowException $e) {
    echo "Can't pick an even number in that set!";
}

```

If you would like to use a modifier with a value not generated by Faker, use the `passthrough()` method. `passthrough()` simply returns whatever value it was given.

```

$faker->optional()->passthrough(mt_rand(5, 15));

```

## Aula 10 - Seeders

### Filtrando os comandos seed com grep

```
Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ php artisan | grep seed
db:seed          Seed the database with records
make:seeder      Create a new seeder class

Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ |
```

### Criando factory e seeder para a tabela Customer

database/Factory/CustomerFactory.php

```
<?php
```

```
use Faker\Generator as Faker;
```

```
$factory->define(App\Customer::class, function (Faker $faker) {
    return [
        'name' => $faker->name,
        'city' => $faker->city,
        'state' => $faker->stateAbbr,
        'birthdate' => $faker->datetime(),
        'special_customer' => $faker->boolean
    ];
});
```

```
php artisan make:seeder CustomersSeeder
```

```
Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ php artisan make:seeder CustomersSeeder
Seeder created successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ |
```

### database/seeds/CustomersSeeder.php

```
<?php

use Illuminate\Database\Seeder;
use App\Customer;

class CustomersSeeder extends Seeder
{
    public function run()
    {
        Customer::truncate();
        factory(App\Customer::class, 50)->create();
    }
}
```

### database/seeds/DatabaseSeeder.php

```
<?php

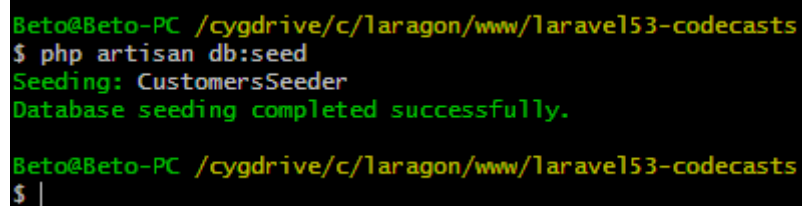
use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    public function run()
    {
        // $this->call(UsersTableSeeder::class);
        $this->call(CustomesrSeeder::class);
    }
}
```

## Populando a tabela Customer

No terminal execute o seguinte comando:

```
php artisan db:seed
```



```
Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ php artisan db:seed
Seeding: CustomersSeeder
Database seeding completed successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ |
```

✔ Showing rows 0 - 24 (50 total, Query took 0.0030 seconds.)

`SELECT * FROM `customers``

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Ref\]](#)

1 ▾

> >>

☐ Show all

Number of rows: 25 ▾

Filter rows:

Sort by key: None ▾

+ Options

|                          |  |  | id | name | city                 | state                  | birthdate | special_customer    | created_at | updated_at          |
|--------------------------|--|--|----|------|----------------------|------------------------|-----------|---------------------|------------|---------------------|
| <input type="checkbox"/> |  |  |    | 1    | Lysanne Wehner       | North Elliottmouth     | WI        | 2014-03-27 21:06:01 | 1          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 2    | Dayna Gaylord        | Delphabury             | MI        | 1994-01-25 14:25:10 | 0          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 3    | Prof. Daphne Lang V  | East Geneviefurt       | OK        | 1998-05-27 13:06:32 | 0          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 4    | Beaulah Torphy       | North Felicitaaborough | IN        | 1998-05-06 19:39:52 | 1          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 5    | Carey Abbott         | Esmeraldaville         | MO        | 2002-08-03 17:03:25 | 1          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 6    | Meghan Willms        | North Masonview        | ND        | 1975-04-20 02:49:57 | 1          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 7    | Genoveva Lebsack Jr. | New Rubieberg          | IL        | 1999-03-01 13:54:44 | 0          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 8    | Prof. Enola Walker   | Rowetown               | IL        | 1989-06-17 03:01:49 | 0          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 9    | Santina Stehr        | South Stella           | WY        | 2006-02-13 03:09:38 | 0          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 10   | Bridget Rau DDS      | Pollichborough         | KS        | 1990-08-13 15:39:00 | 0          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 11   | Ms. Kaya Donnelly    | West Paxtonbury        | ME        | 1999-03-11 12:15:32 | 0          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 12   | Viva Erdman          | Port Jaida             | HI        | 1975-06-16 01:34:22 | 0          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 13   | Toni Halvorson       | New Diamond            | LA        | 2018-04-03 01:08:54 | 0          | 2019-02-28 11:36:48 |
| <input type="checkbox"/> |  |  |    | 14   | Terence Lind PhD     | New Alanis             | MA        | 2006-01-11 22:43:42 | 1          | 2019-02-28 11:36:48 |

## Criando factory e seeder para a tabela de usuários

php artisan make:seeder UsersSeeder

```
Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ php artisan make:seeder UsersSeeder
Seeder created successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ |
```

database/seeds/UsersSeeder.php

<?php

use Illuminate\Database\Seeder;

use App\User;

class UsersSeeder extends Seeder

{

public function run()

{

User::truncate();

\$user = new User();

\$user->name = 'Fabio Vedovelli';

\$user->email = 'vedovelli@gmail.com';

\$user->password = bcrypt(123456);

\$user->email\_verified\_at = now();

\$user->save();

factory(App\User::class, 50)->create();

}

}

## database/seeds/DatabaseSeeder.php

```
<?php
```

```
use Illuminate\Database\Seeder;
```

```
class DatabaseSeeder extends Seeder
{
    public function run()
    {
        $this->call(UsersSeeder::class);
        $this->call(CustomersSeeder::class);
    }
}
```

## Populando apenas a tabela de usuários

```
php artisan db:seed --class=UsersSeeder
```

```
Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ php artisan db:seed --class=UsersSeeder
Database seeding completed successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ |
```

Showing rows 0 - 24 (51 total, Query took 0.0020 seconds.)

SELECT \* FROM `users`

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

1 > >> ☐ Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

|                          |                  | id | name                | email                        | email_verified_at   | password   | remember  |
|--------------------------|------------------|----|---------------------|------------------------------|---------------------|--|-----------|
| <input type="checkbox"/> | Edit Copy Delete | 1  | Fabio Vedovelli     | vedovelli@gmail.com          | 2019-02-28 12:11:16 | \$2y\$10\$70..H7/4PLU6UCgNVv4ErI5Qu542BbbXfZgR8KByHn...  | NULL      |
| <input type="checkbox"/> | Edit Copy Delete | 2  | Prof. Buddy Witting | brandon09@example.org        | 2019-02-28 12:11:16 | \$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | u8ANcNN3  |
| <input type="checkbox"/> | Edit Copy Delete | 3  | Aurelie Krajcik     | ebauch@example.net           | 2019-02-28 12:11:16 | \$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | v3xfSsmKz |
| <input type="checkbox"/> | Edit Copy Delete | 4  | Thurman Breitenberg | kiarra47@example.net         | 2019-02-28 12:11:16 | \$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | 1G3vb3Jr2 |
| <input type="checkbox"/> | Edit Copy Delete | 5  | Nia Yost DDS        | fabian23@example.com         | 2019-02-28 12:11:16 | \$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | Q22ZLL9o  |
| <input type="checkbox"/> | Edit Copy Delete | 6  | Jeanette O'Connell  | reanna.russel@example.com    | 2019-02-28 12:11:16 | \$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | hwZN98zj2 |
| <input type="checkbox"/> | Edit Copy Delete | 7  | Monty Hane          | blick.monte@example.org      | 2019-02-28 12:11:16 | \$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | SJbVfCq8V |
| <input type="checkbox"/> | Edit Copy Delete | 8  | Elva Nienow         | usmith@example.net           | 2019-02-28 12:11:16 | \$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | cdpXGqNP  |
| <input type="checkbox"/> | Edit Copy Delete | 9  | Jordan Kohler       | lourdes.eichmann@example.com | 2019-02-28 12:11:16 | \$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | fNMqnuMH  |
| <input type="checkbox"/> | Edit Copy Delete | 10 | Janis Gerhold       | colton66@example.org         | 2019-02-28 12:11:16 | \$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | u6MHjFFC  |

## Criando o factory e o seeder para a tabela de compras

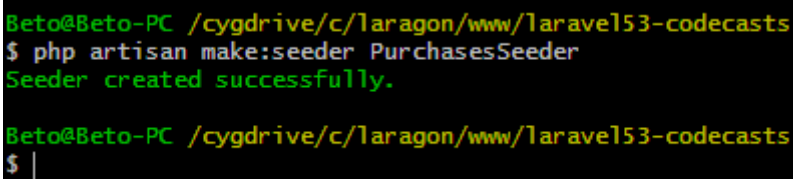
database/Factory/PurchaseFactory.php

```
<?php

use Faker\Generator as Faker;
use App\Customer;

$factory->define(App\Purchase::class, function (Faker $faker) {
    return [
        'customer_id' => $faker->numberBetween($min = 1, $max = 50),
        'amount' => $faker->randomFloat(2, 1, 400),
        'description' => $faker->sentence(),
    ];
});
```

php artisan make:seeder PurchasesSeeder



```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan make:seeder PurchasesSeeder
Seeder created successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ |
```

database/seeds/DatabaseSeeder.php

```
<?php

use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    public function run()
    {
        $this->call(PurchasesSeeder::class);
        $this->call(UsersSeeder::class);
        $this->call(CustomersSeeder::class);
    }
}
```

## database/seeds/PurchasesSeeder.php

<?php

use Illuminate\Database\Seeder;

use App\Purchase;

class PurchasesSeeder extends Seeder

```
{
    public function run()
    {
        Purchase::truncate();
        factory(App\Purchase::class, 120)->create();
    }
}
```

## Populando apenas a tabela de compras

php artisan db:seed --class=PurchasesSeeder

Showing rows 0 - 119 (120 total, Query took 0.0090 seconds.)

`SELECT * FROM `purchases``

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP]

☐ Show all | Number of rows:  Filter rows:  Sort by key:

+ Options

|                          |  |  |  | id | customer_id | amount | description   | created_at          | updated_at          |
|--------------------------|--|--|--|----|-------------|--------|---|---------------------|---------------------|
| <input type="checkbox"/> |  |  |  | 1  | 14          | 81.91  | Ullam praesentium asperiores impedit aut eaque qui... | 2019-02-28 17:11:27 | 2019-02-28 17:11:27 |
| <input type="checkbox"/> |  |  |  | 2  | 14          | 151.62 | Vitae id blanditis dolore dolor.                      | 2019-02-28 17:11:27 | 2019-02-28 17:11:27 |
| <input type="checkbox"/> |  |  |  | 3  | 13          | 230.63 | Doloremque et expedita id ratione eos.                | 2019-02-28 17:11:27 | 2019-02-28 17:11:27 |
| <input type="checkbox"/> |  |  |  | 4  | 30          | 209.79 | Veritatis soluta accusamus explicabo minus tenetur... | 2019-02-28 17:11:28 | 2019-02-28 17:11:28 |
| <input type="checkbox"/> |  |  |  | 5  | 24          | 367.38 | Pariatur quibusdam voluptatem et velit harum.         | 2019-02-28 17:11:28 | 2019-02-28 17:11:28 |
| <input type="checkbox"/> |  |  |  | 6  | 3           | 71.05  | Unde ad autem quis quia qui.                          | 2019-02-28 17:11:28 | 2019-02-28 17:11:28 |
| <input type="checkbox"/> |  |  |  | 7  | 9           | 154.1  | Ut quas aliquid totam quia quia rerum consequatur.    | 2019-02-28 17:11:28 | 2019-02-28 17:11:28 |
| <input type="checkbox"/> |  |  |  | 8  | 14          | 6.26   | Et cum sunt explicabo.                                | 2019-02-28 17:11:28 | 2019-02-28 17:11:28 |
| <input type="checkbox"/> |  |  |  | 9  | 47          | 304.12 | Molestiae eum culpa architecto.                       | 2019-02-28 17:11:28 | 2019-02-28 17:11:28 |
| <input type="checkbox"/> |  |  |  | 10 | 19          | 169.39 | Labore ut sapiente sit.                               | 2019-02-28 17:11:28 | 2019-02-28 17:11:28 |
| <input type="checkbox"/> |  |  |  | 11 | 29          | 116.73 | Est at incidunt dicta iure consequatur ratione fac... | 2019-02-28 17:11:28 | 2019-02-28 17:11:28 |
| <input type="checkbox"/> |  |  |  | 12 | 39          | 235.96 | Omnis ut in eos.                                      | 2019-02-28 17:11:28 | 2019-02-28 17:11:28 |
| <input type="checkbox"/> |  |  |  | 13 | 24          | 225.73 | Eum beatae voluptatum laboriosam ullam occaecati l... | 2019-02-28 17:11:28 | 2019-02-28 17:11:28 |
| <input type="checkbox"/> |  |  |  | 14 | 31          | 260.29 | Laborum itaque qui minus consequatur quia id aut b... | 2019-02-28 17:11:28 | 2019-02-28 17:11:28 |

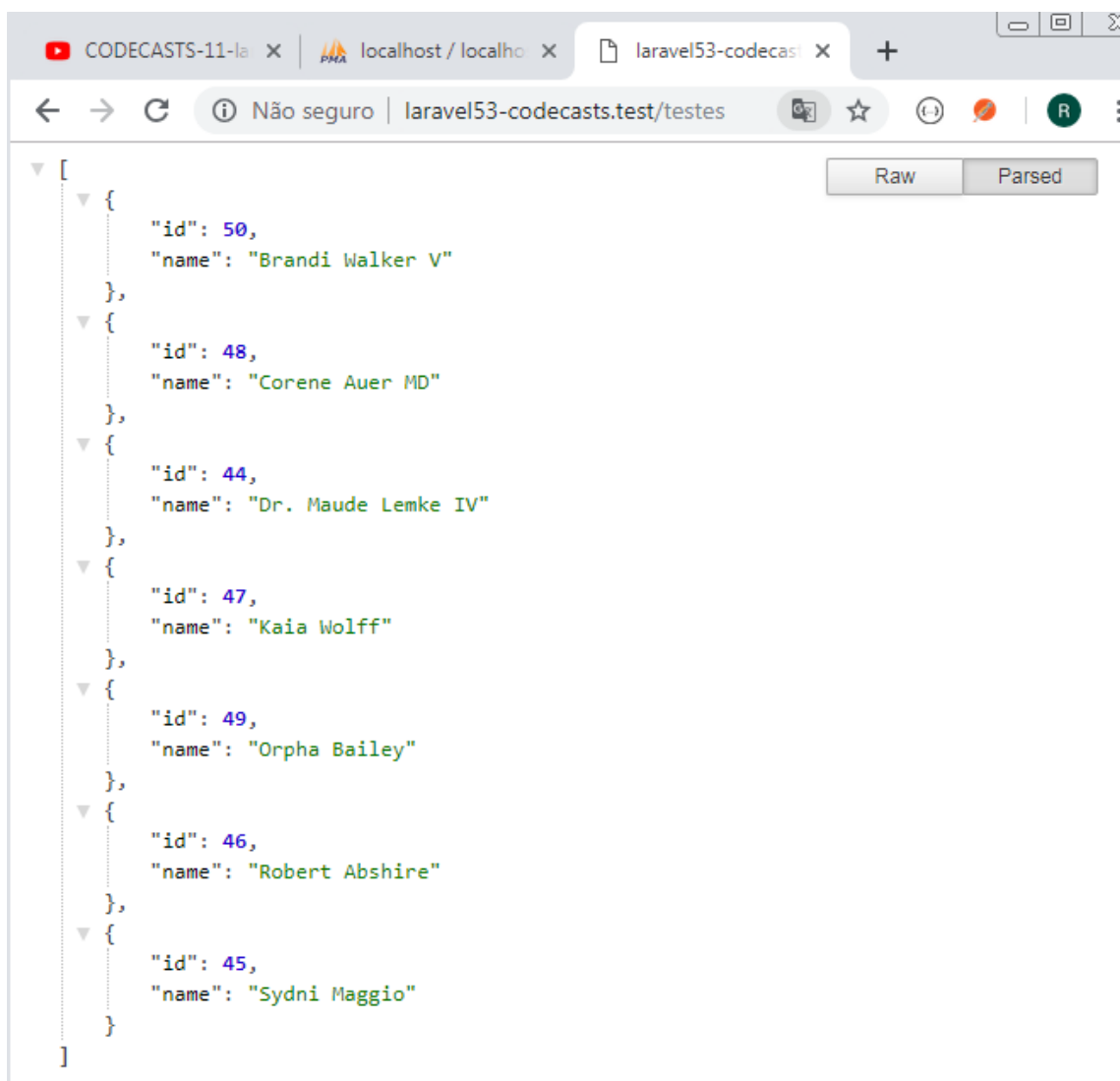


# Aula 11 - Models

## Query Builder

### routes/web.php

```
Route::get('/testes', function () {  
    $query1 = DB::table('customers')  
        ->where('id','>=', 44)  
        ->orderBy('name')  
        ->get(['id','name']);  
  
    return $query1;  
});
```



## Eloquent

### routes/web.php

```
use App\Customer;
```

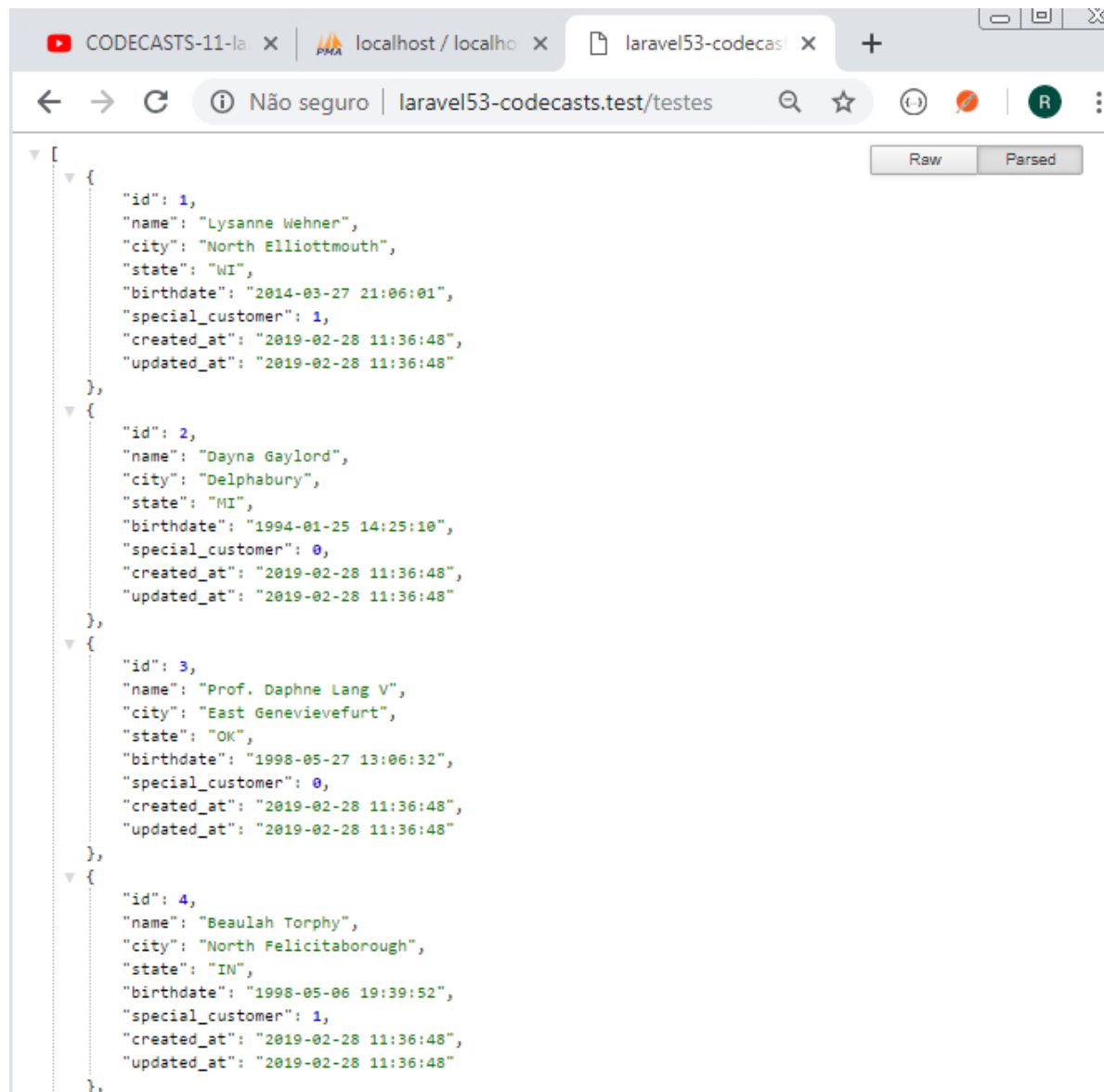
```
Route::get('/testes', function () {  
    $query2 = Customer::find(44);  
    return $query2;  
});
```



## routes/web.php

use App\Customer;

```
Route::get('/testes', function () {  
    $query3 = Customer::all();  
    return $query3;  
});
```



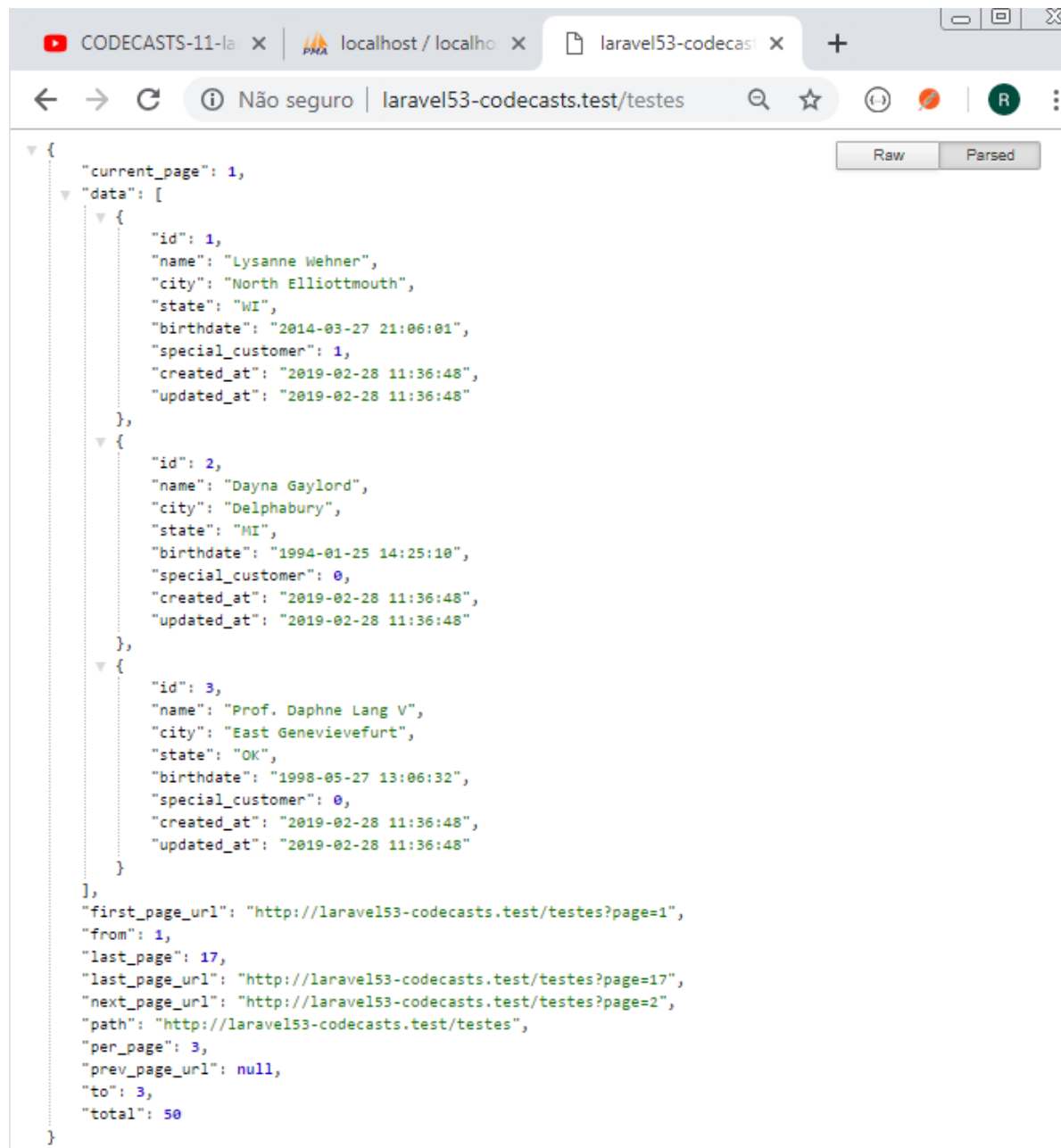
The screenshot shows a web browser window with the address bar displaying "laravel53-codecasts.test/testes". The page content shows a JSON array of four customer objects. The browser's developer tools are open, showing the JSON response in a "Raw" view. The JSON data is as follows:

```
[  
  {  
    "id": 1,  
    "name": "Lysanne Wehner",  
    "city": "North Elliottmouth",  
    "state": "WI",  
    "birthdate": "2014-03-27 21:06:01",  
    "special_customer": 1,  
    "created_at": "2019-02-28 11:36:48",  
    "updated_at": "2019-02-28 11:36:48"  
  },  
  {  
    "id": 2,  
    "name": "Dayna Gaylord",  
    "city": "Delphabury",  
    "state": "MI",  
    "birthdate": "1994-01-25 14:25:10",  
    "special_customer": 0,  
    "created_at": "2019-02-28 11:36:48",  
    "updated_at": "2019-02-28 11:36:48"  
  },  
  {  
    "id": 3,  
    "name": "Prof. Daphne Lang V",  
    "city": "East Genevievefurt",  
    "state": "OK",  
    "birthdate": "1998-05-27 13:06:32",  
    "special_customer": 0,  
    "created_at": "2019-02-28 11:36:48",  
    "updated_at": "2019-02-28 11:36:48"  
  },  
  {  
    "id": 4,  
    "name": "Beaulah Torphy",  
    "city": "North Felicita borough",  
    "state": "IN",  
    "birthdate": "1998-05-06 19:39:52",  
    "special_customer": 1,  
    "created_at": "2019-02-28 11:36:48",  
    "updated_at": "2019-02-28 11:36:48"  
  }  
]
```

## routes/web.php

use App\Customer;

```
Route::get('/testes', function () {  
    $query4 = Customer::paginate(3);  
    return $query4;  
});
```



The screenshot shows a web browser window with the address bar displaying `laravel53-codecasts.test/testes`. The page content is a JSON response, which is displayed in a 'Parsed' view. The JSON data represents a paginated list of customers.

```
{  
  "current_page": 1,  
  "data": [  
    {  
      "id": 1,  
      "name": "Lysanne Wehner",  
      "city": "North Elliottmouth",  
      "state": "WI",  
      "birthdate": "2014-03-27 21:06:01",  
      "special_customer": 1,  
      "created_at": "2019-02-28 11:36:48",  
      "updated_at": "2019-02-28 11:36:48"  
    },  
    {  
      "id": 2,  
      "name": "Dayna Gaylord",  
      "city": "Delphabury",  
      "state": "MI",  
      "birthdate": "1994-01-25 14:25:10",  
      "special_customer": 0,  
      "created_at": "2019-02-28 11:36:48",  
      "updated_at": "2019-02-28 11:36:48"  
    },  
    {  
      "id": 3,  
      "name": "Prof. Daphne Lang V",  
      "city": "East Genevievefurt",  
      "state": "OK",  
      "birthdate": "1998-05-27 13:06:32",  
      "special_customer": 0,  
      "created_at": "2019-02-28 11:36:48",  
      "updated_at": "2019-02-28 11:36:48"  
    }  
  ],  
  "first_page_url": "http://laravel53-codecasts.test/testes?page=1",  
  "from": 1,  
  "last_page": 17,  
  "last_page_url": "http://laravel53-codecasts.test/testes?page=17",  
  "next_page_url": "http://laravel53-codecasts.test/testes?page=2",  
  "path": "http://laravel53-codecasts.test/testes",  
  "per_page": 3,  
  "prev_page_url": null,  
  "to": 3,  
  "total": 50  
}
```

## Relacionamento de tabelas

### app/Customer.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Customer extends Model
{
    public function purchases(){
        // Um cliente possui muitas compras
        return $this->hasMany(Purchase::class);
    }
}
```

### routes/web.php

```
Route::get('/testes', function () {
    // Relacionamento
    $query5 = Customer::with('purchases')->get();

    return $query5;
});
```

## Compras feitas por cliente

```
CODECASTS-11-la x localhost / localho x laravel53-codecast x +
NÃO seguro | laravel53-codecasts.test/testes
Raw Parsed
[
  {
    "id": 1,
    "name": "Lysanne Wehner",
    "city": "North Elliottmouth",
    "state": "WI",
    "birthdate": "2014-03-27 21:06:01",
    "special_customer": 1,
    "created_at": "2019-02-28 11:36:48",
    "updated_at": "2019-02-28 11:36:48",
    "purchases": []
  },
  {
    "id": 2,
    "name": "Dayna Gaylord",
    "city": "Delphabury",
    "state": "MI",
    "birthdate": "1994-01-25 14:25:10",
    "special_customer": 0,
    "created_at": "2019-02-28 11:36:48",
    "updated_at": "2019-02-28 11:36:48",
    "purchases": [
      {
        "id": 40,
        "customer_id": 2,
        "amount": "310.45",
        "description": "Consequatur id repellendus ut assumenda.",
        "created_at": "2019-02-28 17:11:29",
        "updated_at": "2019-02-28 17:11:29"
      }
    ]
  }
],
```

```
{
  "id": 3,
  "name": "Prof. Daphne Lang V",
  "city": "East Genevievefurt",
  "state": "OK",
  "birthdate": "1998-05-27 13:06:32",
  "special_customer": 0,
  "created_at": "2019-02-28 11:36:48",
  "updated_at": "2019-02-28 11:36:48",
  "purchases": [
    {
      "id": 6,
      "customer_id": 3,
      "amount": "71.05",
      "description": "Unde ad autem quis quia qui.",
      "created_at": "2019-02-28 17:11:28",
      "updated_at": "2019-02-28 17:11:28"
    },
    {
      "id": 86,
      "customer_id": 3,
      "amount": "156.81",
      "description": "Quod fugiat perferendis aliquam aut non aut accusamus soluta.",
      "created_at": "2019-02-28 17:11:31",
      "updated_at": "2019-02-28 17:11:31"
    }
  ]
},
{
  "id": 4,
  "name": "Beaulah Torphy",
  "city": "North Felicitaborough",
  "state": "IN",
  "birthdate": "1998-05-06 19:39:52",
  "special_customer": 1,
  "created_at": "2019-02-28 11:36:48",
  "updated_at": "2019-02-28 11:36:48",
  "purchases": []
},
```

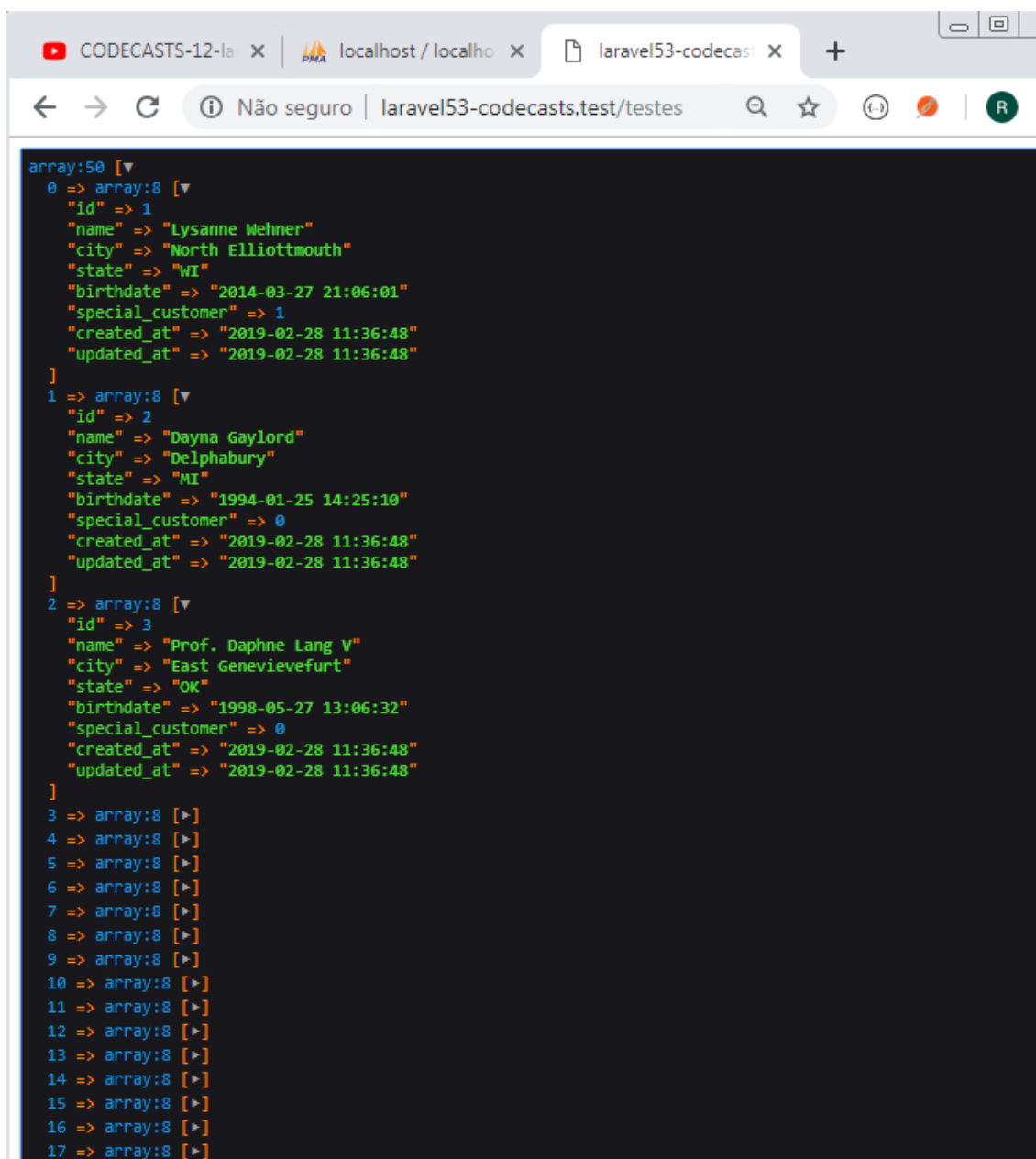
## Aula 12 - Models - Listar registros

routes/web.php

use App\Customer;

use App\Purchase;

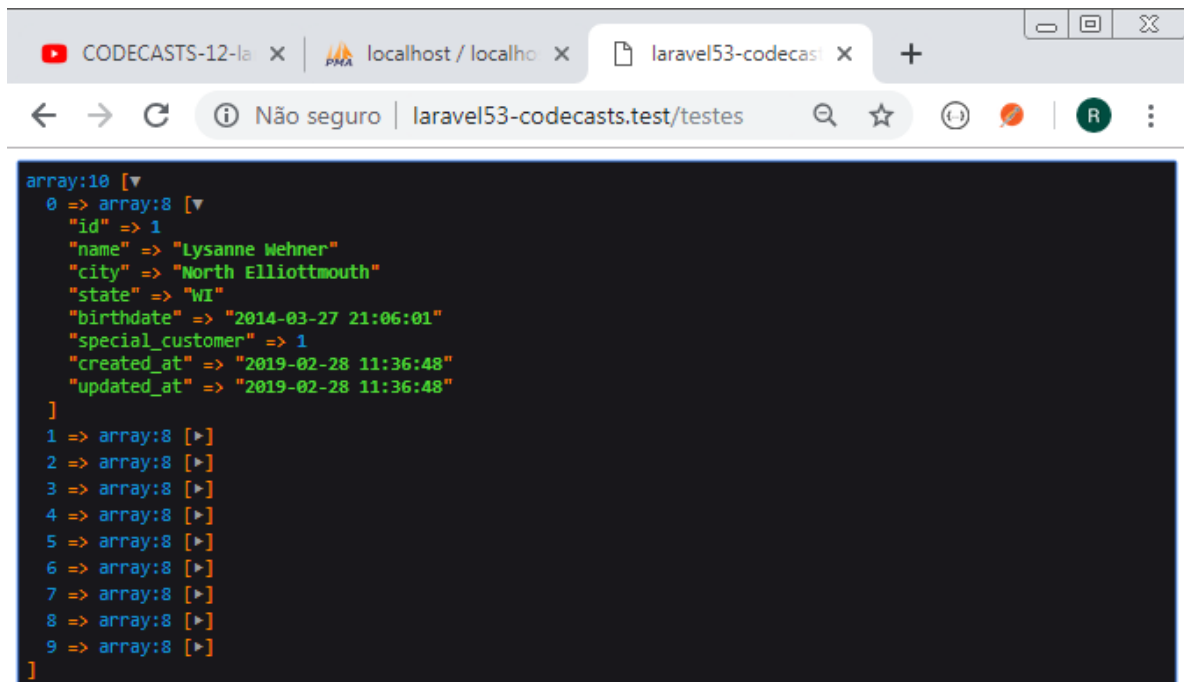
```
Route::get('/testes', function () {  
    $customer1 = Customer::all();  
    dd($customer1->toArray());  
});
```



```
array:50 [▼  
  0 => array:8 [▼  
    "id" => 1  
    "name" => "Lysanne Wehner"  
    "city" => "North Elliottmouth"  
    "state" => "WI"  
    "birthdate" => "2014-03-27 21:06:01"  
    "special_customer" => 1  
    "created_at" => "2019-02-28 11:36:48"  
    "updated_at" => "2019-02-28 11:36:48"  
  ]  
  1 => array:8 [▼  
    "id" => 2  
    "name" => "Dayna Gaylord"  
    "city" => "Delphabury"  
    "state" => "MI"  
    "birthdate" => "1994-01-25 14:25:10"  
    "special_customer" => 0  
    "created_at" => "2019-02-28 11:36:48"  
    "updated_at" => "2019-02-28 11:36:48"  
  ]  
  2 => array:8 [▼  
    "id" => 3  
    "name" => "Prof. Daphne Lang V"  
    "city" => "East Geneviefurt"  
    "state" => "OK"  
    "birthdate" => "1998-05-27 13:06:32"  
    "special_customer" => 0  
    "created_at" => "2019-02-28 11:36:48"  
    "updated_at" => "2019-02-28 11:36:48"  
  ]  
  3 => array:8 [▶]  
  4 => array:8 [▶]  
  5 => array:8 [▶]  
  6 => array:8 [▶]  
  7 => array:8 [▶]  
  8 => array:8 [▶]  
  9 => array:8 [▶]  
  10 => array:8 [▶]  
  11 => array:8 [▶]  
  12 => array:8 [▶]  
  13 => array:8 [▶]  
  14 => array:8 [▶]  
  15 => array:8 [▶]  
  16 => array:8 [▶]  
  17 => array:8 [▶]
```

## routes/web.php

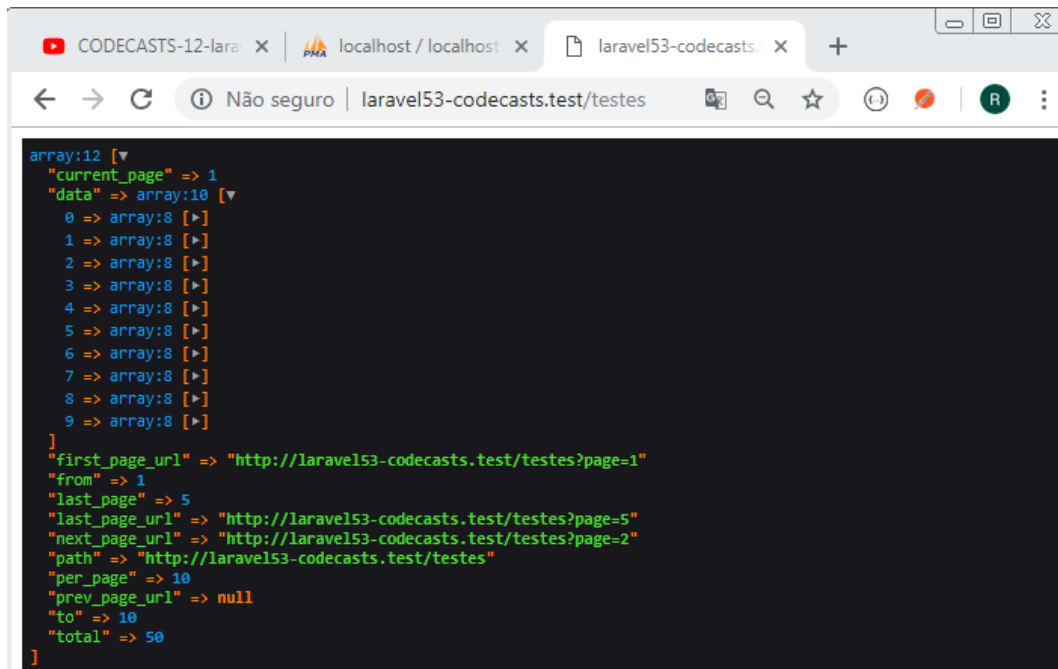
```
Route::get('/testes', function () {  
    $customer2 = Customer::with('purchases')->get();  
    dd($customer2->toArray());  
});
```





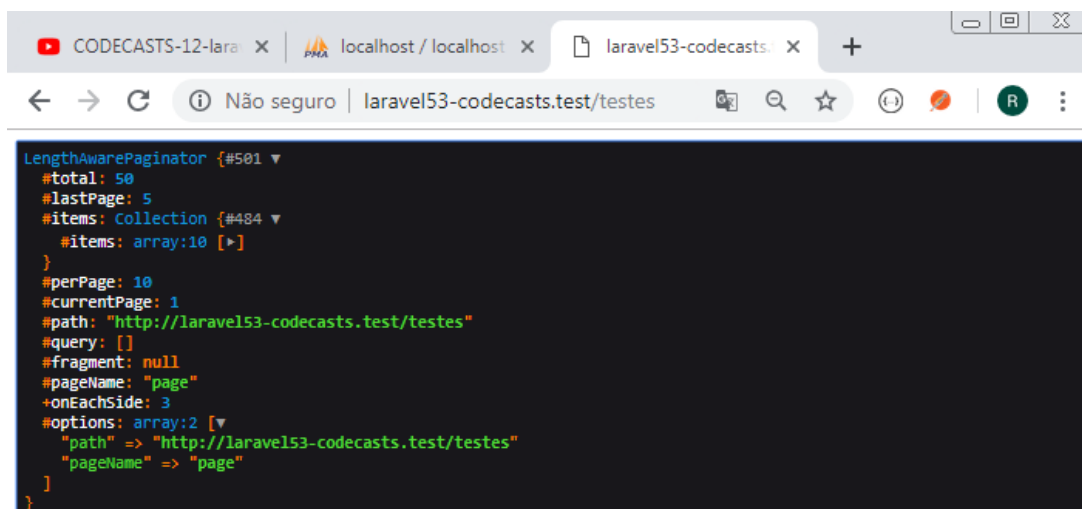
## routes/web.php

```
Route::get('/testes', function () {  
    $customer3 = Customer::take(10)->get();  
    dd($customer3->toArray());  
});
```



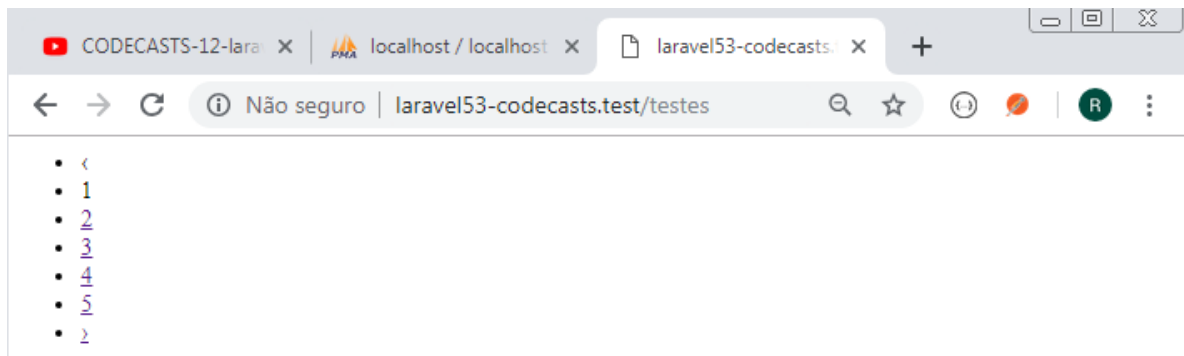
## routes/web.php

```
Route::get('/testes', function () {  
    $customer4 = Customer::paginate(10);  
    dd($customer4);  
});
```



## routes/web.php

```
Route::get('/testes', function () {  
    $customer5 = Customer::render();  
    echo $customer5;  
});
```



## Aula 13 - Models - Criar registros

Efetuar um truncate na tabela customer e posteriormente entrar com o seguinte código:

**routes/web.php**

```
use App\Customer;
```

```
// CRIAR NOVO CUSTOMER
```

```
Route::get('/testes2', function () {  
    $customer = new Customer();  
    $customer->name = 'Fabio Vedovelli';  
    $customer->city = 'Santo Andre';  
    $customer->state = 'SP';  
    $customer->birthdate = '1974-08-07 00:00:00';  
    $customer->special_customer = true;  
  
    $customer->save();  
    dd($customer->toArray());  
});
```

Enquanto o método save não for chamado, o objeto ficará criado apenas em memória.

array:8 [▼  
 "name" => "Fabio Vedovelli"  
 "city" => "Santo Andre"  
 "state" => "SP"  
 "birthdate" => "1974-08-07 00:00:00"  
 "special\_customer" => true  
 "updated\_at" => "2019-03-01 03:00:37"  
 "created\_at" => "2019-03-01 03:00:37"  
 "id" => 1  
]

Showing rows 0 - 0 (1 total, Query took 0.0020 seconds.)

SELECT \* FROM `customers`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 250 | Filter rows: Search this table

+ Options

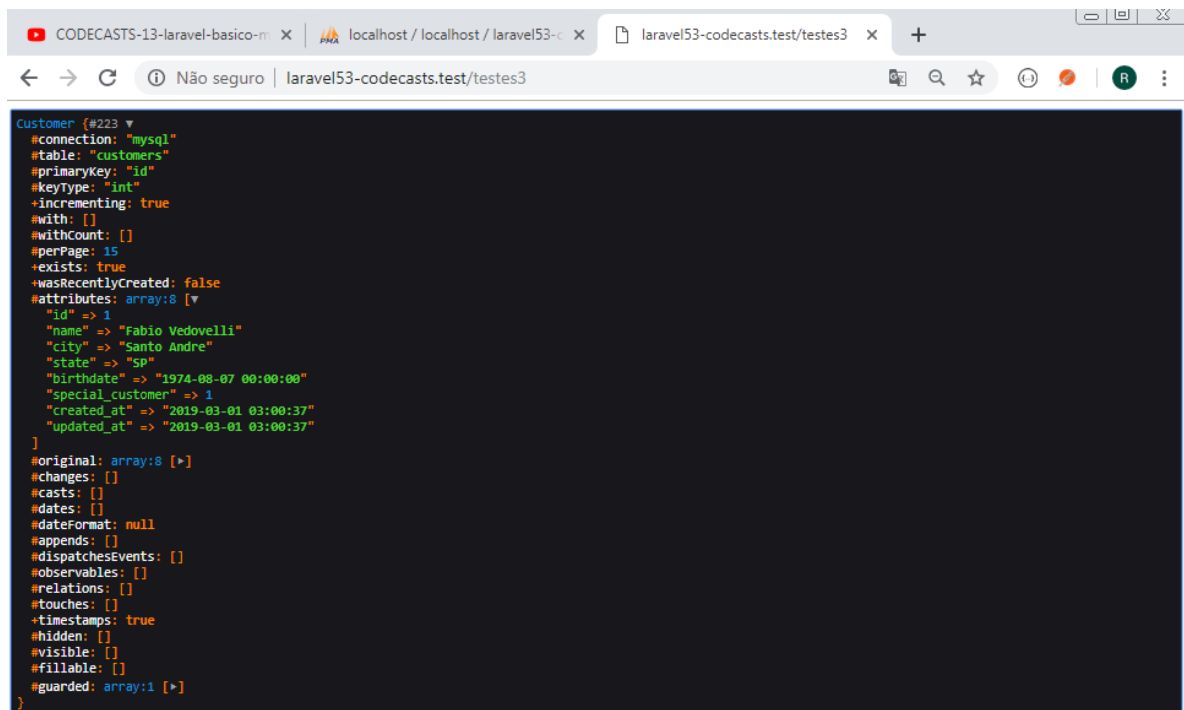
|   | id | name            | city        | state | birthdate           | special_customer | created_at          | updated_at          |
|---|----|-----------------|-------------|-------|---------------------|------------------|---------------------|---------------------|
| <input type="checkbox"/> Edit Copy Delete | 1  | Fabio Vedovelli | Santo Andre | SP    | 1974-08-07 00:00:00 | 1                | 2019-03-01 03:00:37 | 2019-03-01 03:00:37 |

Check all With selected: Edit Copy Delete Export

## routes/web.php

```
// TRAZER CUSTOMER ID 1
```

```
Route::get('/testes3', function () {  
    $customer = Customer::find(1);  
    dd($customer);  
});
```



### app/Customer.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

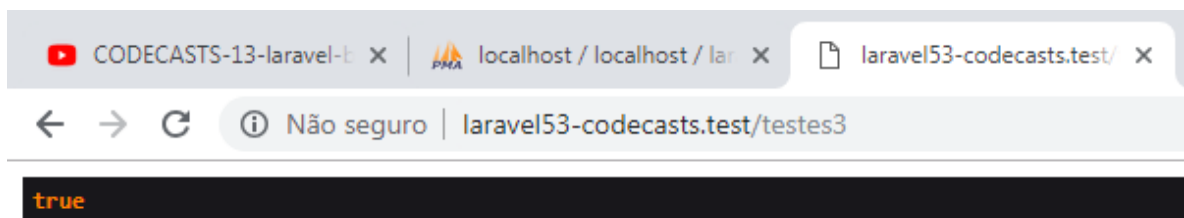
class Customer extends Model
{
    protected $casts = [
        'special_customer' => 'boolean',
    ];

    public function purchases(){
        // Um cliente possui muitas compras
        return $this->hasMany(Purchase::class);
    }
}
```

### routes/web.php

```
// TRAZER CUSTOMER ID 1

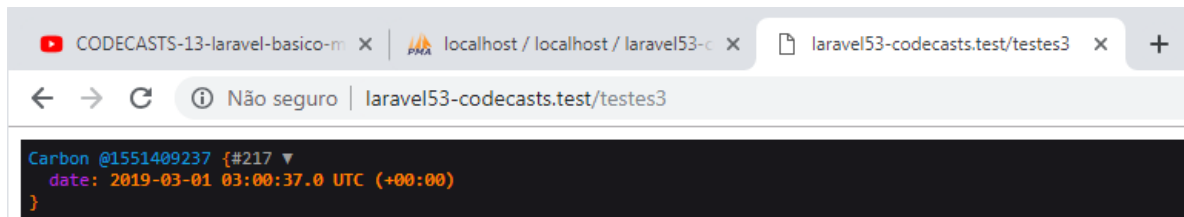
Route::get('/testes3', function () {
    $customer = Customer::find(1);
    dd($customer->special_customer);
});
```



## routes/web.php

```
// TRAZER CUSTOMER ID 1
```

```
Route::get('/testes3', function () {  
    $customer = Customer::find(1);  
    // dd($customer->special_customer);  
    dd($customer->created_at);  
});
```



**Carbon** é uma classe para trabalhar com dados do tipo data. Essa biblioteca faz comparação entre datas.

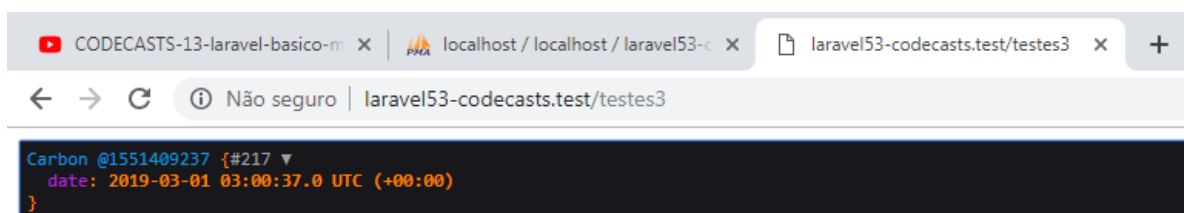
## app/Customer.php

```
<?php
```

```
namespace App;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Customer extends Model  
{  
    protected $casts = [  
        'special_customer' => 'boolean',  
        'birthdate' => 'date'  
    ];  
    public function purchases(){  
        // Um cliente possui muitas compras  
        return $this->hasMany(Purchase::class);  
    }  
}
```



## Aula 14 - Models - Atualizar registros

### routes/web.php

```
use App\Customer;

Route::get('/testes4', function () {
    $data = [
        'name' => 'Pedro',
        'birthdate' => '1989-01-01 12:00:00'
    ];
    $customer = Customer::find(1);
    $customer->update($data);
    return 'Alterado com sucesso!';
});
```

### app/Customer.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Customer extends Model
{
    protected $fillable = [
        'name', 'city', 'state', 'birthdate', 'special_customer'
    ];

    protected $casts = [
        'special_customer' => 'boolean',
        'birthdate' => 'date'
    ];

    public function purchases(){
        // Um cliente possui muitas compras
        return $this->hasMany(Purchase::class);
    }
}
```

Alterado com sucesso!

✓ Showing rows 0 - 0 (1 total, Query took 0.0000 seconds.)

`SELECT * FROM `customers``

☐ Profiling [Edit inline] [Edit] [Explain SQL]

☐ Show all | Number of rows: 250 ▾ Filter rows:

+ Options

|                          |      | id   | name   | city | state | birthdate   | special_customer | created_at          | updated_at                                |
|--------------------------|------|------|--------|------|-------|-------------|------------------|---------------------|---|
| <input type="checkbox"/> | Edit | Copy | Delete | 1    | Pedro | Santo Andre | SP               | 1989-01-01 12:00:00 | 1 2019-03-01 03:00:37 2019-03-01 11:48:13 |

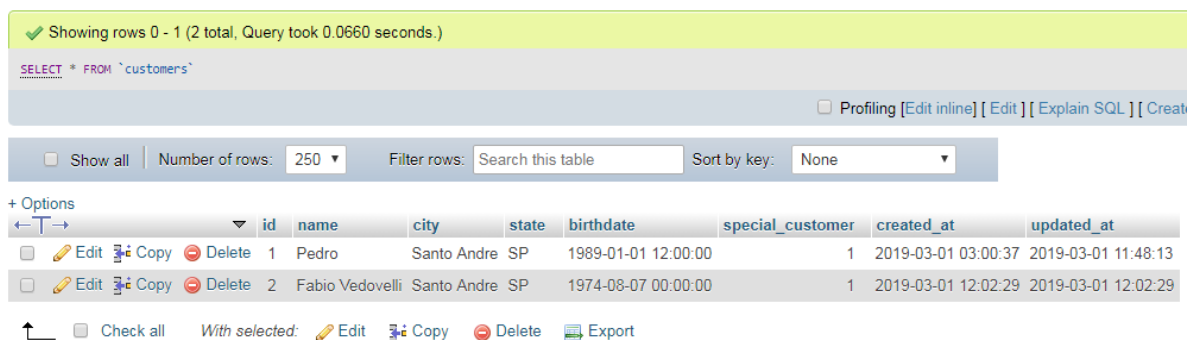
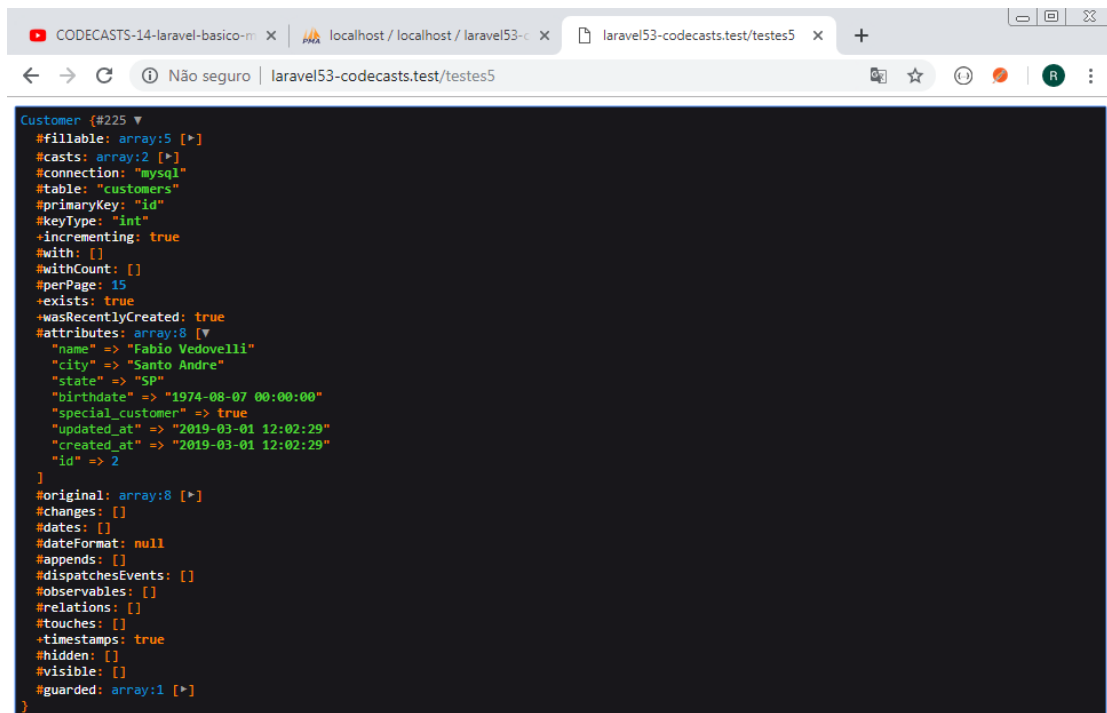
↑ ☐ Check all With selected: Edit Copy Delete Export



## routes/web.php

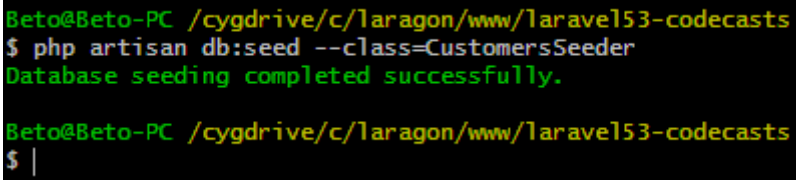
// INSERINDO REGISTRO

```
Route::get('/testes5', function () {
    $data = [
        'name' => 'Fabio Vedovelli',
        'city' => 'Santo Andre',
        'state' => 'SP',
        'birthdate' => '1974-08-07 00:00:00',
        'special_customer' => true
    ];
    $customer = Customer::create($data);
    dd($customer);
});
```



Popular novamente o banco de dados:

```
php artisan db:seed --class=CustomersSeeder
```

A terminal window with a black background and green text. The prompt is 'Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts'. The command entered is '\$ php artisan db:seed --class=CustomersSeeder'. The output is 'Database seeding completed successfully.' followed by a blank line and the prompt '\$ |'.

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan db:seed --class=CustomersSeeder
Database seeding completed successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ |
```

## Aula 16 - Rotas

```
// Listar dados de todos os clientes  
// admin/clientes  
// método GET
```

```
// Listar dados de um cliente específico  
// admin/clientes/1  
// método GET
```

```
// Criar (inserir) novo cliente  
// admin/clientes  
// método POST
```

```
// Alterar dados (atualizar) de um cliente  
// admin/clientes/1  
// método PUT
```

```
// Excluir um cliente específico  
// admin/clientes/1  
// método DELETE
```

## Criando o controller para Customer

php artisan make:controller CustomerController

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts  
$ php artisan make:controller CustomerController  
Controller created successfully.  
  
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts  
$ |
```

## Configurando as rotas

### routes/web.php

```
Route::group(['prefix' => 'admin'], function(){

    Route::get('clientes', ['as' => 'customer.index', 'uses' => 'CustomerController@index']);

    Route::get('cliente/{id}', ['as' => 'customer.get', 'uses' => 'CustomerController@get']);

    Route::post('cliente', ['as' => 'customer.create', 'uses' => 'CustomerController@create']);

    Route::put('cliente/{id}', ['as' => 'customer.update', 'uses' => 'CustomerController@update']);

    Route::delete('cliente/{id}', ['as' => 'customer.delete', 'uses' => 'CustomerController@delete']);

});
```

### app/Http/Controllers/CustomerController.php

```
<?php

namespace App\Http\Controllers;

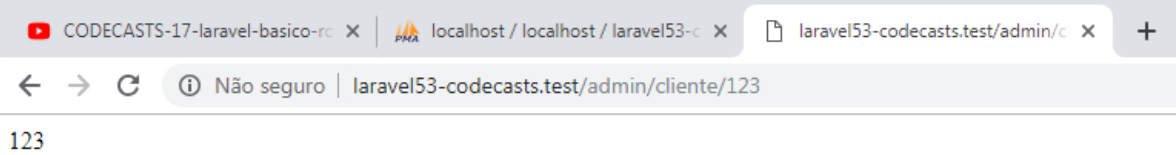
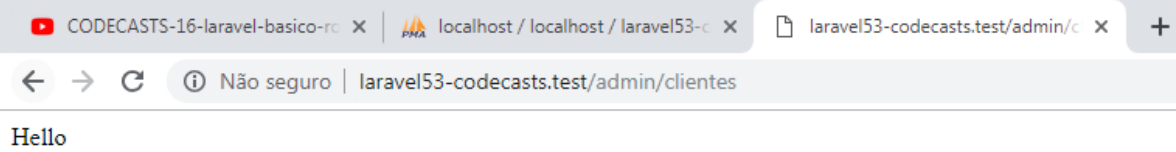
use Illuminate\Http\Request;

class CustomerController extends Controller
{
    public function index(){
        return 'Hello!';
    }
    public function get($id){
        return $id;
    }
    public function create(){

    }
    public function update($id){

    }
    public function delete($id){

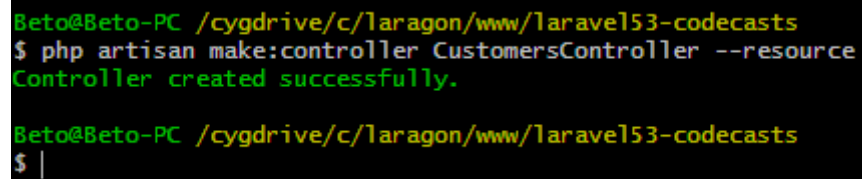
    }
}
```



## Aula 17 - Rotas - Resource

### Criando o controller

php artisan make:controller CustomersController --resource

A terminal window with a black background and yellow/green text. The prompt is 'Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts'. The command entered is '\$ php artisan make:controller CustomersController --resource'. The output is 'Controller created successfully.' followed by a new prompt '\$ |'.

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ php artisan make:controller CustomersController --resource
Controller created successfully.
Beto@Beto-PC /cygdrive/c/laragon/www/laravel53-codecasts
$ |
```

**app/Http/Controllers/CustomersController.php**

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
class CustomersController extends Controller
```

```
{
```

```
    public function index()
```

```
    {
```

```
        //
```

```
    }
```

```
    public function create()
```

```
    {
```

```
        //
```

```
    }
```

```
    public function store(Request $request)
```

```
    {
```

```
        //
```

```
    }
```

```
    public function show($id)
```

```
    {
```

```
        //
```

```
    }
```

```
    public function edit($id)
```

```
    {
```

```
        //
```

```
    }
```

```
    public function update(Request $request, $id)
```

```
    {
```

```
        //
```

```
    }
```

```
    public function destroy($id)
```

```
    {
```

```
        //
```

```
    }
```

```
}
```

php artisan route:list

```
Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ php artisan route:list
```

| Domain | Method    | URI                           | Name             | Action   | Middleware   |
|--------|-----------|-------------------------------|------------------|--|--------------|
|        | GET HEAD  | /                             |                  | Closure  | web          |
|        | GET HEAD  | admin/clientes                | clientes.index   | App\Http\Controllers\CustomersController@index   | web          |
|        | POST      | admin/clientes                | clientes.store   | App\Http\Controllers\CustomersController@store   | web          |
|        | GET HEAD  | admin/clientes/create         | clientes.create  | App\Http\Controllers\CustomersController@create  | web          |
|        | GET HEAD  | admin/clientes/{cliente}      | clientes.show    | App\Http\Controllers\CustomersController@show    | web          |
|        | PUT PATCH | admin/clientes/{cliente}      | clientes.update  | App\Http\Controllers\CustomersController@update  | web          |
|        | DELETE    | admin/clientes/{cliente}      | clientes.destroy | App\Http\Controllers\CustomersController@destroy | web          |
|        | GET HEAD  | admin/clientes/{cliente}/edit | clientes.edit    | App\Http\Controllers\CustomersController@edit    | web          |
|        | GET HEAD  | api/user                      |                  | Closure  | api,auth:api |
|        | GET HEAD  | testes                        |                  | Closure  | web          |
|        | GET HEAD  | testes2                       |                  | Closure  | web          |
|        | GET HEAD  | testes3                       |                  | Closure  | web          |
|        | GET HEAD  | testes4                       |                  | Closure  | web          |
|        | GET HEAD  | testes5                       |                  | Closure  | web          |

```
Beto@Beto-PC /cygdrive/c/laragon/www/larave153-codecasts
$ |
```

routes/web.php

```
Route::group(['prefix' => 'admin'], function(){
    Route::resource('clientes', 'CustomersController');
});
```



## Aula 19 - Views

**app/Http/Controllers/CustomersController.php**

```
<?php

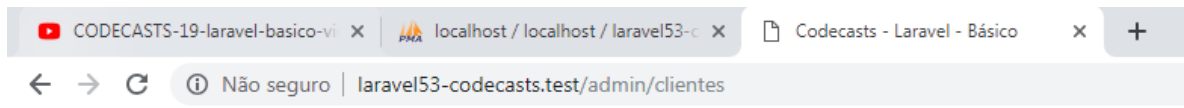
namespace App\Http\Controllers;

use Illuminate\Http\Request;

class CustomersController extends Controller
{
    public function index()
    {
        return view('customers.index');
    }
    public function create()
    {
        //
    }
    public function store(Request $request)
    {
        //
    }
    public function show($id)
    {
        //
    }
    public function edit($id)
    {
        //
    }
    public function update(Request $request, $id)
    {
        //
    }
    public function destroy($id)
    {
        //
    }
}
```

**resources/views/customers/index.php**

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Codecasts - Laravel - Básico</title>
</head>
<body>
  <h1>Bem vindo ao mundo Laravel!</h1>
</body>
</html>
```



**Bem vindo ao mundo Laravel!**

## Aula 20 - Views - Blade

**app/Http/Controllers/CustomersController.php**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Customer;

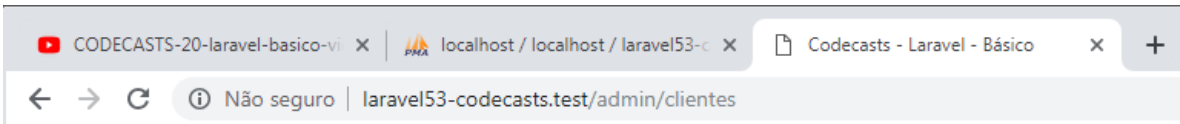
class CustomersController extends Controller
{
    public function index()
    {
        $customers = Customer::all();
        return view('customers.index')->with(compact('customers'));
    }
}
```

**resources/views/customers/index.blade.php**

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Codecasts - Laravel - Básico</title>
</head>
<body>
    <h1>Bem vindo ao mundo Laravel!</h1>

    @foreach($customers as $customer)
        <p>{{ $customer->name }}</p>
    @endforeach

</body>
</html>
```



# Bem vindo ao mundo Laravel!

Otilia Stoltenberg

Marilyne Connelly

Jordon Kuphal

Sunny Gusikowski

Ms. Itzel Doyle MD

Sigurd Bogan III

Dr. Angelita McKenzie

Leta Parker

Austyn Glover

Ms. Clotilde Strosin III

Madyson Rath

Dr. Hardy Blanda

Margret Corwin

Celestino Nader

Ms. Asa Balistreri I

Hillary Aufderhar

Ms. Matilda Orn DDS

## Aula 21 - Bootstrap

**app/Http/Controllers/CustomersController.php**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Customer;

class CustomersController extends Controller
{
    public function index()
    {

        $customers = Customer::all();

        return view('customers.index')->with(compact('customers'));
    }
    public function edit($id)
    {
        return "Edit customer $id";
    }
    public function update(Request $request, $id)
    {
        //
    }
    public function destroy($id)
    {
        return "Delete customer $id";
    }
}
```

## resources/views/customers/index.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Codecasts - Laravel - Básico</title>
    <link rel="stylesheet" type="text/css" href="{{'/css/app.css'}}">
</head>
<body>
    <div class="container">
        <h1 style="margin: 15px 0px" class="text-center">Lista de clientes</h1>
        <table class="table bordered table-striped">
            <thead>
                <th>Id</th>
                <th>Name</th>
                <th>City</th>
                <th>State</th>
                <th>Birthdate</th>
                <th class="text-center">Special Customer</th>
                <th>Ações</th>
            </thead>
            <tbody>
                @foreach($customers as $customer)
                    <tr>
                        <td>{{ $customer->id }}</td>
                        <td>{{ $customer->name }}</td>
                        <td>{{ $customer->city }}</td>
                        <td>{{ $customer->state }}</td>
                        <td>{{ $customer->birthdate }}</td>
                        @if($customer->special_customer == 1)
                            <td class="text-center">Yes</td>
                        @else
                            <td class="text-center">No</td>
                        @endif
                        <td>
                            <a style="padding: 5px;" class="btn btn-xs btn-primary"
href="/admin/clientes/{{ $customer->id }}/edit"> Edit</a>
                            <a style="padding: 5px;" class="btn btn-xs btn-danger"
href="/admin/clientes/{{ $customer->id }}"> Delete</a>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
</body>
</html>
```

| Lista de clientes |                       |                     |       |                     |                  |   |
|-------------------|-----------------------|---------------------|-------|---------------------|------------------|---|
| Id                | Name                  | City                | State | Birthdate           | Special Customer | Ações                                       |
| 1                 | Otilia Stoltenberg    | Enochbury           | ND    | 1991-11-14 00:00:00 | Yes              | <a href="#">Edit</a> <a href="#">Delete</a> |
| 2                 | Marilyne Connely      | North Creolaborough | WA    | 2016-07-14 00:00:00 | No               | <a href="#">Edit</a> <a href="#">Delete</a> |
| 3                 | Jordon Kuphal         | Tyrellchester       | SC    | 2008-07-04 00:00:00 | Yes              | <a href="#">Edit</a> <a href="#">Delete</a> |
| 4                 | Sunny Gusikowski      | East Brenna         | DC    | 2013-02-24 00:00:00 | No               | <a href="#">Edit</a> <a href="#">Delete</a> |
| 5                 | Ms. Itzel Doyle MD    | Lake Alisha         | CA    | 1980-09-02 00:00:00 | No               | <a href="#">Edit</a> <a href="#">Delete</a> |
| 6                 | Sigurd Bogan III      | Schulistfort        | AZ    | 1971-06-17 00:00:00 | No               | <a href="#">Edit</a> <a href="#">Delete</a> |
| 7                 | Dr. Angelita McKenzie | Skilesburgh         | MS    | 2004-09-08 00:00:00 | Yes              | <a href="#">Edit</a> <a href="#">Delete</a> |
| 8                 | Leta Parker           | Port Gerry          | OK    | 2012-05-24 00:00:00 | No               | <a href="#">Edit</a> <a href="#">Delete</a> |
| 9                 | Austyn Glover         | South Juanaburgh    | OH    | 1973-10-12 00:00:00 | Yes              | <a href="#">Edit</a> <a href="#">Delete</a> |