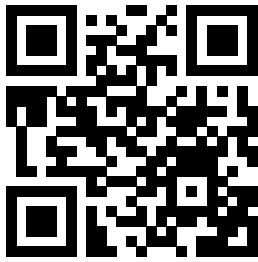


Алексей Рыльцев — Python backend developer



⬆ Middle
📍 Россия, Волгоград
💰 120 000 ₽ (руб/мес)
📅 Полная занятость | Частичная занятость
☁ Удаленная работа
✉ Контакт: geeklink.io/cv-114837



Навыки

Celery, channels, CSS, Django, Docker, Git, HTML, JS, Linux, Python, Redis, RestAPI.

Опыт работы

• ООО МТО, ООО Онлайн Дом

2018 - по настоящее время - Fullstack разработчик

Анализ требований заказчиков, проектирование модулей, написание ТЗ. Взаимодействие с техническими специалистами сторонних сервисов по части интеграций:

- интернет эквайринги (Тинькофф, Сбер, Газпром), робокасса, paykeeper
- ip телефонии (mango, redcom, telphin, rt, domru)
- расчётные центры (приём и передача показаний приборов учёта, получение начислений по ЖКХ).

Backend разработка:

- python, django 2.*, 3.*, 4.*, celery, redis,
- активное использование django ORM (постоянный контроль и оптимизация запросов к БД)
- онлайн чат в кабинетах и мобильных приложениях django-channels + rest
- REST API: django-rest-framework. django-yasg(swagger) (200+ роутов)
- вёрстка админок (html, css, jquery)
- привязка html шаблонов к django
- git
- docker
- медиа сервера для потокового вещания rtsp видеокамер (nginx + nginx-rtmp-module + ffmpeg + django для CRUD по камерам и API для управления + celery для автоматизации работы потоков - запуск и остановка по расписания, перезапуск при потере связи).
- тесты (django)
- постепенный переезд с монолита на микросервисы

Организация работы:

- проектирование системы
- постановка задач разработчикам (фронтенд и мобильные приложения) в redmine и контроль выполнение работы, доработки api по требованию
- планирование разработок (redmine + easy gant pro)
- взаимодействие с системным администратором для поддержания работы зоопарка (2 боевых сервера, 2 dev + 13 медиа с веб камерами (по 160 камер на сервер))

За время работы проекты прошли несколько стадий - от существования на одном сервере, с разными virtualenv с apache, до переносов на выделенные сервера с переездом в docker.

Основной стек технологий:

- django2.*, 3.*, 4.*
- celery - для асинхронных задач
- django-rest-framework - REST API (для фронтенда, мобильных приложений, взаимодействия с

медиасерверами)

- django-channels – взаимодействие по websocket
- xlrd, xlwt, xlswrite, dbf, dbfread – парсинг и запись DBF, XLS, XLSX
- push-notifications – отправка пуш сообщений
- pillow, opencv – работа с изображениями (чтение кадров из видеопотока, обнаружение QR кодов)
- pdf2image, qrcode, pyzbar – обнаружение и разбор QR кодов в pdf файлах

• Рекламное агенство "Дружба"

2015-2018 - WEB разработчик

Проектирование, разработка и тех поддержка web проектов компании.

Обязанности/функции:

- проектирование web систем
- вёрстка web страниц
- backend – pythondjango
- frontend – less, js(jQuery)

работа с разработчиками мобильных приложений, написание API.

• Web-студия КАК НИ КРУТИ

2014-2015 - WEB разработчик

- вёрстка шаблонов по макету
 - разработка и написания функционала
 - взаимодействие с заказчиками, выявление требований и пожеланий
- При работе использовались технологии:
- HTML,
 - CSS (+ grid960, Skeleton, animate.css),
 - jQuery – слайдеры, анимация, пользовательская логика.
 - python + JSON + jQuery – создание калькуляторов и модальных окон.
 - python Django Framework – написание ядра сайта, интерфейс администратора.
 - Joomla CMS

• ООО Пневмострой

2008-2012 - Системный администратор

Администрирование локальной сети офиса(30 ПК).

Заправка картриджей принтеров, замена фотобарабанов.

Установка и настройка ПО, ОС.

Обучение сотрудников работе с ПК и ПО.

Администрирование шлюза SQUID, файл сервера Samba.

Администрирование сайта организации.

Образование

• Прикладная информатика в экономике

ВКИ - ВФ РУК

2009-2013

сокращённо заочно

• Вычислительные машины, комплексы, системы и сети

Волгоградский энергетический колледж

2003-2007

Обо мне

<https://github.com/od-5>

На github лежат проекты, с которых начинал погружаться в разработку на django. В основном это простые сайты, для сбора и обработки заявок. Ничего интересного - много легаси, + вёрстка и jquery.

Проекты crm и distributor в плане работы уже более интересные, т.к. помимо информационной части и приёма заявок, предоставляют интерфейсы для выполнения специфических бизнес задач по части контроля и распространения объявлений, работы с рекламными агентами, расклейщиками объявлений, монтажниками информационных стендов (ведение планов продаж, постановка задач и контроль выполнения для расклейщиков и монтажников). Задачи для разработки шли большим потоком, потому много чего написано напрямую, без сервисного слоя и CBV (сейчас, глядя на api реализованное в функциональном стиле, аж плакать хочется), так же много html и jquery (по мере сил)

Заказчик, который заказывал проекты - данные бизнесы продал, в живых остался только nadomofone.ru

Наработки по этим проектам легли в основу новых проектов (с 2018 года), сначала onlinesadik.com, а затем onlinehome24.com. Сначала код в том же стиле (почти всё на функциях), затем переписывание на CBV, кастомизация базовых классов, middleware, websocket, celery... как следствие, осознание необходимости перевода web кабинетов на SPA....поиск фронтенд-разработчика в команду и вздох облегчения, что больше не надо ковырять JS и можно сосредоточиться только на python. Ведение сопроводительной документации по api для фронтендщика и разработчика мобильных приложений в google docs...осознание тщетности бытия - здравствуй django-swagger....прощай django-swagger...привет django-yasg...Нам нужен devops спец...Спец говорит надо в docker - курим все вместе, переносим всё в контейнеры, разворачиваем zabbix, mattermost, redmine, gitlab...Новые фишки требуют новых зависимостей, новые зависимости требуют python3.* - разворачиваем dev сервера, курим python release notes - за две недели переводим всё на python3. Пилим новые фишки...они требуют django 2.* - дружно курим release notes по джанге, долго гоняем тесты на dev серверах, всё таки 10k живых пользователей в базе и миллионы объектов данных - все проекты мигрируют на вторую версию за выходные. Количество веб камер переваливает за третью сотню - впиливаем мониторинг с информированием через tg бота. Растёт количество медиа серверов, отправлять к ним запросы на запуск и остановку камер по расписанию становится накладно - функционал переезжает на медиа сервера, добавляется репликация данных с проектами. Мониторинг переписывается на async. Начинает подтормаживать БД - запросы к базе вида model.objects.filter(some='bar') дополняются уточняющими select_related и prefetch_related в купе с defer и only, annotate, aggregate, Q, F - база вздохнула с облегчением. С определённой периодичностью удаётся убеждать заказчиков, что нужно тратить время на рефакторинг (получается не часто, т.к. бизнесмен не видит в этом прироста прибыли...но всё таки получается). Постепенно обрастаем сервисным уровнем. Монолит постепенно режется на отдельные небольшие сервисы.