

## - Duplicate Field

Possiamo agilmente *duplicare* un record e applicare anche dei cambiamenti a scelta.  
Nella *bash*:

```
composer require jackabox/nova-duplicate-field
```

Nella *resource* Article:

```
use Jackabox\DuplicateField\DuplicateField;

class Article extends Resource{
    return [
        ...
        DuplicateField::make('Duplicate')
            ->withMeta([
                'resource' => 'articles',
                'model' => 'App\Models\Article',
                'id' => $this->id,
                'relations' => [],
                'except' => ['img'],
                'override' => ['published' => false]
            ]),
    ];
}
```

**except** . Campi da *non* replicare.

**override** . Campi che avranno un campo settato come da indicazione.

## - Spatie Laravel Tags

Possiamo creare un sistema di **tags** associabile ad un Articolo o ad un Utente volendo.  
Installiamo il pacchetto e *pubblichiamo le migrazioni*, visto che creeremo una nuova tabella *Pivot*.

- Nella *Bash*:

```
composer require spatie/laravel-tags
php artisan vendor:publish --provider="Spatie\Tags\TagsServiceProvider"
--tag="migrations"
```

- Nel file *Migration* appena creato ci sono le istruzioni per la creazione di due tabelle.

Una **tags** e l'altra **taggables**.

La *taggable* e' la **pivot** e contiene la relazione *Many to Many* tra gli Article/User e i Tag.

Nel **taggable\_type** specifichiamo il *Model* che abbiamo relazionato.

- Facciamo la Migration:

```
php artisan migrate
```

Per Usare un Tag seguiamo la *documentazione*:

Link - <https://spatie.be/docs/laravel-tags/v2/basic-usage/using-tags>

- Nel *Model* Article e User (o a qualsiasi *model* vogliamo dare dei *tag*) aggiungiamo il **trait**:

```
use Spatie\Tags\HasTags;

class Article extends Model{
    use HasTags;
}
```

- Possiamo testare con Tinker l'**attach** o il **detach** di un Tag:

```
php artisan tinker
$a = App\Models\Article::where('id', 1)->first();
$a->attachTag('prova');
$a->detachTag('prova');
```

- Nelle *view* in *frontend* grazie al **trait** possiamo richiamare così i *tags*:

```
@foreach($article->tags as $tag)
    <span>{{ $tag->name }}</span>
@endforeach
```

Fino ad ora abbiamo agito **lato Laravel** utilizzando la libreria Spatie.  
Adesso creiamo il *field* per inserire i *tag* lato **Nova**.

- Nella *Bash*:

```
composer require spatie/nova-tags-field
```

- Nella *resource*:

```
use Spatie\TagsField\Tags;

class Article extends Resource{
    return [
        ...
        Tags::make('Tags'),
    ];
}
```

Ora nella modifica del *detail* abbiamo tutte le funzionalità di Spatie per i **tag**.