Приложение было создано быстрее и с меньшим количеством проблем благодаря наличию большего количества обучающего материала, такого, как темы на форуме и разделы официальной документации.

Unreal Engine более требователен к ресурсам устройства, но взамен предлагает более высокое качество графики, особенно это заметно в работе со светом на примере отрисовки тени. Также, приложение, созданное с помощью Unreal Engine, запускается быстрее. Проблемы со стереоскопическим эффектом, выявленные при тестировании, могут быть объяснены несовместимостью тестового устройства и движка, так как в VR-симуляции на ПК таких проблем не возникает.

Заключение

В данной статье были рассмотрены инструментарии для VR-разработки и произведен сравнительный анализ движков Unity и Unreal Engine для разработки VR-приложений для мобильных платформ.

Результаты теоретических и практических исследований показывают, что движок Unity хорошо подходит разработчикам, имеющим небольшой опыт в создании 3d-приложений, которые смогут с его помощью быстрее и проще создать проект, благодаря более простому процессу разработки и большому количеству готовых решений. В то же время, продвинутые возможности работы со светом и виртуальным окружением Unreal Engine оценят более опытные разработчики.

Приведенные в статье результаты исследований, описания и результаты тестов должны помочь читателю сформировать мнение относительно приоритетности использования рассмотренных инструментов в их конкретных реализациях.

***

1. Разработка приложений виртуальной реальности // Umbrella IT URL: <https://umbrellait.com/ru/blog/grown-out-of-playground-vr-is-expanding-boundaries/> (дата обращения: 25.12.2020).
2. Best virtual reality SDKs to build VR apps // ThinkMobiles URL: <https://thinkmobiles.com/blog/best-vr-sdk/> (дата обращения: 25.12.2020).
3. Орёл или решка: сравнение Unity и Unreal Engine // DTF URL: <https://dtf.ru/gamedev/7227-orel-ili-reska-sravnenie-unity-i-unreal-engine> (дата обращения: 25.12.2020).
4. Unity vs Unreal Engine for XR Development: Which One Is Better? // Circuit Stream URL: <https://circuitstream.com/blog/unity-vs-unreal/> (дата обращения: 25.12.2020).
5. Джонатан Линовес. Виртуальная реальность в Unity. / Пер. с англ. Рагимов Р. Н. – М.: ДМК Пресс, 2016. – 316 с.
6. Макеффри М. Unreal Engine VR для разработчиков. / Пер. с англ. Н. И. Веселко – М.: Эксмо, 2019. – 256 с.
7. Google VR Quick Start // Unreal Engine URL: <https://docs.unrealengine.com/en-US/ShippingAndReleasing/XRDevelopment/VR/GoogleVR/QuickStart/index.html> (дата обращения: 25.12.2020).
8. Quickstart for Google VR SDK for Unity with Android // Google VR URL: <https://developers.google.com/vr/develop/unity/get-started-android?hl=ru> (дата обращения: 25.12.2020).

Бадокина Т.Е., Полетаев А.Л.

Разработка мультязычности на фреймворке laravel

ФГБОУ ВО «МГУ им. Н. П. Огарёва»

(Россия, Саранск)

doi: 10.18411/lj-04-2021-02

Аннотация

В статье приведено краткое описание особенностей разработки мультязычности на laravel, реализующие переключение языков в CRM-системе.

Ключевые слова: мультязычность, Laravel, php, MySQL, SQL-запросы.

Abstract

The article provides a brief description of the features of multilingual development in laravel, which implement language switching in a CRM-system.

Keywords: Laravel, multilingual, php, MySQL, SQL-query.

В современном мире для успешного развития компании необходимо эффективное времяпровождение сотрудников компании на рабочем месте. Когда в компании три-пять человек все просто. Потому что все, как правило, сидят в одном офисе и постоянно общаются напрямую, а руководитель непосредственно знает, чем занят любой сотрудник в конкретный момент времени. Но когда штат компании увеличивается хотя бы до двадцати человек, все меняется и руководителю довольно сложно запомнить, какая задача стоит перед каждым сотрудником. Для того, чтобы избежать данной проблемы существуют системы контроля и управления доступом и CRM-системы для учета рабочего времени сотрудников. Повсеместная глобализация привела к тому, что во многих ИТ-компаниях работают сотрудники, разговаривающие на разных языках, поэтому вопрос реализации мультиязычности в CRM-система для учета рабочего времени сотрудников является весьма актуальной задачей.

Объектом исследования является CRM-система Time-CRM, предназначенная для учета рабочего времени сотрудников. Именно на ее примере будет изучаться возможность реализации мультиязычности на фреймворке laravel. Данная система имеет следующие возможности:

- получить общую для компании стандартизированную базу контактов;
- эффективно осуществлять контроль качества работы сотрудников в любой момент времени;
- получить статистику и аналитику эффективности работы;
- исключить потерю информацию о затраченном времени, заказах и клиентах;
- планировать повышение качества работы и разрабатывать стратегию ведения бизнеса.

На текущем моменте развития CRM-система представлена как web-приложение. В будущем данная система будет реализована под мобильное приложение Android и IOS.

Мультиязычная CRM-система – это система, которая адаптирована под носителей разных языков. Наличие качественно переведенной версии системы позволяет рассказать о своих преимуществах, технологиях и услугах широкой аудитории.

Мультиязычность на фреймворке Laravel устроена таким образом, что можно добавить новый язык без каких-либо затруднений и нарушений функционала.

При первоначальной настройке будут реализованы три языка: русский, английский и французский.

1 Таблица языков БД

Для реализации переключения языков необходимо создать таблицу БД, в которой будет храниться название и иконки всех языков системы. Миграция таблицы представлена на рисунке 1. Данная таблица состоит из пяти колонок:

- id – автоинкремент, данный столбец будет использоваться как внешний ключ в других таблицах;
- name – тип string, наименования языка;
- icon – тип string, ссылка на изображение иконки языка;
- created_at – тип timestamp, дата добавления записи;
- updated_at – тип timestamp, дата изменения записи;
- deleted_at – тип timestamp, дата удаления записи.

Также в каждой таблице БД будет присутствовать колонка `language_id`, указывающая на каком языке хранится данная запись.

```
/**
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
    Schema::create('languages', function (Blueprint $table) {
        $table->increments('id');
        $table->string('icon');
        $table->string('name');
        $table->timestamps();
        $table->softDeletes();
    });
}
```

Рисунок 1 – Миграция таблицы *languages*

В файле конфигурации `config/app.php` необходимо установить язык по умолчанию. Изначально в системе будет установлен русский язык.

2. Создание словарей

Словари в Laravel представляют собой просто ассоциативный массив `php`. В качестве ключа выступает наименование перевода, а значение – перевод. Для каждого языка, используемого в системе, необходим собственный набор словарей. В корне проекта `resources/lang` создается папка с сокращенным названием языка, а внутри нее создаются файлы с переводом конкретной страницы. На рисунке 2 представлены наборы словарей для языков.

Существует список основных словарей, которые должны обязательно существовать для каждого языка, используемого в системе. К таким словарям относятся:

- `auth` – словарь для страницы авторизации, регистрации и восстановления пароля;
- `errors` – словарь основных ошибок;
- `validation` – словарь ошибок валидации и перевод основных атрибутов.

Данные словари используются для системных ошибок и страниц, которые поставляются вместе с установкой фреймворка.

Рисунок 2 – Список словарей с переводами

3. Создание Middleware

По умолчанию Laravel всегда загружается с языком, выбранным в конфигурации. Чтобы изменить его, необходимо перехватить запрос до того, как тот получит `view` и установить язык методом `App::setLocale()`. Middleware (промежуточное программное обеспечение) — это класс программного обеспечения, предназначенного для объединения компонентов распределенного клиент-серверного приложения или

целых сетевых приложений в единую информационную систему. Код представлен на рисунке 3. В данном коде будет происходить проверка языка, который выбрал пользователь и установка данного языка по умолчанию. Если же пользователь не авторизован, то язык берется из сессии или первый язык из таблицы БД.

```
public function handle($request, Closure $next)
{
    $language = Auth::check() ? Auth::user()->language_value_id : Session::get('language');
    if(empty($language))
        $language = Language::first();
    else
        $language = Language::find($language);
    App::setLocale($language->name);
    return $next($request);
}
```

Рисунок 3 – Middleware для проверки языка

4. Создание route для переключения языка

Регистрируем путь, по которому будем переключать язык в сессии пользователя и перезаписывать значение БД. Для этого выбрана ссылка вида /language/{language_id}, в качестве параметра language_id выступает id значение из БД. При переходе по данной ссылке происходит поиск конкретного языка по id из get параметра language_id. При нахождении записи по id, данный язык записывается в сессию, а если пользователь авторизован, то присваивается выбранный язык конкретному пользователю. Если же язык не найден, то устанавливается первый язык из таблицы БД.

Код изменения языка представлен на рисунке 4.

```
<?php
|
namespace App\Services;
use ...
class LanguageService
{
    public function editLanguage($language)
    {
        $language = Language::find($language);
        if(empty($language))
            $language = Language::first();
        Session::put('language', $language->id);
        if(Auth::check()){
            Auth::user()->language_value_id = $language->id;
            Auth::user()->save();
        }
        return true;
    }
}
```

Рисунок 4 – Код изменения языка

По завершению настройки мультиязычности, каждая страница системы доступна на трех языках: русский, английский и французский.

Таким образом, в статье были рассмотрены особенности разработки мультиязычности на фреймворке laravel, реализующие переключение языков в CRM-системе. Была поставлена проблема и для её решения был реализован функционал на основе базы данных MySQL, словарей с переводами страниц и промежуточного программного обеспечения – middleware. За счет этого система адаптирована под носителей разных языков.

***

1. Документация Laravel [Электронный ресурс]. URL: <https://laravel.com/>.
2. Локализация Laravel [Электронный ресурс]. URL: <https://laravel.su/docs/6.x/localization>.
3. Middleware Laravel [Электронный ресурс]. URL: <https://laravel.com/docs/8.x/middleware>.
4. Создание миграций Laravel [Электронный ресурс]. URL: <https://laravel.com/docs/8.x/migrations>.