

RUTAK

Metodoak

[Oinarrizko bideratzea \(Enrutamiento básico\)](#)

[Ibilbide anitz](#)

[Rutak parametroekin](#)

[Exekuzio denboran emandako akatsak eta ibilibideen errestrikzioak](#)

[Izena duten rutak](#)

[Ibilbideen ordena](#)

[Ibilbide-taldeak kontrolatzilean](#)

[Resource ibilibideak](#)

Laravelek erabiltzaile eta zerbitzari arteko erlazioa egiteko ruta edo ibilibideen bidez egiten du.

Laravelen ibilibide-sistema nahiko erraza da, eta, beraz, pausu gutxitan, mota guztietako ibilibide nahiko konplexuak egin ditzakegu.

Guk ruta orokorrak ikusiko ditugu, ala ere dokumentazio guztia hurrengo estekan duzu: <https://laravel.com/docs/10.x/routing>

Metodoak

Sartzen den informazioa interpretatzen jakin behar du Laravel aplikazioak, eta nabarmen erabili bezeroak eskatzen duena itzultzeko.

Funtsezkoa da HTTP metodoaren metodo nagusiak ezagutzea. Hurrengo hauek dira Laravelek erabiltzen dituen **HTTP aditzak**:

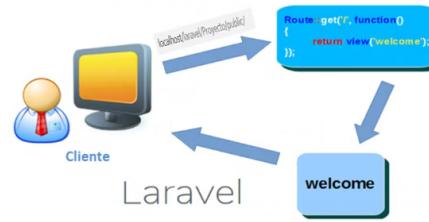
- **Get:** Segurenekin ruta mota hau da gehien erabiliko dena, hauek aldagaiak bidaltzen ez direnean edota get metodoaren bidez bidaltzen dugunean erabiltzen dra.
- **Post:** orrian bertan dauden datuak bidaliko ditu, datu hauek ez dira eskaeran ikusiko, protokooloaren informazioan bidaiatzen dute eta.
- **Put:** Normalean put datuak aldatu nahi dugunean erabiltzen da, update bat egiteko adibidez.
- **Delete:** zeozer ezabatzeko erabiltzen da ruta mota hau
- **Trace, patch, link, unlink, options y connect...**: Hauek ez dira asko erabiltzen, baina egon badaude.
- **Any:** hau komodin bezala erabiliko da, edozein aditzerako balioko du

Gure ruta edo ibilibide guztiak **route/web.php**-n idatziko ditugu, hor egongo bai dira interfaz web erabiltzen diren ruta guztiak.

Laravel instalatzean, lehenengo ruta aurkitzen duguna hurrengo hau da:

```
Route::get('/', function () {
    return view('welcome');
});
```

Hau izango da ruta orokorra da, kasu honetan welcome bista erakutsiko duen ruta.



Hau da kontsulta zikloa

Oinarrizko bideratzea (Enrutamiento básico)

Laraveleko ruta guztiak, erroko direktorioa izango dute oinarri. Beraz, gure zerbitzari lokalean, aurreko rutara deitzeko, gure nabegatzailean horrelako zeozer idatziko genuke:

http://localhost/LARAVEL_PROIEKTUAK/eskola/public/

Beste oinarrizko ruta batzuk ondorengo hauek dira adibidez:

```
Route::get('/probando/ruta', function(){
    //código a ejecutar cuando se produzca esa ruta y el verbo
    return 'get';
});
Route::post('/probando/ruta', function(){
    //código a ejecutar cuando se produzca esa ruta y el verbo POST
    return 'post';
});
Route::put('/probando/ruta', function(){
    //código a ejecutar cuando se produzca esa ruta y el verbo PUT
});
```

```

        return 'put';
    });
Route::delete('/probando/ruta', function(){
    //código a ejecutar cuando se produzca esa rutay el verbo DELETE
    return 'delete';
});

```

Egon daiteke momentu bat, aditz bat baino gehiago erabili nahi duzula zure rutara deitzeko, horrela izatekotan hurrengo adibidean agertzen den antzeko zeozer idatziz beharko zenute, ikusten denez, beste parametro bat pasatu beharko dugu erabiliko diren aditz guztiak duen arraya:

```

Route::match(['get', 'post', 'put'], '/testing', function () {
    echo 'Ruta testing para los verbos GET, POST, PUT';
});

```

Bestetik ere any izango dugu, edozein aditzerako baliogarria izango dena:

```

Route::any("/cualquiercosa", function(){
    echo 'La ruta /cualquiercosa asociada a cualquier verbo';
});

```

Ibilbide anitz

Imagina dezakegungo badugula web bat hiru orriekin, hau da, horrelako zeozer:

1. <http://monsite.fr/1>
2. <http://monsite.fr/2>
3. <http://monsite.fr/3>

Horretarako 3 ibilibide sortu beharko genitzuke, bat orri bakoitzeko:

```

Route::get('1', function () { return 'Soy la página 1!'; });
Route::get('2', function () { return 'Soy la página 2!'; });
Route::get('3', function () { return 'Soy la página 3!'; });

```

Ikusten dugunez, ruta hauek ez dute bista bat erakusten, bakoitzak esaldi bat aterako du soilik.

Lehenengo orri sartzeko, ez bagenuke domeinurik izango, horrelako zeozer izango zen:

<http://localhost/proyecto/public/1>

Baina, hau ikusita, beharko genuke benetan 3 ibilibide sortu, aldaketa bakarra aldatzen den balioa denean?

Rutak parametroekin

Ibilbide edo ruta bat parametrizada dagoela esaten dugu, edozein balio hartu dezakeenean.

```

Route::get('{n}', function($n) {
    return 'Soy la pagina ' . $n . ' !';
});

```



Izan daiteke balio hau aukerakoa izatea, kasu horretan, galdera ikurra izan beharko du:

```

Route::get('{n?}', function($n = 1) {
    return 'Soy la pagina ' . $n . ' !';
});

```

Adibidean ikusi daitekeenez, kasu honetan ere, lehenetistako balioa jarri dugu, hau da, parametrorik ez izanez gero, n-k 1 balioa hartuko du.

Exekuzio denboran emandako akatsak eta ibilibideen errestrikzioak

Aurreko abidideetan, horrelako zeozer jartzeraoa 404 errorea emango liguke:

<localhost/laravel/Proyecto/public/4>

404 | Not Found

Baina aldiz, lehenetsiko balioa jarri dugun horretan ez luke errorea emango.

Bestetik frogatu dezakegu zelan guk jarri dugun moduan, edozein balio sar daitekeen parametro moduan, 'n' horrek, edozein balio hardezake, hurrengo hau jarri ezkerlo:

<localhost/laravel/Proyecto/public/cualquier cosa>

Hau izango zen emaitza:

`;Soy la página cualquier cosa!`

Izena duten rutak

Laravelek aholkatzen digu gure rutei izenak ematea beti, horrela errezago egingo zaigu hauel deitza; hurrengo adibidean ruta horri deitzeko 'home' soilik idatzi beharko genuke.

Sintaxia hurrengo hau izango litzateke:

```
Route::get('/', function() {
    return 'Soy la pagina de inicio!';
})->name('home');
```

Ibilbideen ordena

Oso garratzitsua da jakitea zein den ibilbideen ordena exekutatzerakoan, guk nahi duguna bistaratu dezan.

Araua hau da: **Ibilbideak ruta fitxategian duten ordenean analizatuko dira.**

Hurrengo kasuan adibidez:

```
Route::get('{n}', function($n) {
    return 'Soy la pagina ' . $n . ' !';
});
Route::get('contact', function() {
    return Este es mi contacto.';
});
```

Lehenengo rutan, 'n'-k edozein balio har dezakeenez, 'contact' hitza ere hartu dezake, beraz, guk, *localhost/laravel/Proyecto/public/contact* jarriz gero, emaitza: Soy la pagina contact izango da, "Este es mi contacto" izan beharrean. Beraz, ruta biak ondo funtzionatu dezaten, lehenengo 'contact' ruta jarri beharko genuke eta ondoren bestea.

Ibilbide-taldeak kontrolatzailean

Laravel 9-tik aurrera, **controller()** metodoa erabil dezakezu Kontroladore baten ruta guztiek honen barruan izateko. Horrela eginez gero, kodea txukunago geratuko litzateke.

Adibidez CursoController kontroladorean exekutatuko diren ruta guztiek elkarrekin batzeko, group metodoa erabiliko genuke hurrengo adibidean moduan

```
Route::controller(CursoController::class)->group(function () {
    Route::get('cursos', 'index');
    Route::get('cursos/create', 'create');
    Route::get('cursos/{curso}', 'show');
});
```

Resource ibilbideak

Laravelen ere, gure kontroladoreak resource metodoarekin egiterakoan, ikusi genuen zelan sortu zizkigun hainbat metodo kontroladore horren bidez CRUDa egin ahal izateko, ba kontroladore horren barruan dauden metodo guztii horiek ibilbide bakar bategaz egin dezake, metodo hori **resource** metodoa da:

```
Route::resource('ikasleak', IkasleController::class);
```

Ikusten dugunez, honek asko errazten digu gure ibilbideen kudeaketa, Laravelek sortzen bait ditu beharrezko ruta guztiek. Hau konprobatzeko, artisanen bidez begiratu dezakegu sortuta ditugun ruta guztiek:

```
php artisan route:list
```

Laravelek baditu hainbat kontzentzio rutak sortzerako momentuan, izenak modu zehatz baten idazteko, hauek ez dira derrigorrezkoak, baina bai oso gomendagarriak. Baino resource-ren bidez sortuz gero, kontroladorearen metodoek bai jarraitu behar dituztela kontzentzio horiek: create, store, destroy etab...

