

## SELEZIONE LINGUAGGIO

Scarichiamo e **installiamo** la libreria frontend per visualizzare le bandierine di selezione.

Link - <https://www.npmjs.com/package/flag-icon-css>

- Nella *bash*:

```
npm install flag-icon-css --save-dev
```

**--save-dev** . Lo utilizziamo per avere il collegamento al Json in modo da poterlo far installare anche ad altri colleghi che clonano il nostro progetto tramite *npm install*

- In *app.scss* importiamo la libreria appena installata:

```
@import '~flag-icon-css/sass/flag-icon';
```

~ . Home del *node\_modules*

**flag-icon.scss** . Nella *bash* con *ls node\_modules/flag-icon-css/* troviamo il file di accesso alla libreria.

- Facciamo il *building degli assets* appena importati:

```
npm install
npm run prod
npm run dev
```

Abbiamo inserito in *public/app.css* tutti i riferimenti alle bandiere, e in *images* anche gli Svg delle stesse.

- Inserisco le *flag-icon it/gb/es* nella *navbar*:

```
<li class="nav-item">
  <form action="{route('locale', 'en')}}" method="POST">
    @csrf
    <button class="nav-link" type="submit"
      style='background-color:transparent; border:none;'>
      <span class="flag-icon flag-icon-it"></span>
    </button>
  </form>
</li>
```

Inseriamo la rotta relativa al **Locale** ovvero la sessione contenente il linguaggio desiderato:

```
Route::post('/locale/{locale}', [HomeController::class, 'locale'])->name('locale');
```

- Nel *Controller*:

```
public function locale($locale){
    session()->put('locale', $locale);
    return redirect()->back();
}
```

**session()** . Indichiamo che ogni sessione utente utilizzerà il *\$locale* (la lingua) scelta.

- Creiamo un nuovo *Middleware*:

```
php artisan make:middleware SetLocaleMiddleware
```

In *app\Http\Middleware\SetLocaleMiddleware.php*:

```
class SetLocaleMiddleware{
    public function handle($request, Closure $next){

        $locale = session('locale', 'it');
        App::setLocale($locale);

        return $next($request);
    }
}
```

*Se nella request e' specificato un locale, setta la sessione (**setLocale**) nella lingua scelta, altrimenti nella variabile specifichiamo l'italiano come linguaggio base.*

- In *app\Http\Kernel.php\Kernel* inseriamo il *Middleware* che scattera' per ogni rotta del file *web.php*:

```
protected $middlewareGroups = [
    'web' => [
        ...
        \App\Http\Middleware\SetLocaleMiddleware::class
    ],
]
```

- Creiamo il file *ui.php* in ogni cartella della lingua corrispondente (*en*, *it*, *es*) e in ognuno di essi inseriamo la traduzione degli elementi per ogni testo statico di ogni *view*, mantenendo gli stessi key per tutte le lingue :

```
<?php

return [
    'welcome' => 'Welcome on Presto.it',
];
```

- Nella view modifichiamo in questo modo la frase corrispondente assegnando welcome:

```
<h1>{{ __( 'ui.welcome' ) }}</h1>
```

Nell'*AppServiceProvider* aggiungo un *if* per verificare che la tabella che richiamiamo esista prima di richiamarla. Questo torna utile in caso di rollback o refresh del database:

```
public function boot(){
    if(Schema::hasTable('categories')){
        $categories = Category::all();
        View::share(compact('categories', 'articles'));
    }
}
```