

УДК 004**АРХИТЕКТУРА SPA САЙТА ДЛЯ НАУЧНЫХ КОНФЕРЕНЦИЙ**

Войцеховский С.П., студент гр. ПИб-192, IV курс

Научный руководитель: Крутской Д.Л., старший преподаватель

Кузбасский государственный технический университет

имени Т. Ф. Горбачёва

г. Кемерово

С течением времени требуемый функционал существующих сайтов может устаревать или проседать от веса не запланировано добавленных возможностей, что ухудшает взаимодействие пользователя с этим сайтом. Решая первую проблему, мы получаем в результате вторую, которая решается сложнее: рефакторинг "джунглей" труднопроходимых частей кода, отсечение "мёртвого кода", который может сломать что-то другое, радикальное изменение интерфейса, что приводит в недоумение уже привыкших пользователей. Так или иначе, поддержка такого сайта становится поддержкой балласта. Одно из полных решений второй проблемы является пересоздание проекта с нуля, но уже учитывая эти накопившиеся внеплановые изменения, вписывая их в новое техническое задание.

Примером такого сайта можно взять "Научный портал КузГТУ", пробелы которого заметны на первой, главной, странице: бегущая строка, где можно пропустить информацию, при просмотре новостей иногда появляются пустые зоны, расширенное меню навигации может оказаться ниже картинок новостей, боковая панель повторяется трижды, оставляя за собой много пустого места, там же текст идёт поверх другого, проскальзывает английский текст. Его замена и будет целью.

В настоящее время существует несколько подходов к созданию сайтов: SSG, SSR и SPA.

Страница — HTML документ, состоящий из структурированных элементов (DOM) с информацией о используемыми им ресурсами: CSS (стили и шрифты), JavaScript (динамичный контент), контент (картинки, видео, аудио и пр.).

SSG (англ. "static site generation", букв. "создание статичных (неизменяющихся) веб-сайтов") — подход для создания сайтов без изменяющихся частей на странице: приветствия пользователя, прямой эфир изменений и т. д. Страницы хранятся на сервера, сервер отправляет эту страницу браузеру.

SSR (англ. "server side rendering", букв. "отображение на сервере") — на странице появляется динамичный контент: приветствие вошедшему пользователю, списки чего-то, панель навигации и пр. Этот динамичный контент вставляется в страницу на сервере.

SPA (англ. "single page application", букв. "одностраничное приложение") — метод работы веб-сайта, при котором при переходе на различные стра-

ницы вместо обновление самой страницы, содержание страницы динамично меняется. Это осуществляется за счёт того, что весь контент становится динамичным. Динамичный контент вставляется в браузере.

Размер получаемых ответов SSG и SSR с сервера мал. В SPA ответы куда более большие.

В новом сайте должна иметься возможность создавать конференции, отправлять доклады на участие, проверять статус заявок и другие динамичные действия. С этими требованиями отпадает SSG. С тем, что пользователи сайта могут сами добавлять и изменять контент сайта, сильно проявляется недостаток SSR в перерисовки страницы на сервера и ожидание ответа. Остаётся SPA. Насколько проблемный вес его запросов? Не очень: скорость передаваемых данных по проводу или без увеличивается с каждым годом, цены за эту скорость снижаются.

Отображение страниц при SPA:

1. Браузер запрашивает первую страницу с сайта;
2. Браузер удаляет всё содержимое прошлой страницы или создаёт новую;
3. Сервер заполняет динамичный контент шаблонизатором;
4. Сервер отвечает с информацией о странице, браузер заполняет пустую страницу;
5. Так же в браузер получает JavaScript библиотеку в которой прописывается, как он будет запрашивать данные о новой странице, удалять заменяемую часть, предоставлять полученную часть с сервера;
6. При посещении другой страницы:
 1. Браузер запрашивает страницу через JavaScript библиотеку;
 2. Сервер отвечает с частичным HTML документом, в котором имеются лишь нужные тэги;
 3. JavaScript библиотека, получив ответ, через API браузера [1] удаляет заменяемую часть страницы и вставляет полученный частичный документ, заполняет динамичный контент.

В столь динамичная натура SPA требует взаимодействия бэкенда (серверных библиотек) и фронтенда (браузерных библиотек). За бэкенд могут выступать любые библиотеки, но важно, чтобы он оставлялся главным "брокером" данных: аутентификация, авторизация, маршрутизация, какие HTTP статус коды отправлять в каких ситуациях, бизнес-логика и пр.

Ввиду того, что прошлый сайт написан на PHP, ограничимся библиотеками, написанными на этом языке. Есть несколько кандидатов: Symfony, Laravel, CakePHP, Laminas. Все прекрасно подходят, но выбор у нас зависит и от того, насколько хорошо они работают с библиотекой фронтенда.

Для фронтенда есть несколько кандидатов: Angular, Vue, React, Svelte. Все используют технологию веб-компонентов [2] для работы, которая позволяет разделить страницу на функциональные и структурные составляющие: заголовок, меню навигации, меню редактирования, список чего-либо. Контент страниц редактируется и создаётся пользователями, поэтому отпадает React с его философией неизменности страницы после создания.

Из оставшихся Symfony, CakePHP и Laminas работают как SSR-библиотеки, у Laravel же есть InertiaJS – мост между Vue или Svelte и собой, что упрощает

написание SPA. Vue и Svelte сильно не отличаются, но Vue более популярен с большим сообщество дополнительных библиотек. В итоге: Laravel и Vue.

Часть результата данной работы представлен на рисунке 2.

[Главная](#) | [Зарегистрироваться](#) | [Войти](#)

Конференции

Конференция
2023-04-13
Причём заявок до: 2023-03-27
Участников: 10/47
Конференция
2023-04-13
Причём заявок до: 2023-03-28
Участников: 0/22
Конференция
2023-04-25
Причём заявок до: 2023-04-10
Участников: 5/6
Конференция
2023-05-05
Причём заявок до: 2023-04-21
Участников: 20/43
Конференция
2023-05-31
Причём заявок до: 2023-05-14
Участников: 26/48

[≤ 1](#) ... [3](#) [4](#) [5](#) ... [13](#) [≥](#)

Рисунок 2 — главная (лэндинговая) страница веб-сайта

Список литературы:

1. Документация DOM API браузеров [Электронный ресурс] - Режим доступа: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model
2. Документация создания и использования веб-компонентов [Электронный ресурс] – Режим доступа: https://developer.mozilla.org/en-US/docs/Web/Web_Components