RESTful CRUD - edit

3 CRUD - Aggiornare un utente esistente

Analizziamo la rotta edit

URI	Name	Action
GET /users/{user}/edit	users.edit	UserController@edit

Metodo GET
URI /users/{user}/edit

Nome della rotta
users.edit

Azione / Controller e funzione da chiamare
UserController@edit

Edit - Controller

L’edit è molto simile a ciò che già abbiamo visto con il create.
Consiste in una chiamata GET in cui dobbiamo ritornare una view con il medesimo form della funzione create.

C'è solamente qualche piccola differenza: dobbiamo passare il record che vogliamo editare.
In questo modo nel nostro form andiamo ad inserire i valori del record, così da far capire a chi deve modificare l'entità quali sono i valori attuali.

UserController.php

```php
use App\Models\User;

public function edit(User $user) {
    return view('users.edit', compact('user'));
}
```

Qual è il metodo giusto?

POST, PUT o PATCH?

Sappiamo già che nell'attributo action dobbiamo inserire
la rotta che deve gestire i dati del form e aggiornare l'entità.
Nel nostro caso si tratta della rotta **users.update**

Stando ai dettami REST, questa rotta deve essere
richiamata esclusivamente con i metodi **PUT** o **PATCH**.

Gli unici valori possibili per l'attributo method del tag form
però sono solo GET e POST.

Come facciamo a costruire la rotta giusta?

@method

Nell'attributo method del tag form non abbiamo altra scelta se non usare POST.

Per far sì che Laravel richiami il giusto Controller, possiamo usare la direttiva **@method** di Blade, specificando come parametro il metodo da utilizzare per la mappatura corretta della rotta.

Edit - Esempio di form

Inseriamo nell'attributo action la rotta **update, passando l'id dell'utente da modificare**.

Nell'attributo method del tag form inseriamo il valore POST e poi aggiungiamo all'interno del form la direttiva **@method('PUT')** di Blade.

Ricevendo l'entità da modificare dal controller, possiamo inserire i valori preesistenti utilizzando l'**attributo value** di ogni input.

edit.blade.php

```
1  <form
2   action="{{ route('users.update', ['id' => $user->id])
   }}"
3   method="POST">
4     @csrf
5
6     {{-- aggiungiamo il metodo --}}
7     @method('PUT')
8
9     <label for="name">Nome</label>
10    <input type="text" name="name" id="name"
11      value="{{ $user->name }}">
12
13    <label for="lastname">Cognome</label>
14    <input type="text" name="lastname" id="lastname"
15      value="{{ $user->lastname }}">
16
17    <input type="submit" value="Invia">
18 </form>
```

RESTful CRUD - update

3 CRUD - Aggiornare un utente esistente

Analizziamo la rotta update

URI	Name	Action
PUT|PATCH /users/{user}	users.update	UserController@update

Metodo PUT o PATCH
URI /users/{user}

Nome della rotta
users.update

Azione / Controller e funzione da chiamare
UserController@update

Il method PUT in cosa si distingue dal PATCH?

Update - Controller

Così come in store, anche nel metodo update() **leggiamo i dati** passati tramite il form con la funzione **all()** della classe **Request**.

Usiamo quindi questi dati per aggiornare l'**entità User** tramite il metodo **update()** della classe **Model**.

UserController.php

```php
public function update(Request $request, User $user) {
    $data = $request->all();

    $user->update($data);

    return redirect()->route('users.show', $user->id);
}
```

LIVE CODING

Aggiorniamo un record della nostra tabella

RESTful CRUD - destroy

4 CRUD - Cancellare un utente

Analizziamo la rotta destroy

URI	Name	Action
DELETE /users/{user}	users.destroy	UserController@destroy

Metodo DELETE
URI /users/{user}

Nome della rotta
users.destroy

Azione / Controller e funzione da chiamare
UserController@destroy

Attenzione: il **metodo** è **DELETE** ma la **rotta** e la relativa **funzione** nel controller si chiamano **destroy**

Destroy - Controller

Destroy con metodo DELETE ci servirà per cancellare un record dalla nostra tabella.

Prenderemo come sempre il nostro record e chiameremo il metodo **delete()**.

UserController.php

```php
public function destroy(User $user)
{
    $user->delete();

    return redirect()->route('users.index');
}
```

Destroy - Controller

Per cancellare un record, possiamo lavorare direttamente nella tabella della **view index**, aggiungendo **un pulsante in ogni riga**.

Questo pulsante in realtà sarà il **submit di un form** con nessun input all'interno ma solamente la **action** e la direttiva **@method** compilati correttamente.

index.blade.php

```
1  <form
2   action="{{ route('users.destroy', ['id' => $user->id]) }}"
3   method="POST">
4     @csrf
5
6     {{-- aggiungiamo il metodo --}}
7     @method('DELETE')
8
9     <input type="submit" value="Cancella">
10 </form>
```

LIVE CODING

Eliminiamo un record dalla nostra tabella

ESERCITAZIONE