



31 maggio

Caricare un file in un applicativo.

Carichiamo i nostri file nella `cartellastorage/app/public`

Dobbiamo creare un collegamento tra la cartella `public` e lo storage tramite un symlink:

```
php artisan storage:link
```

Verrà creata una cartella `storage` che sarà sincronizzata con la cartella `storage` in `app/public`.

Per caricare un file dobbiamo ricordare di inserire l'attributo `enctype` dopo il `method` del form.

```
enctype="multipart/form-data"
```

e cambiare se già c'è l'input in file.

In questo modo avremo a disposizione la possibilità di caricare un file dal pc.

Nel nostro controller
usiamo la funzione
Storage::put() per caricare
un file nella cartella
storage/app/public

```
use Illuminate\Support\Facades\Storage;

class PostController extends Controller
{
    public function store()
    {
        // ...
        $img_path = Storage::put('uploads', $data['image']);
        // ...
    }
}
```

Includiamo sempre la facade storage nel controller.

Controllo che verifica se l'immagine è stata selezionata. Se sì, la carica, altrimenti nulla.

```
if ( isset($data['image']) ) {
    $data['image'] = Storage::disk('public')->put('images', $data['image']);
}

// Insert
```

Inviare una mail

Configuriamo il server smtp tramite mailtrap.

Logghiamoci, creiamo un nuovo form/mail content e selezioniamo Laravel come framework.

Copiamo tutte le stringhe/var e copiamole nel file.env al posto di quelle esistenti fino alla stringa encryption.

Possiamo anche aggiungere un email address.

Generiamo una mail tramite laravel adesso.

Creiamo una mail

Ed una relativa **view** da usare nella funzione **build()**

Nella view possiamo usare Plain Text, Blade o Markdown

```
php artisan make:mail SendNewMail
```

```
class SendNewMail extends Mailable
{
    use Queueable, SerializesModels;

    public function __construct()
    {
        //
    }

    public function build()
    {
        return $this->view('view.name');
    }
}
```

Ci creerà un nuovo controller con un costruttore ed una view. Possiamo creare nelle views una nuova cartella, mail ad esempio e dare il path nella view.

Ora mandiamo la **Mail** da un **Controller**



```
use App\Mail\SendNewMail;
use Illuminate\Support\Facades\Mail;

Mail::to('mail@mail.it')->send(new SendNewMail());
```