

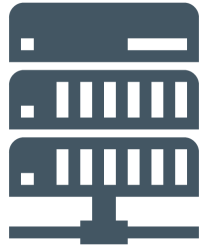


Gestire lo storage dei file

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



File Storage



File Storage

“ Laravel dispone di un livello di astrazione di storage basato su [Flysystem](#). Questa libreria permette di lavorare con lo storage dei file in locale, su Amazon S3, e Rackspace. Cambiare da uno all'altro non comporta cambiamenti di sintassi nel codice.

```
'disks' => [  
  
    'local' => [  
        'driver' => 'local',  
        'root' => storage_path('app'),  
    ],  
  
    'public' => [  
        'driver' => 'local',  
        'root' => storage_path('app/public'),  
        'url' => env('APP_URL').'/storage',  
        'visibility' => 'public',  
    ],  
  
    's3' => [  
        'driver' => 's3',  
        'key' => env('AWS_ACCESS_KEY_ID'),  
        'secret' => env('AWS_SECRET_ACCESS_KEY'),  
        'region' => env('AWS_DEFAULT_REGION'),  
        'bucket' => env('AWS_BUCKET'),  
    ],  
  
],
```

I 'dischi' di storage vengono gestiti in config/filesystems.php



File Storage

Public Disk

Il disco **public** è inteso come il disco dove verranno salvati i file accessibili pubblicamente. Di default, il disco **public** usa il driver local e salva i file in `storage/app/public`. Per renderli accessibili bisogna creare un link simbolico

da `public/storage` a `storage/app/public`

```
php artisan storage:link
```



File Storage

Caricare un file

Consideriamo il seguente form per caricare un avatar

```
<form action="/avatar" method="POST" enctype="multipart/form-data">
  {{ csrf_field() }}

  <input type="file" name="avatar">

  <button type="submit"></button>
</form>
```

Salvare questo file su disco è facilissimo

```
public function update(Request $request)
{
    $path = $request->file('avatar')->store('avatars');

    return $path;
}
```

Estraiamo il file dalla richiesta

Specifichiamo la cartella dove
salvarla. Restituiamo il path del
file



File Storage

Controllo granulare : salvataggio file

Specifichiamo un filename

```
$fileurl = $request->file('avatar')  
->storeAs('avatars', $request->user()->id);
```

```
$fileUrl = Storage::putFileAs(  
    'avatars',  
    $request->file('avatar'),  
    $request->user()->id  
);
```

Dato che la funzione store, userà il vostro disco predefinito, è dunque possibile specificare un altro disco durante il salvataggio file

```
$filePath = $request->file('avatar')->store(  
    'avatars/' . $request->user()->id, 's3'  
);
```

```
$filePath = Storage::disk('s3')  
->putFile('avatars',  
    $request->file('avatar'));
```



File Storage

Controllo granulare : reperire file

Dal disco di default, specificando il path relativo

```
use Illuminate\Support\Facades\Storage;  
  
...  
  
$avatar = Storage::get('avatar.jpg');
```

La funzione userà il vostro disco predefinito, è dunque possibile specificare un altro disco

```
use Illuminate\Support\Facades\Storage;  
  
...  
  
$avatar = Storage::disk('s3')->get('avatar.jpg');
```




File Storage

Controllo granulare : reperire file

È possibile ricevere informazioni aggiuntive sui file in storage, e sulle cartelle presenti sul disco richiesto

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

$peso = Storage::size('avatar.jpg');

$time = Storage::lastModified('avatar.jpg');
```

È possibile elencare tutti i file in una cartella o in tutte le cartelle, e poi elencare ricorsivamente

```
$files = Storage::files($directory); // solo lista file

$files = Storage::allFiles($directory); // lista file e sottocartella

$directories = Storage::directories($directory); // solo lista cartelle

$directories = Storage::allDirectories($directory); // lista cartelle e sottocartelle
```