url и переключение перевода сайта

D:\ystanovki\OpenServer\_5.3.8\domains\refugee\config\app.php

'locale' => 'en',  
'alternative\_lang' => ['uk','ru'],  
'all\_lang' => ['en','uk','ru'],

'providers' => [  
 // ссылка на фасад префикса языка url и смена языка сайта  
 \App\Providers\LocalizationProvider::class,  
]

'aliases' => [  
 // фасад префикса языка url и смена языка сайта  
 'Localization' => \App\Facades\LocalizationFacades::class,  
],

D:\ystanovki\OpenServer\_5.3.8\domains\refugee\routes\web.php

Route::group([  
 // переключение url и translation сайта  
 'prefix' => Localization::*locale*(),  
], function () {  
 Route::get('/', 'IndexController@index')->name('index');  
});

D:\ystanovki\OpenServer\_5.3.8\domains\refugee\app\Facades\LocalizationFacades.php

class LocalizationFacades extends Facade  
{  
 // обратиться к ячейке provider  
 public static function getFacadeAccessor()  
 {  
 return "Localization";  
 }  
}

D:\ystanovki\OpenServer\_5.3.8\domains\refugee\app\Providers\LocalizationProvider.php

class LocalizationProvider extends ServiceProvider  
{  
 // создать имя фасада и ссылку на service  
 public function register()  
 {  
 $this->app->bind("Localization", 'App\Services\LocalizationService');  
 }  
}

D:\ystanovki\OpenServer\_5.3.8\domains\refugee\app\Services\LocalizationService.php

class LocalizationService  
{  
 // динамический префикс языка и назначение translation сайта  
 public function locale()  
 {  
 // префикс альтернативного языка присутствует  
 $locale = request()->segment(1, '');  
 if($locale && in\_array($locale, config("app.alternative\_lang"))){  
 App::setLocale($locale);  
 return $locale;  
 }  
 // язык указан по умолчанию - назначить язык по умолчанию  
 elseif( $locale && $locale == config("app.locale") ) {  
 App::setLocale(config("app.locale"));  
  
 // редирект без префикса  
 $segments = request()->segments();  
 $segments[0] = '';  
 $string = implode("/", $segments);  
 return redirect($string)->send();  
 }  
  
 // все что не папало в условие App::getLocale() отдает по умолчанию (en)  
  
 return "";  
 }  
  
}

меню языков и изменение url для языка

D:\ystanovki\OpenServer\_5.3.8\domains\refugee\config\app.php

'providers' => [  
 // данные по умолчанию для фронт части  
 \App\Providers\DefaultValueProvider::class,

]

D:\ystanovki\OpenServer\_5.3.8\domains\refugee\app\Providers\ComposerServiceProvider.php

class DefaultValueProvider extends ServiceProvider

{  
 public function register() { }  
  
 public function boot()  
 {  
 View::composer('\*', function ($view) {  
 // языковое меню  
 $lang = (new LanguageService())->getLanguageArray();  
 $user = Auth::user();  
  
 $view->with(compact('lang','user'));  
 });  
 }  
}

D:\ystanovki\OpenServer\_5.3.8\domains\refugee\app\Services\LanguageService.php

class LanguageService  
{  
 */\*\*  
 \* выбрать флаги и языки для менюхи  
 \** ***@return*** *array  
 \*/* public function getLanguageArray() {  
 $lang = [];  
 $array\_land = config('site.locale.languages'); // мой масив языков  
 $lang\_local = App::getLocale();  
 $avatar = $array\_land[$lang\_local][3]['avatar'];  
 // для front "`${lang.prefix\_lang} more url `"  
 $prefix\_lang = in\_array($lang\_local, config("app.alternative\_lang")) ?  
 "/$lang\_local/" :  
 '/';  
 // для back-end - redirect(session('prefix\_lang'));  
 session(['prefix\_lang' => $prefix\_lang]);  
  
 // создать масив языков для клиентской части  
 foreach ($array\_land as $array){  
 $lang[] = (object) $array[3];  
 }  
  
 return compact( 'avatar', 'lang', 'lang\_local', 'prefix\_lang');  
 }  
}

D:\ystanovki\OpenServer\_5.3.8\domains\refugee\resources\views\layouts\app.blade.php

<top-menu-component  
 :lang="{{json\_encode($lang)}}"  
></top-menu-component>

D:\ystanovki\OpenServer\_5.3.8\domains\refugee\resources\js\components\menu\TopMenuComponent.vue

<ul class="navbar-nav ml-auto">  
 <!-- Language menu -->  
 <li class="nav-item dropdown button-menu">  
 <a class="nav-link" data-toggle="dropdown" href="#">  
 <img :src="this.lang.avatar" class="lang\_flag" alt="flag language">  
 </a>  
 <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">  
 <div v-for="***item*** in this.lang\_sort" :key="***item***.title">  
 <a class="dropdown-item"  
 :href="`${lang.prefix\_lang}/language/${***item***.alias}`"  
 >  
 <img :src="***item***.avatar" class="lang\_flag" alt="flag language">  
 {{***item***.title}}  
 </a>  
 <div class="dropdown-divider"></div>  
 </div>  
 </div>  
 </li>  
</ul>

<script>  
 export default {  
 props: [  
 'lang', // масив названий и url языка  
 ],  
 }  
</script>

lang.prefix\_lang – содержится язык если он не основной для сайта

:href="`${lang.prefix\_lang}/language/${***item***.alias}`"

Для Blade шаблонов подстановка url / автоматически подставляет префикс языка

<a href="{{ route('index') }}">111</a>