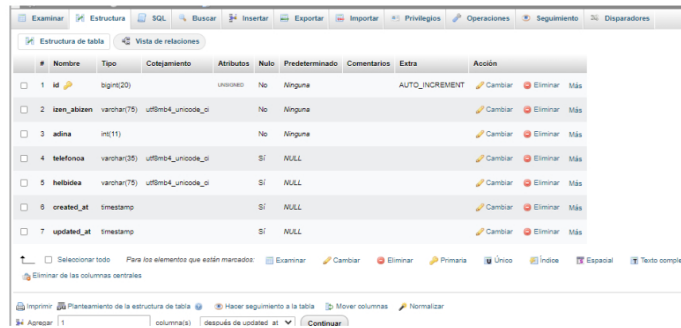


BALIDAZIOAK

Balidazioak

Laravelek baditu txertaturik balidazioetarako balio zaizkigun funtzio batzuk. Hauek balidazioak datu basean datu bat sartu baino lehen egingo ditugu.

Ezer baino lehen, begiratuko dugu gure datu basean zeintzuk datu dira derrigorrezkoak, zeintzuk dira datu horien mota etab.



#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	bigint(20)	UNSIGNED	No	Ninguna			AUTO_INCREMENT	Cambiar Eliminar Más
2	izen_abizen	varchar(75)	utf8mb4_unicode_ci	No	Ninguna				Cambiar Eliminar Más
3	adina	int(11)		No	Ninguna				Cambiar Eliminar Más
4	telefonoa	varchar(35)	utf8mb4_unicode_ci	SI	NULL				Cambiar Eliminar Más
5	helbidea	varchar(75)	utf8mb4_unicode_ci	SI	NULL				Cambiar Eliminar Más
6	created_at	timestamp		SI	NULL				Cambiar Eliminar Más
7	updated_at	timestamp		SI	NULL				Cambiar Eliminar Más

1. Akatsak *store()* metodoaren barruan kontrolatuko ditugu. Formularioko eremu guztiak dituen **store** funtzioaren parametro gisa iragaten den **request** aldagaia erabiliko dugu. **Validate()** metodoari array bat emango diogu, formularioko eremuak eta baliozkotzeak dituen.

Laravelek adierazi egiten du, momentu baten baten balidazioa betetzen ez baldin bada, ez da hurrengo kode lerroa pasatuko eta akatsa emango digu. Aldiz, balidazio guztiak ondo daudenean aurrera jarraituko du.

Gure kasuan, *store()* metodoan idatziko dugu, izena eta adina balidatuko ditugu:

```
$request->validate([
    'izen_abizen' => 'required|max:75',
    'adina' => 'required|integer',
]);
```

Akatsak mezuetan adierazi

2. Orain, gure inprimakiak ikusiko ditugu. Hor inprimatuko ditugu mezuak formularioko eremuetan, gure kontrolatzaileetan definitzen ditugun baliozkotze-akatsak daudenean.

3. Inprimakiak baliozkotzen hasi aurretik, css kodea erantsiko diegu gure ikuspegi guztiei, errore-mezuiei estiloa emateko.

```
p { color: red; }
```

Inprimakia ikasleen ikuspegian (bistan) baliozkotzea

4. **berria.blade.php** bistan sortuko ditugu balidazio guztiak.

Input elementuaren azpiko *Izen-deiturak* eremuan, eremu honetako baliozkotze-kodea gehitzen dugu Bladek eskaintzen digun zuzentarauarekin. **@error** aukerak eremu honi dagokion errore-mezua inprimatzeko aukera ematen digu.

```
@error('izen_abizen')
    <p class="error-message">{{ $message }}</p>
@enderror
```

5. Gure input-ari gehituko diogu **old()** metodoa, blade-ko direktiba bat dena. Honegaz lortuko dugu balidazioa egin baino lehen sartutako datua mantentzea.

```
<input type="text" name="izen_abizen" placeholder="Izen abizena" value="{{ old('izen_abizen') }}">
```

6. Gauza berdina adina elementaun egingo dugu:

```
<label>Adina:</label>
<input type="text" name="adina" placeholder="Adina" value="{{ old('adina') }}">
@error('adina')
    <p class="error-message">{{ $message }}</p><br>
@enderror
```

7. Sartutako telefonoa eta helbidea ere mantenduko ditugu:

```
<label>Telefonoa:</label>
<input type="text" name="telefonoa" placeholder="Telefonoa" value="{{ old('telefonoa') }}">
<label>Helbidea:</label>
<input type="text" name="helbidea" placeholder="Helbidea" value="{{ old('helbidea') }}">
```

8. Zer gertatzen da, **balidazioa editatzerakoan egiten baldin badugu**? Kasu horretan, formularioara sartzen garen lehenengo aldian, datu baseko datua aterata nahi dugu, eta akatsa egotekoran **old** metodoan bueltatzen duguna, hori egiteko, **old** metodoari bigarren parametro bat asatuko diogu:

```
<label>Telefonoa:</label>
```

```
<input type="text" name="telefonoa" placeholder="Telefonoa" value="{{ old('telefonoa', $ikaslea-telefonoa) }}">
```

Ikusten dugunez, mezuak ingeleraz aterako dira, hori aldatu nahi izatekotan, hizkuntza aldatu beharko dugu.

Form Request

Gure aplikatiboan balidazio asko egiten baldin baditugu, gure kodea asko zikindu daiteke, horretarako form request erabili dezakegu, honek esan nahi du gure balidazio kodea fitxategi berri baten jarriko ditugula.

Terminala zabaltzen dugu eta hurrengo kodea idatziko genuke, kasu honetan Ikasle kontroladorean store metodoari egingo dugun balidazioa sartzeko:

```
php artisan make:request StoreIkasle
```

Ikusiko dugu zelan fitxategi berri bat sortu zaigu *Http>Request* karpetan

Fitxategi honetan klase bat sortu da, StoreIkasle izenekoa bi metodo dituena. *authorize()* eta *rules()* metodoa. Lehenengo izango litzateke kontrolatzeko ia erabiltzailea baimenak dituen ala ez, baina normalean ez da horreta egiten, policekin egin oi da, beraz guk honetan tru balioa jarriko dugu, hurrengo metodoa exekutatu ahal izateko.

Eta *rules()* metodoan gure kontroladorean ditugun balidazioak jarriko ditugu:

```
{
    /**
     * Determine if the user is authorized to make this request.
     */
    public function authorize(): bool
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string, \Illuminate\Contracts\Validation\Validator
     */
    public function rules(): array
    {
        return [
            'izen_abizen'=> 'required|min:3',
            'adina'=> 'required'
        ];
    }
}
```

Kontroladorean ordun balidazioak kenduko genituzke, eta gure store() metodoan Request objektuaren instantzia bat izan beharrean, esango diodu StoreRequest objektuaren intantzia izatea.

```
public function store(StoreIkasle $request){
```

Balidazioa idazten dituen mezuak aldatu nahi baditugu, adibidez, euskaratu, message() metodo bat sortuko genuke, hurrengo adibidean ikusten dugun moduan:

```
public function rules(): array
{
    return [
        'izen_abizen'=> 'required|min:3',
        'adina'=> 'required'
    ];
}

public function messages():array{
    return [
        'izen_abizen.required'=> 'Izen abizena sartzea derrigorrezkoa da',
        'izen_abizen.min'=> 'Gutxienez 3 hizki izan behar ditu',
        'adina.required'=> 'Adina derrigorrezkoa da'
    ];
}
```

Atributuen izenak ere aldatu ditzakegun hemen, horretarako attributes() metodoa erabiliiko genuke.

Laravelen balidazio guztiak hurrengo estekan izango dituzue: <https://laravel.com/docs/10.x/validation#available-validation-rules>