Laravel Cheat Sheet

A quick reference guide for Laravel developers, covering essential commands, concepts, and configurations.

Basic Commands & Concepts

Artisan Commands

`php artisan --version`	Display the Laravel version.
`php artisan make:controller ControllerName`	Create a new controller.
`php artisan make:model ModelName`	Create a new Eloquent model.
`php artisan make:migration create_table_name_table`	Create a new migration file.
`php artisan migrate`	Run pending migrations.
`php artisan migrate:rollback`	Rollback the last migration.
`php artisan serve`	Start the built-in PHP development server.
`php artisan tinker`	Enter the interactive Tinker shell.
`php artisan route:list`	Display all registered routes.

Directory Structure

`app/` - Contains the core code of your application.
`bootstrap/cache/` - Framework bootstrap files.
`config/` - Application configuration files.
`database/` - Database migrations and seeds.
`public/` - Publicly accessible files (CSS, JavaScript, images).
`resources/views/` - Application views (Blade templates).
`routes/` - Route definition files (web.php, api.php).
`storage/` - Storage directory for files and sessions.

Routing and Controllers

Basic Routing

```
Route::get('/route', function () {
    return 'Hello, World!';
});
```

```
Route::post('/route', function () {
    // Handle POST request
});
```

Route Parameters

Required parameter.

```
Route::get('/user/{id}',
function ($id) {
    return 'User ID: ' . $id;
});
```

Optional parameter.

```
Route::get('/user/{name?}',
function ($name = null) {
    return 'Name: ' . $name;
});
```

Controllers

```
Route::get('/users',
'App\Http\Controllers\UserController@index');
```

```
namespace App\Http\Controllers;

use App\Http\Controllers\Controller;

class UserController extends Controller
{
    public function index()
    {
        return view('users.index');
    }
}
```

Eloquent ORM

Basic Model Operations

Get all records.

```
$users = App\Models\User::all();
```

Find a record by primary key.

```
$user = App\Models\User::find(1);
```

Create a new record.

```
$user = new App\Models\User;
$user->name = 'John Doe';
$user->email = 'john@example.com';
$user->password = bcrypt('secret');
$user->save();
```

Update an existing record.

```
$user = App\Models\User::find(1);
$user->name = 'Jane Doe';
$user->save();
```

Delete a record.

```
$user = App\Models\User::find(1);
$user->delete();
```

Relationships

`One To One`: A user has one profile.

```
public function profile()
{
    return $this->hasOne('App\Models\Profile');
}
```

`One To Many`: A user has many posts.

```
public function posts()
{
    return $this->hasMany('App\Models\Post');
}
```

`Many To Many`: A post has many tags.

```
public function tags()
{
    return $this->belongsToMany('App\Models\Tag');
}
```

Blade Templating

Basic Syntax

`{{ $variable }}`	Display a variable (automatically escaped).
`{!! $variable !!}`	Display a variable without escaping.
`@if (condition)` `@endif`	Conditional statement.
`@foreach ($items as $item)` `@endforeach`	Looping.

Components and Layouts

`@extends('layouts.app')` - Extend a layout.
`@section('content')` ... `@endsection` - Define a section.
`@include('partials.header')` - Include a partial view.
`@component('components.alert')` ... `@endcomponent` - Render a component.

Directives

`@auth` ... `@endauth`	Check if the user is authenticated.
`@guest` ... `@endguest`	Check if the user is a guest.
`@csrf`	Generate a CSRF token field.