

Валько Дмитрий Олегович

28 лет, родился 7 июня 1993

+7 (906) 565-65-67

nosefu@gmail.com

<https://t.me/atmega> — предпочтаемый способ связи

<https://github.com/NosefU>

<https://nosefu.github.io> — портфолио

Проживает: Белгород

Гражданство: Россия

Готов к переезду, готов к редким командировкам

Junior Python Backend Developer

Занятость: полная занятость

График работы: полный день

Желательное время в пути до работы: не имеет значения

Ключевые навыки

В процессе изучения Python 3 писал проекты с использованием Django, Flask, SQL (sqlite, postgresql), SQLAlchemy, requests, selenium, BeautifulSoup 4, JavaScript, Bootstrap 5.

В текущих рабочих задачах использовал geopandas, pyproj, python-docx.

Работаю под Linux, macOS.

Знаком с концепциями ООП, SOLID, моделями OSI и TCP/IP.

Опыт работы

Июль 2017 —
настоящее время
4 года 10 месяцев

ООО "Экосвязь-Центр"

Белгород, www.satellite31.ru/

Специалист по обслуживанию систем видеонаблюдения, ОПС, СКУД

Анализировал геоданные и формировал документацию в целях расстановки оборудования заказчика в соответствии с предоставленным им ТЗ. Использовал Qgis и Python. Благодаря использованию ГИС-системы и автоматизации на Python, удалось сократить время проработки проекта и разработки документации более чем в два раза.

Работаю с системами диспетчеризации городского коммунального хозяйства: от монтажа и настройки оборудования диспетчеризации до разработки и поддержки конфигураций для SCADA-систем заказчика. Благодаря разработанным мной решениям упростились и ускорилась поддержка и масштабирование систем диспетчеризации, во многих случаях для решения текущих задач достаточно квалификации штатного персонала заказчика.

Занимался монтажом и пусконаладкой систем пожарной сигнализации и дымоудаления жилых домов. На данном проекте мне удалось оптимизировать проект, уменьшив в два раза суммарную стоимость используемого оборудования, упростить монтаж, а значит уменьшить сроки монтажа в два раза, но при этом использовать более качественное и надёжное оборудование, чем то, что было заложено изначально.

Прочие задачи и обязанности:

- монтаж и техническое обслуживание различных систем видеонаблюдения,

- контроля доступа, охранной и пожарной сигнализации, в т.ч. на промышленных и режимных объектах
- поддержка и обслуживание сетевой инфраструктуры заказчиков
 - монтаж и пусконаладка систем диспетчеризации пожарной сигнализации и дымоудаления жилых домов
 - подбор и настройка оборудования для вышеназванных систем. Тесная работа с оборудованием Bolid, построение сетей на базе оборудования Ubiquiti
 - развёртывание городской IoT-сети на базе технологии LoRaWAN: монтаж, настройка и обслуживание базовых станций и клиентских устройств LoRaWAN
 - осуществление полного комплекса работ с системой видеомониторинга на транспорте: от монтажа и настройки оборудования на автомобилях заказчика до администрирования работы серверного ПО и поддержки заказчиков

Август 2016 —
Июнь 2017
11 месяцев

ООО "Варимед-Сервис"

Белгород, www.waremed.info/

Системный администратор

Разработал на Delphi с использованием библиотеки FastReport внутреннее ПО для сервисного персонала, собирающее данные с оборудования заказчика и формирующее соответствующие диагностические отчёты.

Основные задачи:

- поддержка IT-инфраструктуры офиса
- техническое обслуживание и мелкий ремонт медицинской техники
- монтаж ЛВС, телефонии
- монтаж систем контроля доступа, пожарной сигнализации и дымоудаления

Июль 2014 —
Март 2015
9 месяцев

АО "Белгородские молочные фермы", ГК "Зеленая долина"

Белгородская область

Системный администратор

Внедрил общую корпоративную и персональные адресные книги для Outlook на базе AD, что упростило работу как сотрудников (легкий поиск адреса нужного сотрудника/контрагента в общей адресной книге), так и мою (отпада нужда бэкапить файл конфига аутлука при замене/ремонте компьютера сотрудника).

Основные задачи:

- поддержка IT-инфраструктуры предприятия, производственного программного обеспечения
- поддержка пользователей
- системное и сетевое администрирование
- резервное копирование и архивирование данных
- ремонт СВТ
- монтаж и поддержка видеонаблюдения на базе оборудования Acumen и Hikvision
- монтаж и поддержка Wi-Fi сетей

Март 2013 —
Июнь 2014
1 год 4 месяца

ООО "ОНикА"

Белгород

Технический специалист

Поднял внутреннюю IP-АТС на базе Elastix/FreePBX с внешним транком, голосовым меню, записью разговоров, маршрутизацией звонков (в т.ч. и наружу, на мобильные телефоны сотрудников)

Основные задачи:

- расширение и поддержание работоспособности малых лвс (до 20 компьютеров, до 60 сетевых устройств)
- расширение и поддержание работоспособности системы видеонаблюдения на базе ПО Trassir и оборудования Ubiquiti

- настройка/обслуживание IP-АТС на базе Elastix/FreePBX;
- обслуживание офисной техники, замена/заправка картриджей
- установка и обновление типовых конфигураций 1С 8.x
- помочь пользователям в решении вопросов, связанных с работой оборудования и ПО, в том числе удаленная

Образование

Среднее

2010

Белгородский инженерный юношеский лицей-интернат

Повышение квалификации, курсы

2018

Белгородский строительный колледж

Белгородский строительный колледж, Слесарь КИПиА

Навыки

Знание языков

Русский — родной

Английский — базовые знания, на уровне чтения документации

Опыт вождения

Имеется собственный автомобиль

Права категории В

Дополнительная информация

В свободное время реализовал ряд небольших проектов, среди которых:

- Агрегатор сеансов с сайтов кинотеатров.

Разбит на два сервиса: сервис скрапинга и сервис вывода информации.

- Сервис скрапинга: скрапинг данных с сайтов кинотеатров осуществляется с помощью requests, selenium и bs4, сохраняется в бд на postgresql через sqlalchemy.

https://github.com/NosefU/cinema_aggregator_scrap

- Сервис вывода информации: Django тянет данные из БД и выводит пользователю.

Пользователь может выбирать желаемые город и дату (передаются в параметрах GET-запроса). К фронту прикручен Bootstrap.

https://github.com/NosefU/cinema_aggregator_backend

- Расширение для chrome, которое позволяет присваивать метки пользователям pikabu.ru. Серьёзный недостаток имеющихся на Пикабу заметок — их нет перед глазами. Для того, чтобы их посмотреть, приходится либо вызывать всплывающую панель профиля, либо переходить в сам профиль пользователя. Метки данного расширения всегда на виду, справа от никнейма, либо под ним. Отображаются метки и в профиле, и в постах, и в комментариях. Если метки не помещаются в строку с никнеймом пользователя, то все они переносятся на следующую. Реализовано на чистом JS + CSS. Использованы иконки Font Awesome. Метки хранятся на стороне пользователя в хранилище chrome.storage.sync.

https://github.com/NosefU/pikabu_badges

- Проигрыватель музыки на Arduino.

Форкнув проект APSG я значительно оптимизировал использование program для хранения мелодии (только одна из оптимизаций помогла уменьшить размер мелодии на 43,7%). Кроме того, на Python был написан скрипт, конвертирующий специально подготовленные MIDI-файлы в необходимый для ардуино формат. Благодаря этому значительно упростилась работа над написанием и отладкой мелодий.

<https://github.com/NosefU/ArduinoProgrammableSoundGenerator>

- Телеграм-бот, мониторящий новые коммиты в репозитории.

В момент начала моего обучения у сервиса был лишь один канал уведомлений: электронная почта. Письмо отправлялось в тот момент, когда преподаватель отчитывался о проверке задания на платформе курсов. Для меня это было неудобно и медленно, поэтому я сделал телеграм-бота, который мониторил репозиторий на наличие новых коммитов (для ускорения работы тянулись только метаданные репозитория) и отправлял метаданные новых коммитов мне в телеграм. Таким образом я узнавал о проверке работы ещё до того, как преподаватель отпишется на платформе.

<https://github.com/NosefU/telegram-commit-notifier>

- Реверс-инжиниринг камеры DJI FC40.

Для работы с камерой было доступно лишь приложение для смартфона, которое уже устарело и не работает на актуальных ОС. При помощи старого смартфона был отловлен трафик общения приложения с камерой, анализ проводился в Wireshark. Протокол не был зашифрован и представлял собой набор эндпоинтов, на которые отправлялись HTTP-запросы с командами управления. Был написан скрипт на Python, который инициализировал работу камеры, и открывал rtsp-поток для чтения любой программой, поддерживающей воспроизведение потокового видео. В процессе работы скрипт отправлял статусные сообщения, оповещавшие камеру о том, что клиент не отключился, а также выводил уровень заряда встроенного в камеру аккумулятора.

- Светомузыка для 0x40 hues.

Утилита, собранная в HiAsm, брала цвет пикселя с экрана компьютера и передавала по rs232 на Arduino. МК принимал эти данные и через полевые транзисторы выдавал этот цвет на RGB-ленту. <https://vk.cc/8ZoVor>