

all_providers

code review

1 Deploy

Деплой-скрипты присутствуют. Они почти работают, но development версия потребовала модификации.

Недостатки

- 1) Настройки для supervisor неверные, с левыми скриптами. *Исправлено*
- 2) Отсутствует шаг восстановления базы данных из дампа. На пустой машине не работает.
- 3) База данных, api, админка и клиент запускаются все на одной машине даже в production конфигурации.
- 4) Все зависимости устанавливаются скопом - даже для Python нет никакой изоляции (хотя бы virtual environment настроить).

Нужно

- 1) Наладить автоматические бэкапы базы данных, восстанавливать из последнего бэкапа.
- 2) Сделать запуск через docker.

2 Backend

- 1) Логирование кроме stdout не предусмотрено. Да есть логирование событий, но нет например логирования ошибок, какой-то debug информации.
- 2) Все запросы обернуты в catch-all try catch, при этом в JSON результате возвращается traceback. Что мягко говоря недопустимо для production приложения.
- 3) API возвращает HTTP код 200 даже в случае ошибок.
- 4) Тесты отсутствуют, их просто нет, никаких.
- 5) Там где нужно передавать параметры фильтрации/сортировки, они передаются прямо в виде операторов MongoDB, без валидации, каких либо проверок. Во-

первых это приводит к неудобному API на стороне клиента, во-вторых к потенциальным проблемам безопасности.

- 6) Просто “шикарный” по уровню разумности и эффективности сервис фоновой обработки xls. Нужно просто переписать эту часть заново.

3 Админка

- 1) Наименее качественный код во всем проекте
- 2) Катастрофа с дублированием кода. Код писался методом copy'n'paste
- 3) При огромном количестве кода интерфейс не сильно лучше, чем если бы это была автоматически-сгенерированная админка
- 4) Слишком много логики в клиенте, логика написана очень коряво
- 5) Выбор Nuxt.js с серверным рендерингом для админки неоправдан. Она работала бы лучше в качестве обычного SPA приложения
- 6) Неординарное сочетание языка coffeescript, синтаксиса шаблонов pug и Vue.js, к которому нелегко привыкнуть. (Мне самому нравились coffeescript и jade-подобные шаблоны в свое время, но это очень быстро прошло. Так как минусы перевешивают все плюсы).
- 7) Рекомендация - позже админку выкинуть и написать новую с нуля. Сейчас довести до приемлемой кондиции.

3 Клиент

- 1) Все как с админкой, но в значительно в меньшей степени
- 2) Код можно постепенно рефакторить, переписывать с нуля нет смысла.

См. следующую страницу --->

Недописанный функционал и баги

Админка (с дампами базы данных из репозитория) по факту не функционирует нормально. *Возможно с production БД, которая осталась на production сервере, она работает лучше.* **Подняли сайт лежавший на prod. сервере, проблемы нет.**

Какая-то работа началась над CRM и TV-каналами, но даже по состоянию кода понятно, что там все не очень.

Клиент работает лучше, но многое сделано очень халтурно. Не тянет на 95% готовности. Например клиент генерирует гигантское количество POST запросов. Даже при относительно малом количестве посещений, производительность API может сильно просесть. Картинки местами отдаются через сервер клиента, а не nginx.

Для генерации подсказок улиц в клиенте используется Dadata API, то есть довольно ограниченный ресурс тратится на тривиальную задачу..

Backend API тоже далековато до production качества. Кэширования нет совсем, безопасности почти нет - даже нет csrf токенов, аутентификация и авторизация - в зачаточном состоянии. Работа с MongoDB посредственная - часто сортировка и поиск делаются на Python, или даже на клиенте. Почему это не делать в самом Mongo?