

# РАМИЛЬ АЛИЯКБЕРОВ

Full stack web-разработчик



Москва, Россия

[r@me0.biz](mailto:r@me0.biz)

+7 925 394-31-50



[www.rame0.ru](http://www.rame0.ru)

[github.com/rame0](https://github.com/rame0)

[StackOverflow](#)

Language: [RU](#) / [EN](#)

## ПРОЕКТЫ

На подавляющем большинстве проектов использовались: PHP (от 5.1 до 7.4), JQuery, MySQL.

### 1 Разработка более 200 сайтов

За все время работы в веб-студии, фрилансером и во время работы в SEO, мной и с моим участием, было создано более 200 вебсайтов разного типа. От визиток до интернет-магазинов. Более 90% проектов я делал один, если не считать дизайн. Изредка попадались интересные проекты вроде:

- разработки тестирования сотрудников (PHP),
- разработка MVP конструктора лендингов,
- разработки софта для опроса и последующей разработки рекомендаций для бизнеса (C#),
- разработки одностраничника с большим количеством меняющейся анимации при скролле (JavaScript)
- разработка системы для управления складами и оптовыми продажами (PHP, JS)

В подавляющем большинстве случаев, сайты разрабатывались на внутренней CMS, созданной для простых сайтов еще в 2005 на PHP 5.1. Она использовалась вплоть до 2015 года. Попытки ее доработки не прекращались до полного отказа. Она была не гибкая, но очень простая и быстрая в разработке, поэтому её оставляли для чего-то что нужно сделать "вчера". По сути, вся CMS была одним файлом (если не считать файлы с логикой авторизации, работы с БД, и рендера самого сайта) с большим количеством \*if'ов\* и кусков \*HTML\*. На ней же разработана система управления складами и продажами для поставщика ковров, на которую я частично перешел работать в 2016 году.

- Большим плюсом и, в то же время минусом, системы было то, что всё (кроме главного файла) можно было изменять непосредственно в самой системе, включая таблицы в БД.
- Минусом же были сложность добавления функционала, невозможность переиспользовать код (которая частично исправлялась со временем), а так же "дырявость" и устаревшая версия языка.

### 2 Система управления складом и оптовыми продажами

Этот и последующие проекты приведены в хронологическом порядке, поскольку каждый следующий базируется на предыдущем.

#### 2.1 Версия 1

Первая версия системы была написана в 2013 году. Для её разработки была использована упомянутая ранее CMS с огромным количеством доработок. Некоторые из них:

- Была переписана система авторизации.
- Переписана сама БД и работа с ней.
- Переписан, частично, "шаблонизатор".
- Были написаны несколько видов API: для AJAX запросов, для партнеров, для маркетплейсов.

Самыми большими недостатками явились:

- Минимальная связанность частей системы.

- Невозможность добавления складов, в результате чего, для разных складов приходилось создавать отдельные копии системы.
- Устаревший код который стало, практически, невозможно поддерживать и дополнять.

В таком виде система просуществовала до 2018 года пока не было принято решение её написать с нуля.

## 2.2 Версия 2

В 2018 году удалось убедить руководство, что дальнейшая поддержка системы практически невозможна. И было принято решение начать разрабатывать новую версию с нуля.

Она уже разрабатывалась на шаблоне MVC\*. Стандартные логические слои: Данные - Логика - Представления.

Backend:

- Роутер собственного производства, похожий на FastRoute
- За работу с БД отвечает FluentPDO
- Шаблонизатор - Twig
- Кеширование шаблонов Twig делает сам, кеширование других данных реализованно на файлах
- Для работы Моделей, реализован класс Model наследуемый от Entity (общее описание структуры) и реализующий методы выбора/изменения/сохранения этой структуры запросами к БД через FluentPDO
- Для работы с наборами данных реализованы классы Collection и Dictionary (используется редко)
- Создание основных объектов, вроде App или Request, реализованно через шаблон Singleton
- Для отладки используется мой форк умершего [ChromePHP](#) (чуть более старая версия, т.к. новому логеру нужен PHP7.4)
- PHP 7.2 + MySQL (MariaDB)

Frontend:

- JQuery - JS фреймворк
- Bootstrap - CSS фреймворк
- Все CSS стили пишутся на SASS
- Сборка и минификация JS и SCSS осуществляется с помощью Gulp

Other:

- Docker используется для разворачивания DEV среды
- Composer - менеджеры пакетов PHP
- Yarn - менеджеры пакетов JS
- Реализованы Yarn скрипты для:
  - запуска build (dev/prod) и watch для сборки JS и CSS
  - установки и обновления пакетов Composer в Docker образе
  - запуска выполнения Cron задач в образе

## 2.3 Основные функции

- Управление клиентами
- Управление остатками товаров
- Удобный табличный вид оформления заказа с навигацией стрелками клавиатуры

	2x3	3x4
Ковёр 1	<input type="text"/>	<input type="text"/>
	1 шт.	5 шт.
Ковёр 2	<input type="text"/>	<input type="text"/>
	0 шт.	10 шт.

- Управление коллекциями, товарами и ценами
- Рассылки клиентам различных уведомлений
- Экспорт/импорт товаров и остатков
- API для партнеров (восстановление из v1 в процессе)
- API для маркетплейсов (только в v1)

## 3 Основной розничный сайт

В 2020 году на базу разработанного кода системы управления был перенесен основной розничный сайт компании (<https://www.kovrotexs.ru/>).

На первой версии сайта Back'ом и Front'ом занимался только я. Во время перехода на новый backend, для ускорения работы, для реализации front'a был нанят фрилансер (я подготовил для него базу и следил за выполнением).

- Frontend, по-прежнему, JQuery + Bootstrap
- Были добавлены методы кеширования каталогов, фильтров и характеристик товаров.
- В качестве поискового движка используется Manticore Search.
- Полностью переработаны методы работы с товарами и остатками.
- Управление повторяющимися задачами - Crunz (Cron)
- Реализован клиент API для получения остатков с сайтов партнеров.

## 4 Система управления транспортной компанией

Во втором квартале 2021 года началась работа над созданием системы для управления перевозками. Велись обсуждения и исследования и прикидывались основная схема работы и общий функционал.

В конце лета 2021 началась основная фаза разработки.

- Frontend - JQuery + Bootstrap
- PHP 7.4
- Было принято решение выделить предыдущие наработки в отдельный фреймворк и работать на нем
- В процессе переноса отказался от Singleton'ов
- Заменил собственный роутер на FastRoute
- Сборщик Glup заменен на Webpack (он показал себя минимум в 10 раз быстрее)
- На данный момент разработка еще ведется. В целом система готова (прием грузов и заказов, погрузка/разгрузка, управление сотрудниками/клиентами/водителями/машинами). Идет тестирование и, часто, полное изменение некоторого функционала.