

Тема:

Новостной сайт. Перенос с Drupal 7 на Nest.js + Next.js. Весь серверный функционал в Docker.

Автор: Мельников Вячеслав Юрьевич

Цели, процесс, перечень технологий:

Новостной сайт **rostof.ru** ("Ростов Официальный") реализован на платформе Drupal версии №7, серверная платформа Centos 7.

Предпосылки:

1. Изменились требования со стороны поисковых систем, возникли новые бизнес-идеи, требуется оптимизация ресурсов. В 2021 году заканчивается поддержка 7 версии Drupal.

Всё это натолкнуло меня на идею не просто перейти на новую версию Drupal, но и полностью уйти от "монолита", который не только затрудняет развитие отдельно взятого функционала,

но и часто просто блокирует возможность реализации новой идеи в силу плотной связанности отдельных блоков системы в рамках ядра Drupal.

Отмечу также, что попытки обойти ядро в целях упрощения разработки (что часто делают новички или не добросовестные разработчики),

всегда приводят к серьёзным последствиям в развитии любого Drupal - проекта (потеря безопасности, блокировка обновления с потерей функционала).

При этом стоит отметить, что переход на Drupal8 является серьёзной задачей, в отличие от дальнейшего обновления на версию 9, где реализован упрощённый update. Заканчивая тему Drupal, хочется обратить внимание на реализацию «из коробки» в версиях, начиная с 8 такой технологии: «headless (иногда ещё его называют decoupled) Drupal». В таком подходе Drupal используют как серверную часть с админкой для реализации REST. Причём сохраняется возможность использования Views. Данные через JSON получает фронтенд, реализованный на React, Angular или ещё как угодно...

Именно этот подход реализован мною через модуль на 7 версии Drupal.

В результате стала доступна выборка данных не только по отдельным нодам (статьям), но и возможность агрегирования любых данных на стороне старой системы для передачи в новую систему.

2. Для сохранения показателей SEO, возникла необходимость в серверном рендеринге (SSR). Рассматривался вариант Gatsby (SSG), но остановился на Nest.js, как более гибком и универсальном решении.

3. С целью использования одного языка для всей системы (JavaScript

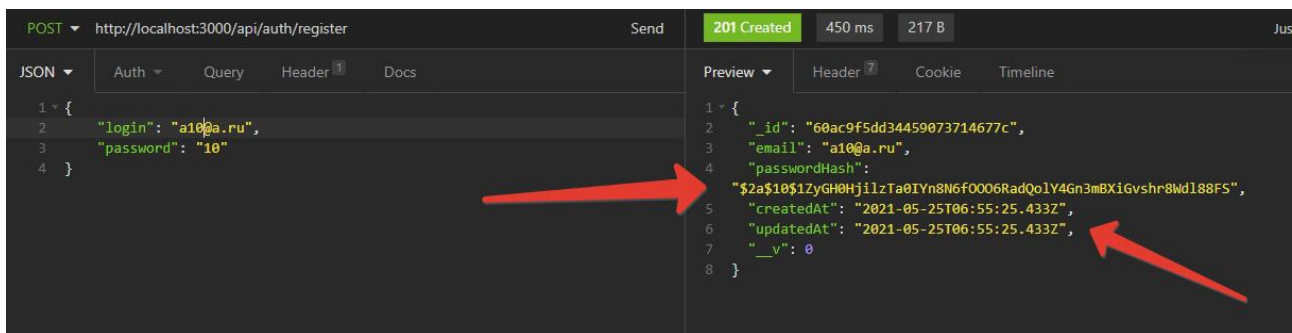
(TypeScript)), выбран стек: Nest.js + Next.js + Ionic (или React Native для мобильных приложений).

4. На сервере использование Docker даёт возможность гибкой перестройки, быстрого переноса и замены функционала. Кроме этого, индивидуальные особенности условий предоставления серверных ресурсов, склоняют к преимуществам такой мобильности.

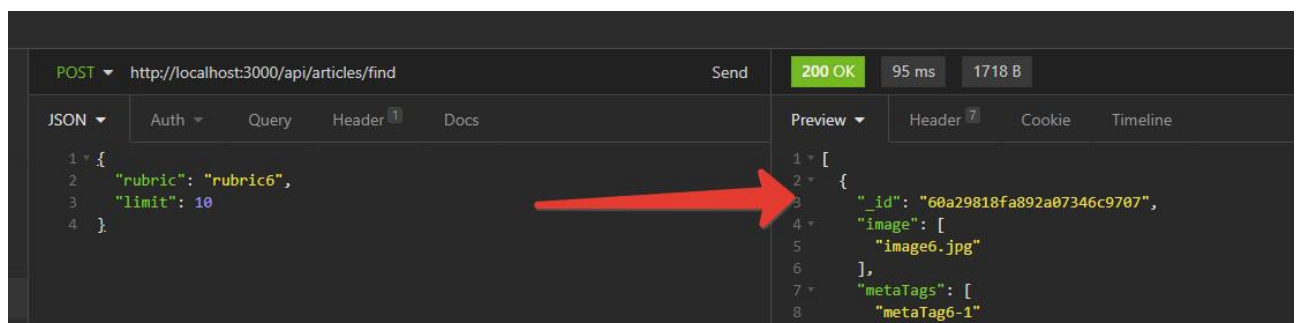
Что получилось:

1. Разработан базовый функционал серверной части на Nest.js / MongoDB

- JWT – авторизация с проверкой имеющейся в базе электронной почты при регистрации



- Создание, удаление и правка статьи по ID.



- Создание, удаление комментария к статье.
- Агрегация в MongoDB на примере объединения статьи и комментариев к статье.

```
Preview Header Cookie Timeline
22   "author": "author6",
23   "blockInfo": false,
24   "position": "position6",
25   "createdAt": "2021-05-17T16:21:44.533Z",
26   "updatedAt": "2021-05-17T16:21:44.533Z",
27   "__v": 0,
28   "comments": [
29     {
30       "_id": "60a29840fa892a07346c970a",
31       "name": "name autor comment6_3",
32       "bodyComment": "Тело комментария6_3",
33       "articleId": "60a29818fa892a07346c9707",
34       "createdAt": "2021-05-17T16:22:24.994Z",
35       "updatedAt": "2021-05-17T16:22:24.994Z",
36       "__v": 0
37     },
38     {
39       "_id": "60a29838fa892a07346c9709",
40       "name": "name autor comment6_1",
41       "bodyComment": "Тело комментария6_1",
42       "articleId": "60a29818fa892a07346c9707",
43       "createdAt": "2021-05-17T16:22:16.164Z",
44       "updatedAt": "2021-05-17T16:22:16.164Z",
45       "__v": 0
46     },
47     {
48       "_id": "60a29830fa892a07346c9708"
```

2. На живом сервере развёрнуты контейнеры с Nginx+Certbot(Let's Encrypt – сертификаты), MongoDB, Nest.js, React.

Сопроводительная информация:

https://github.com/otus-dn-core/-Any_4-1

Ссылка на репозиторий с вложенными папками:

- контейнер MongoDB
- контейнер Nginx+Certbot
- файлы Docker и Docker-compose (в корне) для запуска проекта Nest.js
- экспорт записей запросов из Insomnia

В работе:

1. Большая работа по организации централизованного обмена и систематизированного по темам архивирования изображениями, которые

использованы и используются в процессе редакторской деятельности при работе со статьями. При этом в базе должны сохраняться не только исходные (полноразмерные) изображения с метаданными, но и данные по их обрезке, размерам и соотношению сторон.

2. Кроме реализации серверной части на Nest.js, фронтенд (PWA) с SSR на Next.js, админка на Angular и в перспективе мобильные приложения на Ionic или React Native.

Идёт доработка.