Tools

* [www.laravel.com](http://www.laravel.com)

Sublime

* To quickly find a file in sublime use cards/index.blade.php

Model

* Create a model inside Models folder

php artisan make:model Models/Card

* $card = App\Models\Card::first() will return the first row

Laravel Notes

* Use ctrl + c for quit from artisan serve
* In laravel public folder is the document root
* To create a database file touch database/database.sqlite

Laravel Artisan

* Artisan is the name of the command-line interface included with Laravel
* It provides a number of helpful commands for your use while developing your application
* It is driven by the powerful Symfony Console component
* To view a list of all available Artisan commands : php artisan or

php artisan list

* Every command also includes a "help" screen which displays and describes the command's available arguments and options :

php artisan help migrate

* php artisan migrate:refresh rolled back everything & migrated every table again

Laravel Migrations

* Migrations are a type of version control for your database. They allow a team to modify the database schema and stay up to date on the current schema state. Migrations are typically paired with the Schema Builder to easily manage your application's schema.
* Migration is useful when we work within a team then if we change the column name or column laravel keep the record using migration which is useful
* To create migration in laravel php artisan make:migrationcreate\_art
* By laravel migration we can create database schema & modified & seeding database & most important feature is that migration is used for version control.
* Migration naming convention:

php artisan make:migrationcreate\_cards\_table

php artisan make:migrationadd\_username\_to\_users\_table

* To create a migration table with schema

php artisan make:migrationcreate\_cards\_table --create=cards

* Running Migrations or to migrate databaseor for create table in database - php artisan migrate
* To rollback Migrations - php artisan migrate:rollback

Interact with Application

* Command: php artisan tinker
* This command provides a REPL (Read-Eval-Print Loop) for PHP with your application's settings already loaded.
* Created Record by Tinker : DB::table('cards')->insert(['title' => 'My New Card', 'created\_at' => new DateTime, 'updated\_at' => new DateTime]);
* Get data : DB::table('cards')->get();
* Return first Row: DB::table('cards')->where('title', 'My New Card')->first();
* Delete: DB::table('cards')->where('title', 'My New Card')->delete();
* DB::table(‘cards’)->get()
* App\Models\Card::all()

Insert data into database using tinker

* $note = new App\Models\Note;
* $note->body = “Some note for the card”;
* $note->card\_id = 2;
* $note->save();
* App\Models\Note::all();

Fetch Data Using tinker

* First row: $card = App\Models\Card::first();
* Call a method relate with Card $card->notes;
* Representing as an array : $card->notes[0];
* Retrieve first node: $card->notes->first(); here $card->notes fetch all notes then fetch first object.
* $card->notes()->first() i.e fetch all notes but care about just only first() note ignore others
* DB::listen(function($query) { var\_dump($query->sql); });
* After that result return with query
* $card = App\Models\Card::first();
* $card->notes; gives the result with query but in second it didn’t return the query because of laravel associating caching
* $card return all of the database value
* $card->fresh()->notes->first();

Create new node using Eloquent

* $note = new App\Models\Note;
* $note->body = ‘Here is another note.’;
* $card = App\Models\Card::first();
* $card->notes()->save($note);
* Use the relationship to set card\_id in notes table
* $card->addNote($note);
* $card->notes()->create(["body" => 'Yet another note about this card.']);

Use Tinker

* To see the readable array
* $card = new App\Models\Card;
* $card->with(‘notes’)->get();
* Or Use App\Models\Card::with(‘notes’)->get();
* $card = App\Models\Card::with(‘notes.user’)->find(1);
* i.e. call the notes method from the card table & then call the user method from the notes table

Laravel Assets

* All of the laravel assets like css, js put in public directory

Laravel Validation

Laravel : An MVC Framework

* MVC = Model-View-Controller
* Software structure with emphasis on modularity
* Model: Application data and functions
* View: Representation of the output, e.g. HTML
* Controller: Interacts with user, model and view

Laravel’s Real-World Foundation

* Recognizes common tasks required by web apps
* Included functionality (among many others)
* Authentication – Verifying users and handling access
* Routing – Directing requests efficiently
* Database – Management, input and output
* Mail – Sending mail, with attachments and inline files
* Symfony

1. Core functionality: simulation, file access, debugging, etc.

* Composer

1. Dependencies manager; hundreds of packages available

* Eloquent ORM(Object Relational Mapper)

1. Object-oriented database management and migration

Key Laravel components

* Symfony
* Composer
* Eloquent ORM
* Blade – templating engine

Laravel&Symfony

* Symfony: A PHP developmental framework
* Symfony components used in Laravel

1. Browsekit, Console, Debug, FileSystem, et al

* Synchronized future release schedule
* Symfony users can work with Laravel components

Laravel& Composer

* Composer: A PHP dependencies manager
* Works on a per project, not global, basis
* Mac, Windows and Unix compatible
* Declare dependencies in JSON file, composer.json
* Third-party libraries aka packages @ packagist.org
* To learn more about composer check out: [www.getcomposer.org](http://www.getcomposer.org)
* For installing composer – [www.packagist.org](http://www.packagist.org)

Laravel& Eloquent ORM

* Eloquent ORM: Object relational manager for databases
* Developed for Laravel, but can be used outside of it
* Uses ActiveRecord for insert, update & delete
* Handles one-to-one, one-to-many and many-to-many
* Tables created, seeded and managed through migration

Laravel& Blade

* Blade: Templating engine
* Store view with ‘.blade.php’ extension
* Passes variables with {{ variable }} syntax
* Translates PHP code, like {{ date(‘m d, yy’) }}
* Conditional controls and template inheritance supported

Installing Composer

* [www.getcomposer.org](http://www.getcomposer.org)
* Getting started – Installation windows – composer-setup.exe – continue the installation – check it install or not – write composer in cmd – to check it install or not

Installing Laravel

* Install mcrypt-decrypt – download from – for Mac - topicdesk.com/downloads/mcrypt/mcrypt-download
* composer create-project laravel/laravel firstapp - -prefer-dist
* clear screen: mac: ctrl+lwin:cls+enter

Installing laravel in your desired destination

* Redirect to your folder by using just D: for directory change
* For composer update by using git
* Open git bash & write composer self-update
* Then give laravel installation command for installing the laravel project
* If composer not updated then install it again like composer install

Laravel Command

* php artisan serve ; here artisan is laravel command line environment

Laravel Schema

* If laravel is not worked fine then configure following:

First go to .env.php file

Configure following

DB\_HOST=localhost

DB\_DATABASE=laravelu

DB\_USERNAME=root

DB\_PASSWORD=''

* If still not worked fine then run php artisan config:clear