

Laravel Workshop

1. Basic Environment Setup

1. install php and related components . (Using Apt)

1. php7.0
2. php7.0-common
3. php7.0-cli
4. php7.0-mbstring
5. php7.0-xml
6. php7.0-zip

2. composer

1. apt-get install composer

3. Atom IDE

1. wget <https://atom.io/download/deb>
2. dpkg -i deb
3. apt-get install -f (If dependency problems occurs)

Further reading : <https://laravel.com/docs/5.4/installation>

2. Create Laravel Project

1. composer create-project laravel/laravel <project_name>

3. Laravel Project Folders

1. app- contains the core code of the application
2. config-configuration files for the application
3. database-database migrations and seeds
4. public--applications document root. also contains assets for the app
5. resources--templates, localization and language files.
6. storage--contains app storage like file uploads,application cache, application logs
7. test & bootstrap
8. vendor--contains the composer dependencies

Further reading : <https://laravel.com/docs/5.4/structure>

4. Running Our Project

1. cd <project_folder>
2. php artisan serve
3. access it on 127.0.0.1:8000 from web browser

5. Routing

1. front gate to the web application.Web routes are found in routes/web.php
2. eg. Route::get('/test',function(){ return "hello test"; });

3. possible http verbs—get,post,put,patch,delete,options
Further reading: <https://laravel.com/docs/5.4/routing>.

6. Introduction to MVC

1. Model `</app>`
2. View `</resources/views>`
3. Controller `</app/http>`
4. `php artisan make:model Task`
5. `php artisan make:controller TaskController`.

7. Database

1. install mysql-server using apt
2. install php7.0-mysql connector
3. create a admin user for mysql-server
 1. `mysql -u root -p`
 2. `CREATE USER 'admin'@'localhost' IDENTIFIED BY 'space123';`
 3. `GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost' WITH GRANT OPTION;`
 4. `FLUSH PRIVILEGES;`
4. install phpmyadmin using apt
5. create a database for the application using phpmyadmin
6. configure .env file in laravel
 1. `DB_DATABASE.`
 2. `DB_USERNAME.`
 3. `DB_PASSWORD.`

8. Migrations.

1. Prepare the Migration File
 1. eg. `php artisan make:migration create_tasks_table --create=tasks`
 2. Add entries as required inside the up function.
 1. Eg `$this->string('name');`
 3. Run Migrations
 1. `php artisan migrate`
 4. Other useful Migrate options
 1. `php artisan migrate:rollback --steps=2`
 2. `php artisan migrate:reset`
 3. `php artisan migrate:refresh`.
- Further reading: <https://laravel.com/docs/5.4/migrations>

9. Seeding

1. php artisan make:seeder TasksTableSeeder
 2. update run() function with appropriate entries
 3. modify DatabaseSeeder file to call TasksTableSeeder
 1. eg. `$this->call(TasksTableSeeder::class);`
 4. run seeder
 1. php artisan db:seed
 5. Along with migration
 1. php artisan migrate --seed
- Further reading: <https://laravel.com/docs/5.4/seeding>

10. Models

1. php artisan make:model Task
2. relationship between Task model and tasks table
3. Tinker REPL for PHP
 1. php artisan tinker
4. seeding using model
 1. eg. `\App\Task::create(['name'=>'task1']);`

11. Controller

1. php artisan make:controller ToDoController
2. sample function inside a controller

```
public function index(){
    $tasks=\App\Task::all();
    return view('index')->with('tasks',$tasks);
}
```

Further reading : <https://laravel.com/docs/5.4/controllers>

12. Blade Templating

1. `{{usage}}`
2. `@foreach` and `@endforeach`
3. eg.

```
@foreach($tasks as $task)
    {{$task->name}}
@endforeach
```

Further reading: <https://laravel.com/docs/5.4/blade>

13. Saving To Database

```
Route::post('/tasks','ToDoController@save');
```

```
public function save(Request $request)
{
    $task = new \App\Task;
    $task->name= $request->name
    $task->save();
    return \Redirect::to('/tasks');
}
```

14. Form with csrf token

```
<input type="submit" value="add" />
<form action="/tasks" method="POST" >
    {{ csrf_field() }}
    <input type="text" name="name" required />
</form>
```

Further reading: <https://laravel.com/docs/5.4/csrf>

15. Auth Plugin

1. php artisan make:auth
2. if error key length
 1. ref: <https://laravel.com/docs/master/migrations#creating-indexes>

16. Route listing.

1. For Knowing what all routes exists in our application , we can try
2. php artisan route:list

17. Plugin Integration

1. eg DomPDF Plugin
 1. composer require barryvdh/laravel-dompdf
 2. php artisan vendor:publish --provider="\Barryvdh\DomPDF\ServiceProvider"
 3. modify config/app.php
 1. service providers
 2. aliases
2. in the view
 1. Generate pdf
3. route
 1. Route::get('/pdf','ToDoController@pdf');
4. controller
5. public function pdf(){
 \$pdf=\App::make('dompdf.wrapper');

```
$pdf->loadHTML('<h1>This is my test paragraph</h1>');  
return $pdf->stream();  
}
```

Further reading: <https://github.com/barryvdh/laravel-dompdf>