


## Route Model Binding con Eloquent

## Sidrit Trandafli

 @sidis405 @strandafli

- Sviluppatore PHP 15+ anni
- **Laravel** da 4 anni
- Back-end
  - php
  - java
  - node
  - python
- Front-end
  - VueJS
  - Vanilla JS
  - TailwindCSS
  - Sass
- Devops + Sysops da 13 anni



# Cos'è il Route Model Binding?



## Route Model Binding

*“ È il meccanismo di Laravel che ci permette di iniettare un'istanza di un modello Eloquent direttamente in una route*



# Route Model Binding

Immaginiamo di dover prendere un utente nel database

```
...  
// il parametro 'id' è l'id dell'utente  
// esempio http://mioblog.it/utenti/20  
Route::get('utenti/{id}', function ($id) {  
  
    // troviamo l'utente usando il parametro $id  
    $utente = User::find($id);  
  
    // se non esiste  
    if (!$utente) return abort(404);  
  
    // restituiamo la view e l'utente trovato  
    return view('utenti.show', compact('utente'));  
});  
...
```



# Route Model Binding

Possiamo semplificare nel seguente modo

```
...  
// il parametro 'id' è l'id dell'utente  
// esempio http://mioblog.ite/utenti/20  
Route::get('utenti/{id}', function ($id) {  
  
    // troviamo l'utente usando il parametro $id  
    $utente = User::findOrFail($id);  
  
    // restituiamo la view e l'utente trovato  
    return view('utenti.show', compact('utente'));  
});  
...
```



# Route Model Binding

Il Route Model Binding ci permette di semplificare il tutto ulteriormente - (routing implicito)

```
...
// il parametro 'id' è l'id dell'utente
// esempio http://mioblog.ite/utenti/20
Route::get('utenti/{id}', function ($utente) {

    // abbiamo già l'utente $utente

    // restituiamo la view e l'utente trovato
    return view('utenti.show', compact('utente'));
});
...
```

Nota:

```
return view('utenti.show', compact('utente'));
```

È equivalente a:

```
return view('utenti.show', ['utente' => $utente]);
```

e:

```
return view('utenti.show')->with('utente' => $utente);
```

È una funzione **php** ed è solo una preferenza :)



# Route Model Binding

*“ Laravel permette il route model binding*

- implicito
- esplicito





# Route Model Binding

Vediamo un esempio di routing implicito

```
Route::get('utenti/{utente}', function (App\User $utente) {  
    // L'oggetto $utente è stato reperito dal db  
});
```

1. Laravel ha capito che il parametro :utente è un oggetto di tipo App\User
2. ha reperito il valore 'id' da :utente
3. ha eseguito una query App\User::find('chiave-modello-utente'); // id

Cosa succede se il nostro modello non si identifica per id?

```
// :post è lo slug del post es: mioblog.it/posts/il-prim-post  
  
Route::get('posts/{post}', function (App\Post $post) {  
    // ???  
});
```



# Route Model Binding

Modifichiamo la chiave identificativa del modello App\Post

```
class Post extends Model {  
    public function getRouteKeyName() {  
        return 'slug';  
    }  
}
```

1. Laravel capisce che :post è un oggetto di tipo App\Post
2. ha reperito il valore 'slug' da :post // tramite \$model->getRouteKeyName()
3. ha eseguito una query App\Post::find('chiave-modello-utente'); // **slug**

```
// :post è lo slug del post es: mioblog.it/posts/il-primo-post  
  
Route::get('posts/{post}', function (App\Post $post) {  
    // L'oggetto post identificato per slug adesso è disponibile  
});
```



# Route Model Binding

In uso in un controller

```
// routes/web.php
```

```
Route::get('posts/{post}', 'PostsController@show');  
Route::get('posts/{post}/edit', 'PostsController@edit');  
Route::post('posts/{post}', 'PostsController@update');  
Route::delete('posts/{post}', 'PostsController@destroy');
```

È possibile fare questa azione manualmente ma **artisan** ci aiuta :)

```
php artisan make:controller PostsController -r -m Post
```

**-m** specifica il modello per il quale creare il controller

**-r** specifica che il controller deve essere **resourceful** - dunque index, create, show etc

Se il modello Post non esiste, verrà creato



# Route Model Binding

In uso in un controller

```
// routes/web.php
```

```
Route::get('posts/{post}', 'PostsController@show');  
Route::get('posts/{post}/edit', 'PostsController@edit');  
Route::post('posts/{post}', 'PostsController@update');  
Route::delete('posts/{post}', 'PostsController@destroy');
```

È possibile fare questa azione manualmente ma **artisan** ci aiuta :)

```
php artisan make:controller PostsController -r -m Post
```

**-m** specifica il modello per il quale creare il controller

**-r** specifica che il controller deve essere **resourceful** - dunque index, create, show etc

Se il modello Post non esiste, verrà creato

```
public function show(Post $post)  
{  
    //  
}
```

Potremmo usare resources nel file routes/web.php e non dover specificare ogni route

```
Route::resource('posts', 'PostsController');
```



# Route Model Binding

Modello restful - Metodologia REST

Verbo	uri	Metodo
GET	/utenti	UserController@index
GET	/utenti/crea	UserController@create
POST	/utenti	UserController@store
GET	/utenti/{utente}	UserController@show
GET	/utenti/{utente}/modifica	UserController@edit
PUT	/utenti/{utente}	UserController@update
DELETE	/utenti/{utente}	UserController@destroy