

НЕКОТОРЫЕ ОСОБЕННОСТИ РАБОТЫ С СЕРВИСОМ NETLIFY

Netlify это сервис для публикации и размещения статических сайтов, с довольно большими возможностями. Здесь есть и dns-хостинг и настройки редиректов и заголовков для определенных файлов, автоматическая минификация js/css файлов и автоматическая встройка js-скриптов на страницы опубликованного сайта. Для регистрации и последующего логина можно использовать свой логин в Github/Gitlab. И для деплоя сайта есть несколько возможностей:

- использование CI Netlify при комите в репозиторий;
- ручная загрузка файлов сайта через веб-интерфейс;
- использование node-приложения для публикации сайта;
- использование API-сайта.

Самое простое – это привязать свой репозиторий к Netlify, указать в настройках результирующую директорию и команду для генерации. Все остальное будет проходить автоматически. Но тут есть ряд ограничений, как например нет возможности использования сторонних пакетов для генерации. И к примеру, включить для jekyll сайта LSI в данном варианте будет невозможно.

Сервис существует за счет привлечения денег со всех проектов, которым требуются дополнительные возможности и которые не являются Open Source. Так как мой сайт является Open Source проектом, и его код доступен всем желающим в репозитории juev.org, провел обновление своего аккаунта до уровня Open Source, и тем самым получил все возможности сервиса. Но так как Continuous Integration не поддерживает установку сторонних пакетов, пришлось отключить Related Pages после каждой статьи.

Сборка и публикация сайта теперь проходит от 20 до 30 секунд. Что более чем в 4 раза быстрее, чем при использовании Travis.

Хотелось немного поподробнее остановиться на возможности задавать редиректы и веб-заголовки файлов. Для этого используются два дополнительных файла: `_redirects` и `_headers`, в которых прописывается требуемая конфигурация. Но как показала практика, иногда этого недостаточно при создании контента. Указанные файлы должны попадать в корень директории, что используется при публикации. А в jekyll все файлы, начинающиеся с подчеркивания, не включаются в деплой. Потребовалось добавить в `_config.yml` контента дополнительный параметр: **include:** `["_redirects", "_headers"]`. Именно он

позволяет включать данные файлы без изменения имени в результирующую директорию. После добавления данной опции и заголовки и редиректы исправно заработали.

Также обратите внимание на порядок следования правил в файле заголовков. Сначала определяются общие правила, и только потом частные. В противном случае частные правила не будут отработываться.

Отметим некоторые возможности Netlify.

1. Атомное развертывание с мгновенной публикацией и откатами. В план проекта включим «развертывание» «вживую» или «переключение». Каждое успешное построение Netlify приводит к появлению недавно развернутого экземпляра вашего сайта. Публикация в узлах NetNEID CDN и такие вещи, как недействительность кэша, происходят автоматически. Развертывания являются неизменными. То есть, каждое развертывание приводит к версии сайта, которая никогда не изменяется. Последующие обновления создают новые экземпляры сайта для замены предыдущих версий. Это означает, что вы можете вернуться к любой предыдущей сборке в любое время одним нажатием кнопки в консоли администратора либо через API.

2. Уведомления и постоянные ссылки. Netlify позволяет настраивать уведомления на основе различных событий развертывания. Вы можете определить, кто получает уведомления о таких событиях, как начало развертывания или, когда успешное развертывание завершается неудачей (блокируется или разблокируется). Вы можете настроить уведомления, чтобы перейти на адреса электронной почты или на канал. Также есть возможность обеспечить функционал, чтобы эти уведомления запускали веб-чек или добавляли сообщения Git для комментирования или комментариев, что делает эти уведомления еще более полезными, так это они включают постоянную ссылку на развертывание, о котором идет речь. Это означает, что каждый может иметь собственный URL-адрес, к которому может получить доступ, чтобы увидеть это развертывание. Наличие постоянных ссылок для каждого развертывания огромно. Происходит разблокировка возможности совместного использования состояния любой версии вашей сборки с группой тестирования, клиентом. И этот мгновенный доступ доставляется из каждого уведомления.

3. Развертывания филиалов и субдомены. Очень удобно иметь возможность развертывать ветви помимо развертывания с помощью мастера. Netlify дает вам некоторый контроль над развертыванием. Вы можете выбрать, нужно ли развертывать только свою рабочую ветвь, все ветви или несколько определенных ветвей. После развертывания

каждая ветвь будет доступна на субдомене с именем, полученным от имени ветки, из которой она получена: `my-branch-name-my-site-name.netlify.com`. С управлением DNS Netlify вы также можете назначить свои собственные поддомены ветвям. Вы получаете полный контроль над тем, как разные ветви управляют контентом на разных субдоменах вашего сайта.

4. Тестирование A / B, многовариантное тестирование или раздельное тестирование. Разработчики ссылаются на тестирование A / B несколькими терминами, но Netlify называет это «разделенным тестированием» следующим образом: разбивает трафик на ваш сайт на любые части, которые вы указали. Вы можете разделить свой трафик на столько частей, сколько хотите, указав, какой процент трафика должен идти на ту или иную часть. Подход Netlify обслуживает каждый вариант непосредственно из собственного CDN. Все форматы формирования трафика, варианты взвешивания, согласованности пользователей и другие вещи происходят прямо на CDN – по краю, насколько это возможно ближе к пользователю. Каждый вариант обслуживается и представляется «как производство».

5. Команды контекстной сборки. Вы можете не только развернуть разные части сайта, но и настроить контент и среды в своих развертываниях в соответствии с различными контекстами, такими как постановка, тестирование и визуализация. Вы можете создать `staging.your-project.com` и `staging.your-project.com` или `staging.your-project.com` и еще, что вам необходимо, рядом с вашим `www.your-project.com` с дополненной конфигурацией с идентичными условиями размещения, что является важным аспектом для надежной стратегии разработки и публикации контента.

Возможно, вам захочется запустить несколько различных команд сборки в зависимости от среды, в которой вы разворачиваетесь, или для создания функции, которая еще не создана. Это можно сделать, настроив различные контексты сборки, например, сборка для сборки с помощью `gulp build:prod` и для построения ветвей функций с помощью `gulp build:my-feature`. Настраивается конфигурационным файлом, который можно сохранить в корне проекта для доступа ко всем видам опций в ваших сборках Netlify: `netlify.toml`.

6. Управление SSL и бесплатный SSL от Let's Encrypt. Хотя это может быть и не сразу очевидным, на самом деле очень важно обслуживать веб-сайты поверх HTTPS, а не просто HTTP, даже если они обслуживаются как статические сайты.

Netlify упрощает настройку HTTPS в пользовательских доменах. Существует опция управляемого SSL, пользовательского SSL и даже

выделенного IP-протокола SSL для тех предприятий, которые этого требуют. Большинство пользователей могут получить то, что им необходимо от управляемого SSL благодаря бесплатным сертификатам, предоставленным Let's Encrypt. Сертификаты обновляются автоматически.

7. Выполнение тестов с непрерывной интеграцией Netlify. Также Netlify предоставляет контейнерную среду для запуска ваших сборок в дополнение к специально созданному CDN для размещения ваших сайтов. Это означает, что любая сборка, которую вы запускаете в локальной среде сборки или на сервере непрерывной интеграции, может работать непосредственно в Netlify. Если ваша сборка включает тесты, Netlify выполнит их для вас и даст либо успешную сборку, либо неудачную, сообщив вам о результатах.

8. Обработка формы. Если вам необходима форма на вашем сайте, которая принимает контент, представленный вашими посетителями, Netlify поможет реализовать это. Просто добавив атрибут в HTML вашей формы, Netlify создаст подходящую конечную точку для формы и обработает все данные, доступные вам через интерфейс администратора и API. Благодаря контенту, доступному через API, вы можете получить доступ к нему во время сборки и использовать его в генерации вашего сайта. Представления форм также могут инициировать уведомления. Возможны уведомления в webhooks или даже интеграция Zapier.

9. Перенаправление, переписывание и проксирование. Не оставляйте URL позади `_redirects` файл `_redirects` в развернутом каталоге, вы можете получить доступ к расширенным параметрам конфигурации для перенаправления и перезаписи их. Это происходит на узлах CDN, которые делают их невероятно быстрыми и эффективными. У вас также есть возможность указать код ответа HTTP в файле `_redirects`, создавая, например, пользовательские уведомления (404 и т.п.) и даже проксирование других ресурсов.

В итоге хочется посоветовать разработчикам обратить внимание на сервис Netlify. В перспективе идет работа над новым функционалом сервиса, который тестируются энтузиастами Alpha-тестеров. В Twitter есть возможность получения обновления функций, изучив документацию или посмотрев работу над расширением экосистемы JAMstack с помощью проектов с открытым исходным кодом.