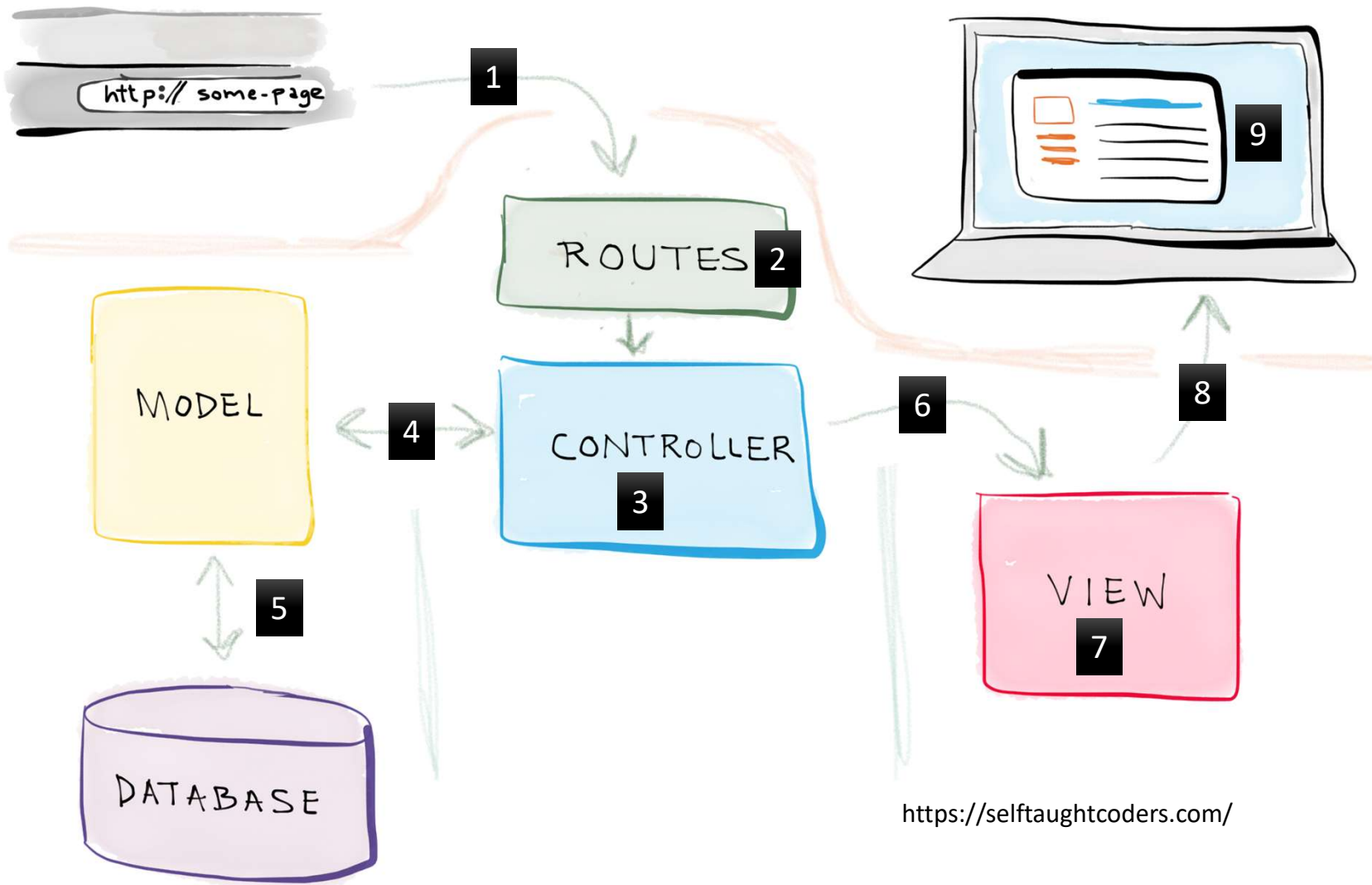# Laravel for Beginners 02

The Anatomy of Laravel

„Hello Laravel" Application

The Laravel MVC-Team players

1. The browser sends a GET or POST request to the web server
2. The web server analyzes the request with the routing table and calls the matching controller method
3. The controller logic is processed.
4. The controller might call a model
5. The model loads or stores data from or into the database
6. The controller loads the view with the data to show in the view
7. The view + data are processed/transformed to pure html/js response
8. The html/js code is sent to the browser
9. The browser presents the resulting page of the response

https://selftaughtcoders.com/
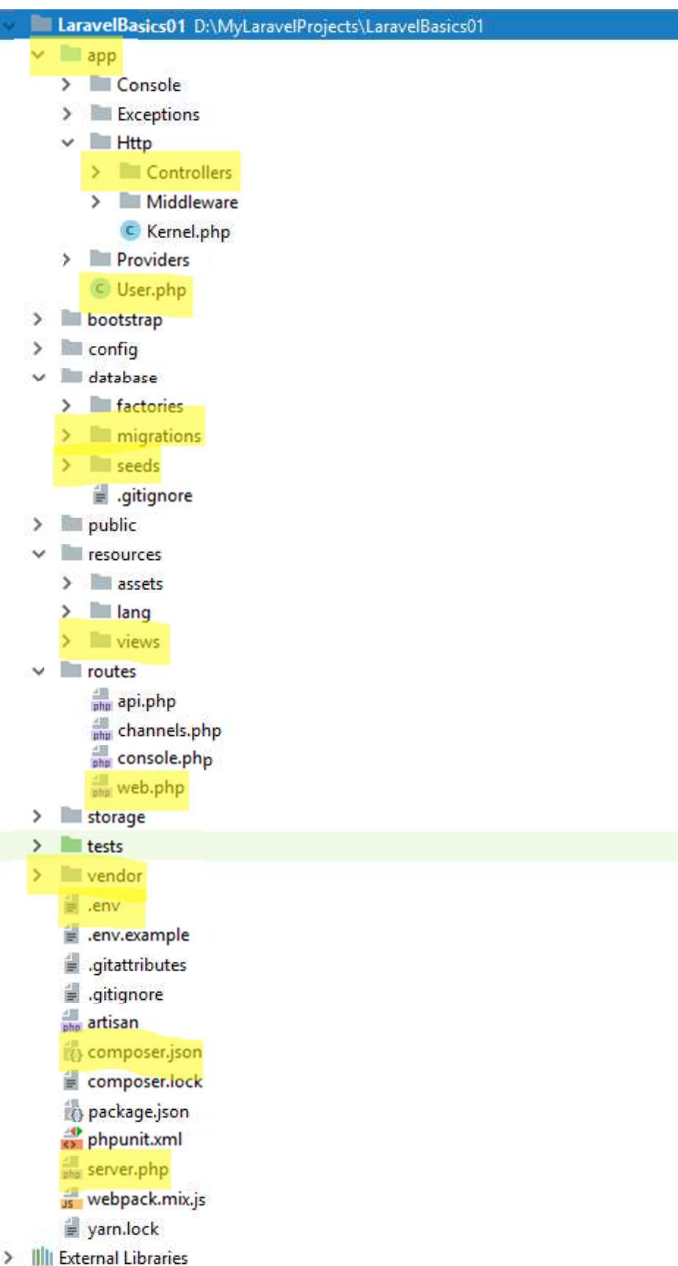
Where is what?

- **app** – the application source code. Put the models here *(e.g. User.php)*
- **app/Http/Controllers** – all controller classes are here
- **config** – all configuration files are here *(they use the params in .env)*
- **database/migrations** – classes to build up your database
- **database/seeds** – classes to fill your database with test data
- **public** – the document root *(= entry point for the web server)*
- **resources/views** – all the views are here
- **routes/web.php** – define the routes for the web server here
- **vendor** – libraries, installed by composer *(do not touch!)*
- **.env** – central configuration file
  (put all your project params in this file – not in files in the config directory!)
- **composer.json** – the configuration file for the composer
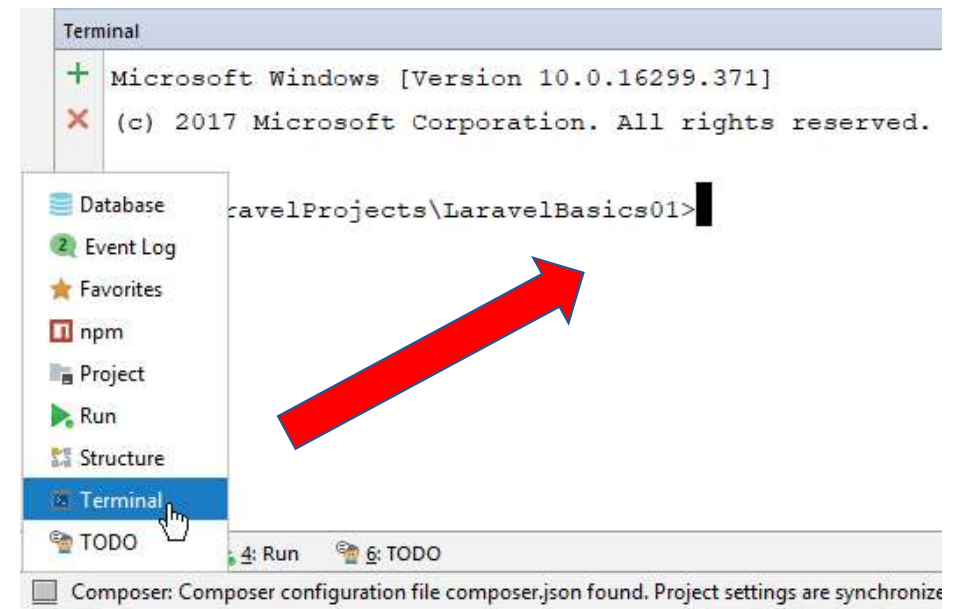- **server.php** – the router script *(do not change!)*

Say „Hello Laravel!" - Your first application

Some few steps:
- Build a Controller
- Define the Route
- Build the View
- Test ... and ☺

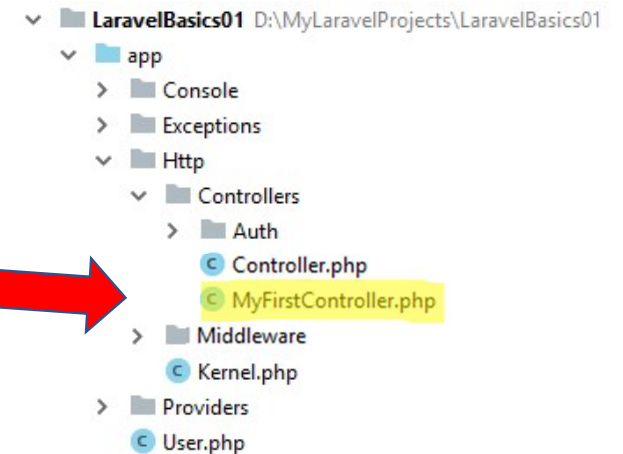... but first of all you need a terminal:
- Hover over the left lower corner and open a terminal

Build the Controller

- Generate a Controller class with the artisan command (artisan is your new best friend ... ☺)

```
php artisan make:controller MyFirstController
```

- Insert this method in the empty Controller class:

```php
class MyFirstController extends Controller
{
    public function sayHello()
    {
        // this is the response to the browser
        // no view yet ...
        return 'Hello Laravel!';
    }
}
```


5

Define the route and test ☺

- Add the following line in the file routes/web.php :

```
// my first route:
Route::get('/hello', 'MyFirstController@sayHello');
```

GET request   URI   Callback Action: Controller@method

- Now you can call the url from the browser:

MVC without M and V

http://localhost:9999/hello

```
Route::get('/hello', 'MyFirstController@sayHello');
```

ROUTES

MODEL

CONTROLLER

'Hello Laravel!'

VIEW

DATABASE

```
public function sayHello()
{
    // this is the response to the browser
    // no view yet ...
    return 'Hello Laravel!';
}
```
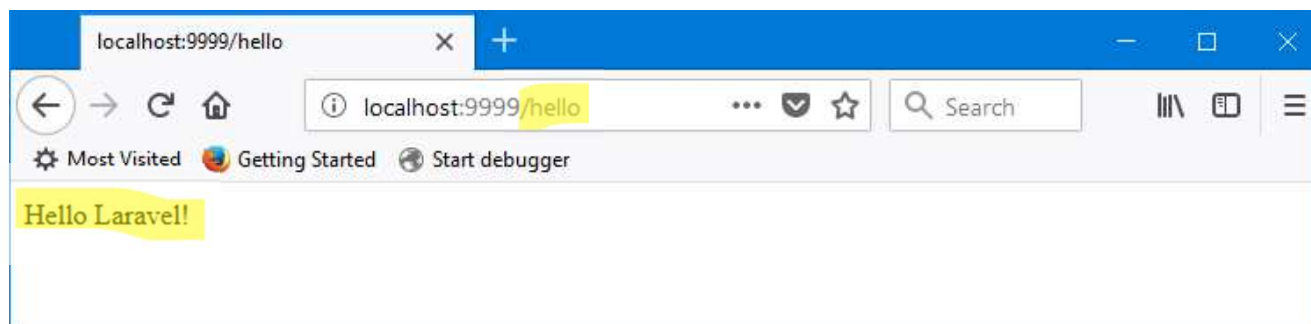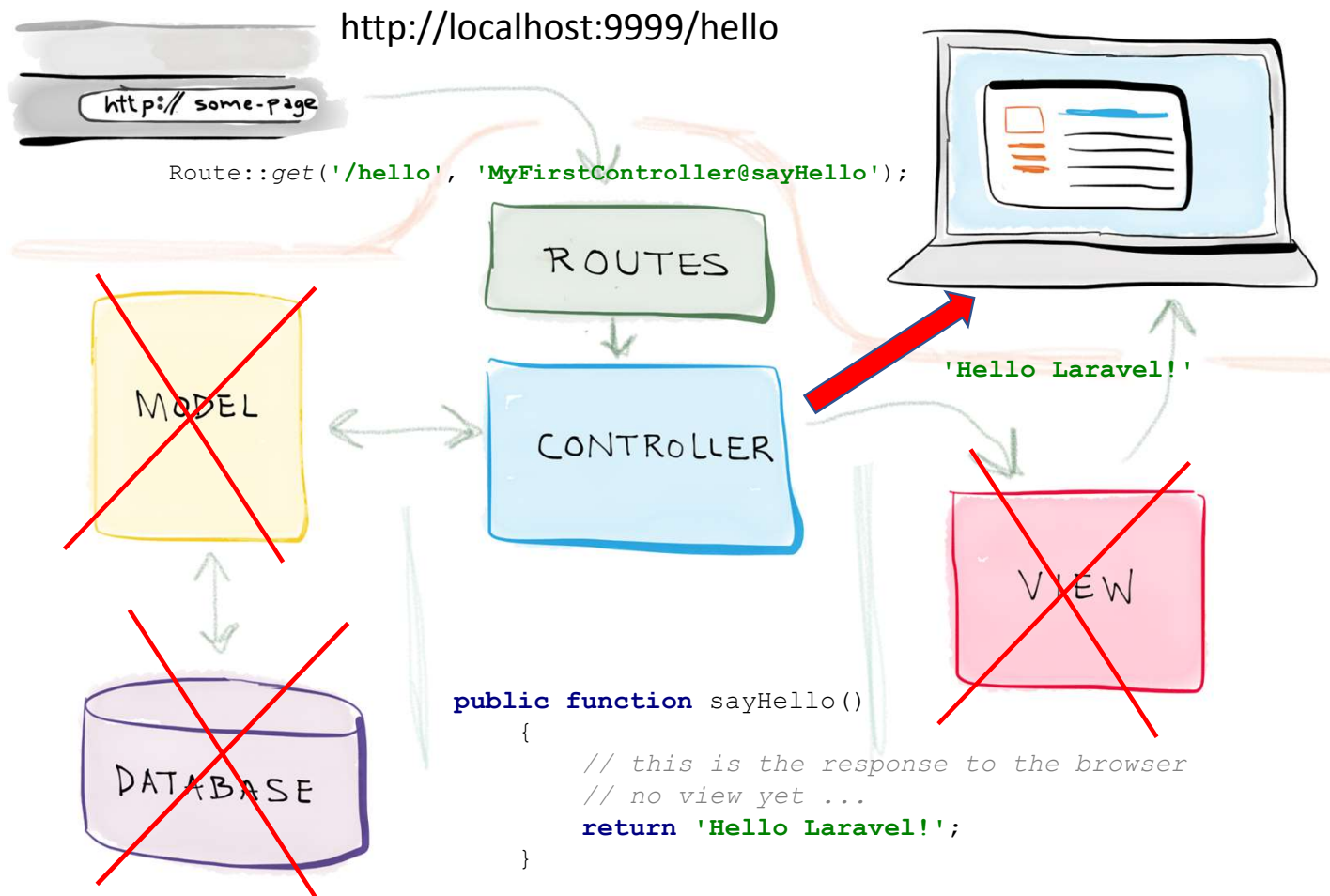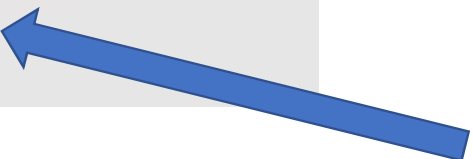
Add a very simple View

- Laravel has a template engine called **blade** to process views (https://laravel.com/docs/5.6/blade). The views can contain special placeholders which will be transformed to php code before the first use.

- Create a php file in the „resources/views" directory with the name *greeting.**blade**.php*

- The content of the file should look like this:

```html
<html>
    <body>
        <h1>Hello, {{ $name }}</h1>
    </body>
</html>
```

<?php echo $name; ?>

Load the view from the controller

- Change the code in the method sayHello:

```php
public function sayHello(Request $request)
{
    $name = $request->get('name');
    if ($name == null)
    {
        $name = 'stranger';
    }
    return view('greeting')->with('name', $name);
}
```

- Send a GET-parameter: