

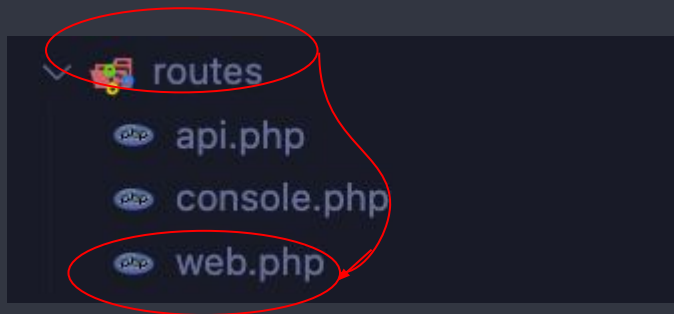
# Modulo Laravel

## Routes

# Le Routes

Laravel implementa un fantastico sistema di routing, di facile utilizzo e comprensione. Non importa quanti percorsi avrai o vorrai creare, tutto è gestito dal file:


**/routes/web.php**



```
<?php  
  
use Illuminate\Support\Facades\Route;  
  
Route::get('/', function () {  
    return view('welcome');  
});
```

# Le Routes

Per definire una routes, la sintassi generale sarà:



```
Route::facades ($uri, $callback);
```

**Route:** è la classe principale;

**Facades:** è il metodo statico utilizzato per accedere alla risorsa (GET, POST, PUT, VIEW ecc..);

**\$uri:** l'indirizzo richiamato nel browser;

**\$callback:** la chiamata da associare alla route (Un controller, una funzione ecc..).

# Routes e Funzioni

Appena configurato troverai la seguente route abilitata:



```
Route::get('/', function () {  
    return view('welcome');  
});
```


# Pagina 404

Per gestire una pagina 404 occorre creare il file views in:

**`resources/views/errors/404.blade.php`**

# Le Routes - Parametri

Nulla ti vieterà di rendere le tue rotte dinamiche.  
Potrai inserire tutti i parametri che desideri:



```
Route::get('/profile/{id}', function ($parametro_formale_id) {  
    return $parametro_formale_id;  
});
```

# Routes name

Una buona abitudine è quella di assegnare sempre un nome ad ogni Route creata. Oltre che rispettare il principio di integrità all'interno del progetto (*trattando la route come una costante con un nome univoco, potrai cambiare tranquillamente l'URI in web.php senza problemi*) renderà il tuo progetto più chiaro e pulito.

Ad esempio, una route del tipo:

```
Route::get('/', function () {  
    return view('welcome');  
});
```

Potrà essere richiamata utilizzando semplicemente **homepage**.

```
Route::get('/', function () {  
    return view('welcome');  
})->name('homepage');
```

# Richiamare routes name

```
<a href="{{route('homepage')}}"> Link </a>
```

```
<a href="{{route('profile', ['id' => 4])}}"> Link </a>
```



# Route List

Con il comando `artisan route:list` potrai avere sempre il polso della situazione delle route attive sul tuo progetto.



```
php artisan route:list
```