

РАЗРАБОТКА ВЕБ-САЙТОВ\*, ПОИСКОВАЯ ОПТИМИЗАЦИЯ, JAVASCRIPT\*, БЛОГ  
КОМПАНИИ RUVDS.COM

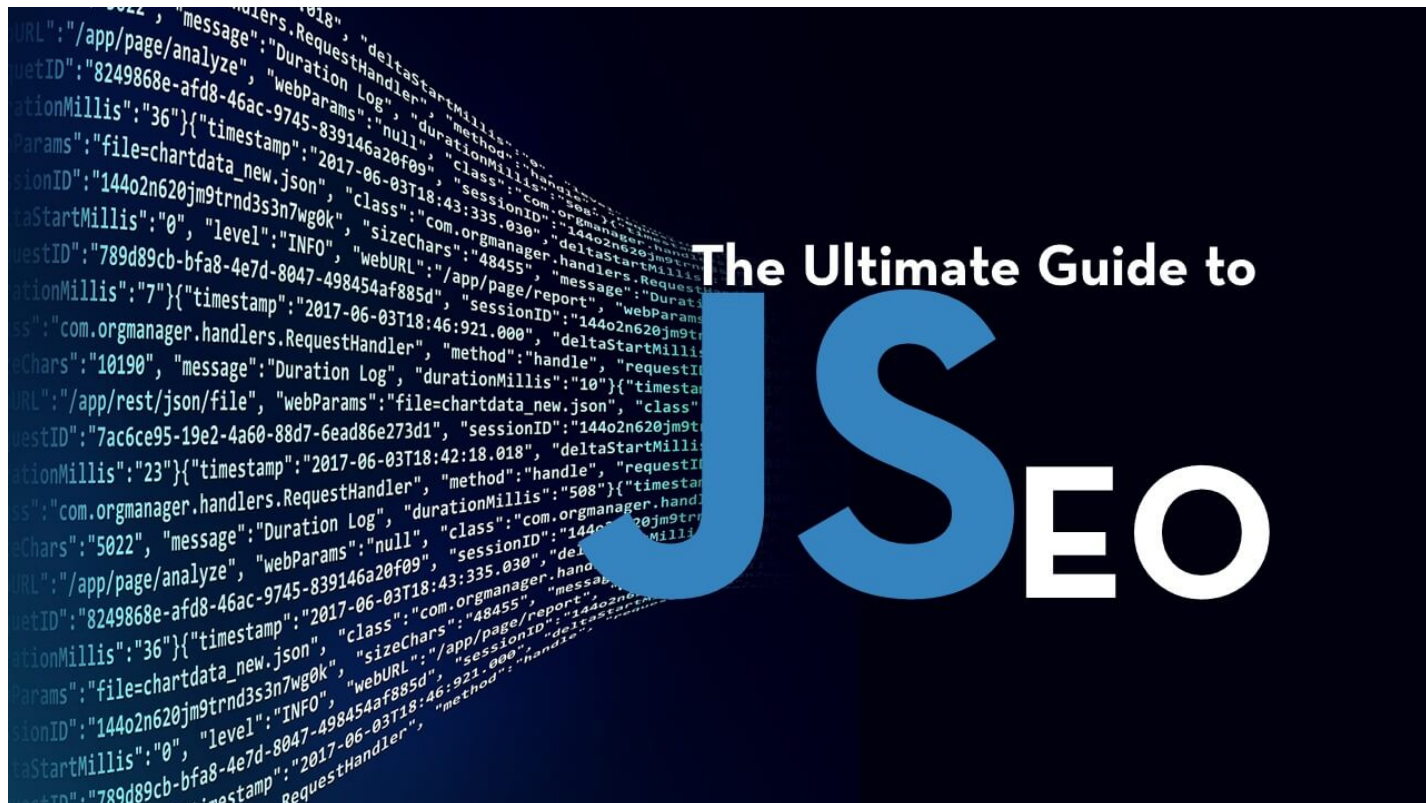
## Руководство по SEO JavaScript-сайтов. Часть 2. Проблемы, эксперименты и рекомендации

ПЕРЕВОД

ru\_vds вчера в 12:04 👁 2,3k

Оригинал: [Tomasz Rudzki](#)

В [первой части](#) перевода этой статьи мы говорили о том, как веб-мастер может взглянуть на свой ресурс глазами Google, и о том, над чем стоит поработать, если то, что увидит разработчик сайта, окажется не тем, чего он ждёт.



Сегодня, во второй части перевода, Томаш Рудски расскажет о

наиболее распространённых SEO-ошибках, которым подвержены сайты, основанные на JavaScript, обсудит последствия грядущего отказа Google от механизма AJAX-сканирования, поговорит о предварительном рендеринге и об изоморфном JavaScript, поделится результатами экспериментов по индексированию. Здесь, кроме того, он затронет тему особенностей ранжирования сайтов различных видов и предложит вспомнить о том, что помимо Google есть и другие поисковики, которым тоже приходится сталкиваться с веб-страницами, основанными на JS.

## О ресурсах, необходимых для успешного формирования страницы

Если, по вашему мнению, Googlebot вполне сможет обработать некую страницу, но при этом оказывается, что данная страница не индексируется так, как нужно, проверьте, чтобы внешние и внутренние ресурсы (JS-библиотеки, например), требуемые для формирования страницы, были доступны поисковому роботу. Если он не сможет что-то загрузить и правильно воссоздать страницу, об её индексировании можно и не говорить.

## О периодическом использовании Google Search Console

Если вы вдруг столкнулись со значительным падением рейтинга сайта, который до этого работал стабильно и надёжно, рекомендую воспользоваться описанными ранее средствами Fetch и Render для того, чтобы проверить, в состоянии ли Google правильно обработать ваш сайт.

В общем случае, полезно время от времени использовать Fetch и Render на произвольном наборе страниц сайта для того, чтобы проверить возможность его правильной обработки поисковиком.



Список ресурсов, заблокированных с помощью Robots.txt

## Опасное событие onClick

Помните о том, что Googlebot — это не настоящий пользователь, поэтому примите как должное то, что он не «щёлкает мышью» по ссылкам и кнопкам и не заполняет формы. У этого факта есть немало практических последствий:

- Если у вас имеется интернет-магазин и некие тексты скрыты под кнопками вроде «Подробнее...», но при этом они не присутствуют в DOM до щелчка по соответствующей кнопке, Google эти тексты не прочтёт. Это же имеет отношение и к ссылкам, появляющимся по такому же принципу в меню.
- Все ссылки должны содержать параметр `href`. Если вы полагаетесь только на событие `onClick`, Google такие ссылки не воспримет.

Это [подтверждает](#) и Джон Мюллер:



John Mueller

7 Jun



Hi Laurent

If the menu is in the DOM after the page has loaded, and if the menu items use "a" elements with href attributes pointing at URLs, then that should work. It would likely not work if the menu were only generated when a user interacts with the page (extreme case: click on the menu button, an AJAX call is made to fetch the menu items & render the result -- which would be slow regardless). Also, if the menu uses JS events (eg, div's with onclick handlers), then that makes it harder for us to crawl through it. One way to roughly double-check is to load the page in the browser, and use inspect-element to double-check what the menu part of the DOM looks like.

We're looking into a way of exposing the Googlebot-rendered DOM, which would help a bit, but at the moment I don't have a concrete date or even clear confirmation that I can share.

Cheers  
John

*Комментарий Джона Мюллера, подтверждающий вышесказанное*

В одном из моих предыдущих [материалов](#), который посвящён

использованию Chrome 41 для исследования сайтов, имеется краткое руководство, демонстрирующее процесс проверки меню на предмет доступности его для системы индексирования Google. Советую с этим материалом ознакомиться.

## Использование значков # в ссылках

Всё ещё распространена практика, в соответствии с которой JS-фреймворки создают URL-адреса со знаком #. Существует реальная опасность того, что Googlebot не обработает подобные ссылки.

Вот несколько примеров:

- Плохой URL: `example.com/#/crisis-center/`
- Плохой URL: `example.com#URL`
- Хороший URL: `example.com/crisis-center/`

Обратите внимание на то, что это не относится к URL с последовательностью символов # ! (так называемый hashbang).

Возможно вы решите, что всё это не имеет особого значения. Действительно — большое дело — всего один дополнительный символ в адресе. Однако, это очень важно.

Позволю себе в очередной раз процитировать Джона Мюллера:

«(...) С нашей точки зрения, если мы видим тут нечто вроде знака #, это означает, что то, что идёт за ним, возможно, неважно. По



большей части, индексируя содержимое, мы подобные вещи игнорируем (...). Когда вам нужно сделать так, чтобы это содержимое действительно появилось в поиске, важно, чтобы вы использовали ссылки, выглядящие более статично».

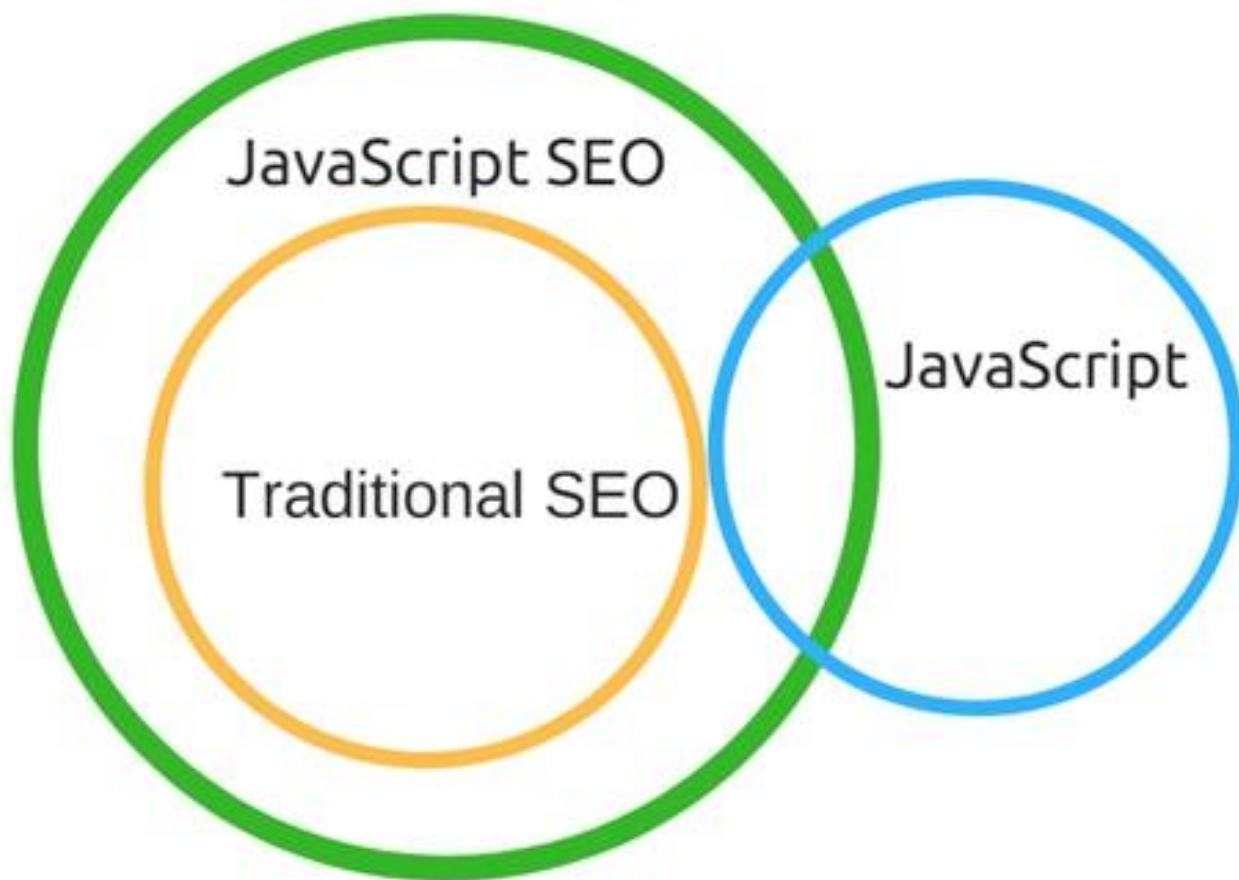
В итоге можно сказать, что веб-разработчикам стоит постараться, чтобы их ссылки не выглядели как нечто вроде `example.com/resource#dsfsd`. При использовании фреймворков, формирующих такие ссылки, стоит обратиться к их документации. Например, Angular 1 по умолчанию, использует адреса, в которых применяются знаки #. Исправить это можно, соответствующим образом **настроив** `$locationProvider`. А вот, например, Angular 2 и без дополнительных настроек использует ссылки, которые хорошо понимает Googlebot.

## О медленных скриптах и медленных API

Многие сайты, основанные на JavaScript испытывают проблемы с индексированием из-за того, что Google приходится слишком долго ждать результатов работы скриптов (имеется в виду ожидание их загрузки, разбора, выполнения). Медленные скрипты могут означать, что Googlebot быстро исчерпает бюджет сканирования вашего сайта. Добейтесь того, чтобы ваши скрипты были быстрыми, и Google не приходилось бы слишком долго ждать их загрузки. Тем, кто хочет узнать подробности об этом, рекомендую ознакомиться с **этим материалом**, посвящённым оптимизации процесса визуализации страниц.

# Плохая поисковая оптимизация и SEO с учётом особенностей сайтов, основанных на JavaScript

Хочу поднять здесь проблему, которая может повлиять даже самую лучшую поисковую оптимизацию сайтов.



*Традиционный подход к SEO и SEO с учётом особенностей JS*

Важно помнить, что SEO с учётом особенностей JavaScript выполняется на базе традиционной поисковой оптимизации. Невозможно хорошо оптимизировать сайт с учётом особенностей JS, не добившись достойной оптимизации в обычном смысле этого слова. Иногда, когда вы сталкиваетесь с проблемой SEO, вашим первым ощущением может стать то, что это связано с JS, хотя, на самом деле, проблема — в традиционной поисковой оптимизации.

Не буду вдаваться в подробности подобных ситуаций, так как это уже отлично объяснил Джастин Бриггс в своём материале «[Core Principles of SEO for JavaScript](#)», в разделе «Confusing Bad SEO with JavaScript Limitations». Рекомендую почитать эту полезную статью.

## Со второго квартала 2018 года Googlebot не будет использовать AJAX-сканирование

Компания Google сообщила о том, что со второго квартала 2018 года она больше не будет использовать AJAX-сканирование (AJAX Crawling Scheme). Значит ли это, что Google прекратит индексировать сайты, используя Ajax (асинхронный JavaScript)? Нет, это не так.

Стоит сказать о том, что AJAX-сканирование появилось в те времена, когда компания Google поняла, что всё больше и больше сайтов использует JS, но не могла правильно обрабатывать такие сайты. Для того, чтобы решить эту проблему, веб-мастерам предложили создавать особые версии страниц, предназначенные для поискового робота и не содержащие JS-скриптов. К адресам этих страниц надо было добавлять `_escaped_fragment_`.

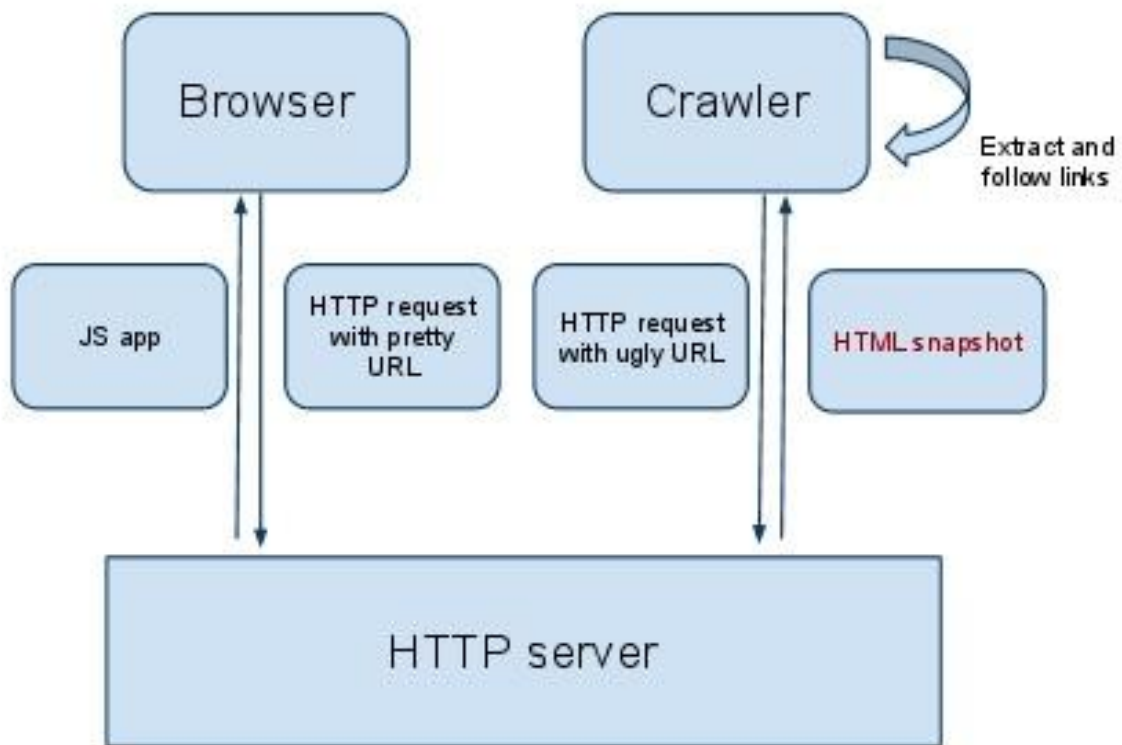
На практике это выглядит так: пользователи работают с полноценным и приятно выглядящим вариантом `example.com`, а Googlebot посещает не такой симпатичный эквивалент сайта, к которому ведёт ссылка вида `example.com?`

`_escaped_fragment_` (это, кстати сказать, всё ещё очень



популярный подход подготовки материалов для ботов).

Вот как это работает (изображение взято из блога Google).



*Разные версии сайта для пользователей и для ботов*

Благодаря такому подходу веб-мастера получили возможность убить двух зайцев одним выстрелом: и пользователи довольны, и поисковый робот счастлив. Пользователи видят версию сайта, оснащённую возможностями JavaScript, а поисковые системы могут правильно индексировать страницы этого сайта, так как им достаются лишь обычные HTML и CSS.

**О проблемах AJAX-сканирования**

Проблемы Ajax Crawling Scheme заключаются в том, что так как пользователи и поисковые роботы получают разные версии сайта, очень сложно анализировать неполадки, связанные с индексированием сайтов. Кроме того, у некоторых веб-мастеров возникали проблемы с предварительной подготовкой версий страниц, предназначенных для поискового робота.

Именно поэтому Google сообщила, что начиная со второго квартала 2018 года веб-мастерам больше не нужно создавать две разные версии веб-сайта:

The [AJAX crawling scheme](#) was introduced as a way of making JavaScript-based webpages accessible to Googlebot, and we've previously announced our [plans to turn it down](#). Over time, Google engineers have significantly improved rendering of JavaScript for Googlebot. Given these advances, in the second quarter of 2018, we'll be switching to rendering these pages on Google's side, rather than on requiring that sites do this themselves. In short, we'll no longer be using the AJAX crawling scheme.

### *Сообщение об отключении AJAX-сканирования*

Как это повлияет на веб-разработчиков?

1. Google теперь будет формировать страницы сайтов своими средствами. Это означает, что разработчикам необходимо обеспечить то, чтобы у Google была техническая возможность сделать это.
2. Googlebot прекратит посещать ссылки, содержащие `_escaped_fragment_` и начнёт запрашивать те же

материалы, которые предназначены для обычных пользователей.

Кроме того, разработчикам понадобится найти способ сделать свои сайты доступными для Bing и других поисковых систем, которые пока далеко позади Google в деле обработки сайтов, основанных на JavaScript. Среди возможных решений можно отметить серверный рендеринг (универсальный JavaScript), или, как ни странно, продолжение использования старого AJAX-сканирования для Bingbot. Об этом мы поговорим ниже.

## **Будет ли Google использовать самый современный браузер для обработки страниц?**

Сейчас неясно, планирует ли Google обновлять свою службу обработки сайтов с целью поддержки самых свежих технологий. Нам остаётся лишь надеяться на то, что это будет сделано.

## **Как быть тем, кто не хочет, чтобы Google обрабатывал его страницы, основанные на JS, а довольствовался лишь заранее подготовленными страницами?**

Я много об этом думал, и именно поэтому решил спросить Джона Мюллера на форуме, посвящённом JavaScript SEO, о том, могу ли я обнаружить Googlebot по User-Agent и просто отдать ему заранее подготовленную версию сайта.

me (Tomek Rudzki change)

10:39 (6 hours ago)



John, in Q2 2018 you will start turning down an old Ajax Crawling Scheme

"we'll be switching to rendering these pages on Google's side, rather than on requiring that sites do this themselves"

<https://webmasters.googleblog.com/2017/12/rendering-ajax-crawling-pages.html>

But since my website is strongly dependent on modern JS features (ES6+), I am afraid Googlebot will not be able to crawl my website at all.

Since I prerender for Bing, I wish I could have the possibility to prerender for Googlebot as well.

So I am thinking about such a solution:

When Googlebot or Bingbot requests /page.html -> the server detects that it's a bot and returns a pre-rendered version of the page

User requests /page.html -> the server returns the SPA version of my page.

Is such behavior considered cloaking?

This is why I am worried about it:

My website is strongly dependent on ES6. Can we be sure the Web Rendering Service will fully support ES6 in Q2 2018?

I am afraid Googlebot will only partially render my website because of the timeouts.

Since I still have many users coming from Bing, I am prerendering for Bing and it's only natural to do it for Googlebot as well.

*Вопрос Джону Мюллеру о версии сайта, подготовленной специально для Googlebot*

Вот что он ответил:



**John Mueller**

16:44 (46 minutes ago)



Hi Tomek

If you can be certain that your pre-rendering serves equivalent content at approximately a DOM level, that would be acceptable for Google. It's important to us that this version matches that which a normal user would see. On top of that, you probably want to monitor your pre-rendered version regularly, to make sure you're not accidentally serving search engines error pages (which you might not notice if you're only looking at the ES6 version of your site).

Hope that helps!

John

*Ответ Джона Мюллера*

Джон согласен с тем, что разработчик можете узнать, что его сайт сканирует Googlebot, проверяя заголовок User-Agent и отдать

ему заранее подготовленный HTML-снимок страницы. Вдобавок к этому он советует регулярно проверять снимки страниц для того, чтобы быть уверенным в том, что предварительный рендеринг страниц работает правильно.

## Не забудьте про Bing!

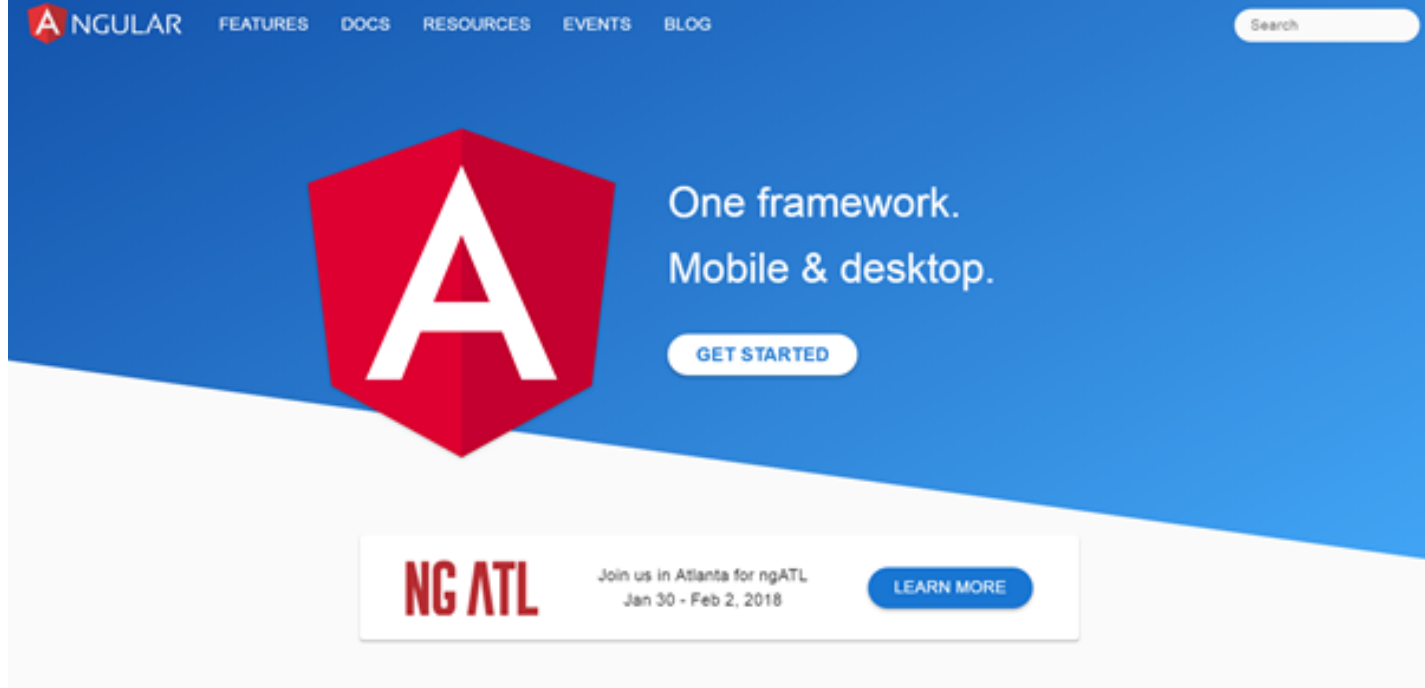
Представим себе, что Googlebot идеально обрабатывает сайты, основанные на JS-фреймворках и с ним у вас нет никаких сложностей. Значит ли это, что можно забыть обо всех проблемах, связанных с индексированием таких сайтов? К сожалению — не значит. Вспомним о поисковике Bing, который, в США, применяет около трети интернет-пользователей.

В настоящее время разумно будет считать, что Bing вообще не занимается обработкой JavaScript (ходят слухи, что Bing обрабатывает JS на страницах с высоким рейтингом, но мне не удалось найти ни одного подтверждения этим слухам).

Позвольте мне рассказать об одном интересном исследовании.

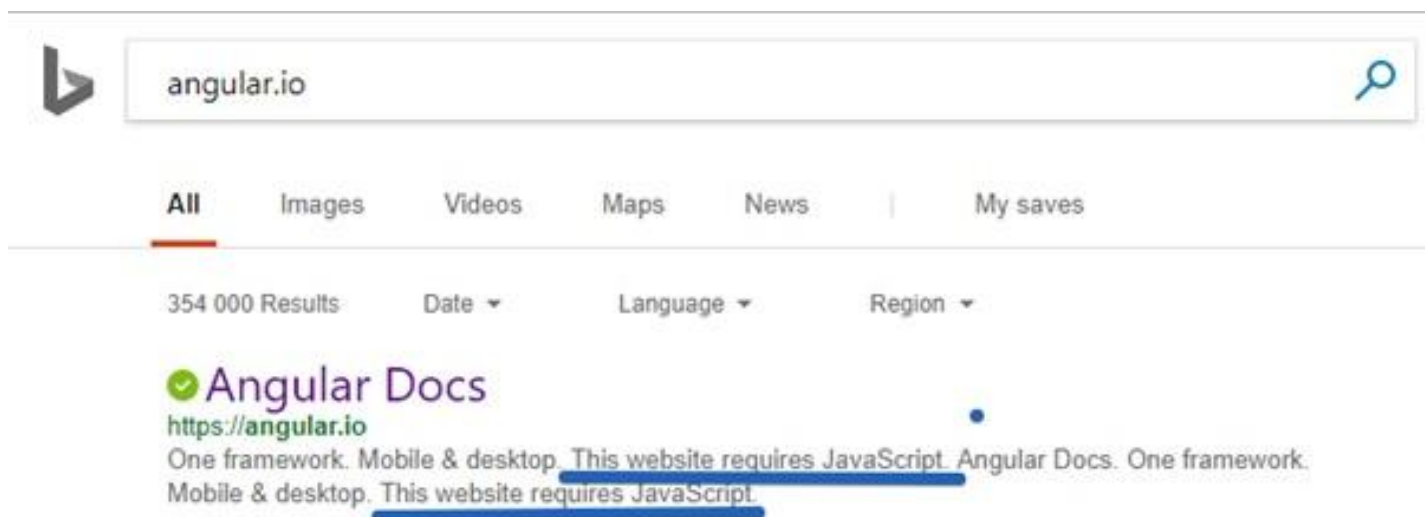
Angular.io — это официальный веб-сайт Angular 2+. Некоторые страницы этого сайта созданы в виде одностраничных приложений. Это означает, что их исходный HTML-код не содержит никакого контента. После загрузки такой страницы подгружается внешний JS-файл, средствами которого и формируется наполнение страниц.





*Сайт Angular.io*

Возникает такое ощущение, что Bing не видит содержимого этого сайта!

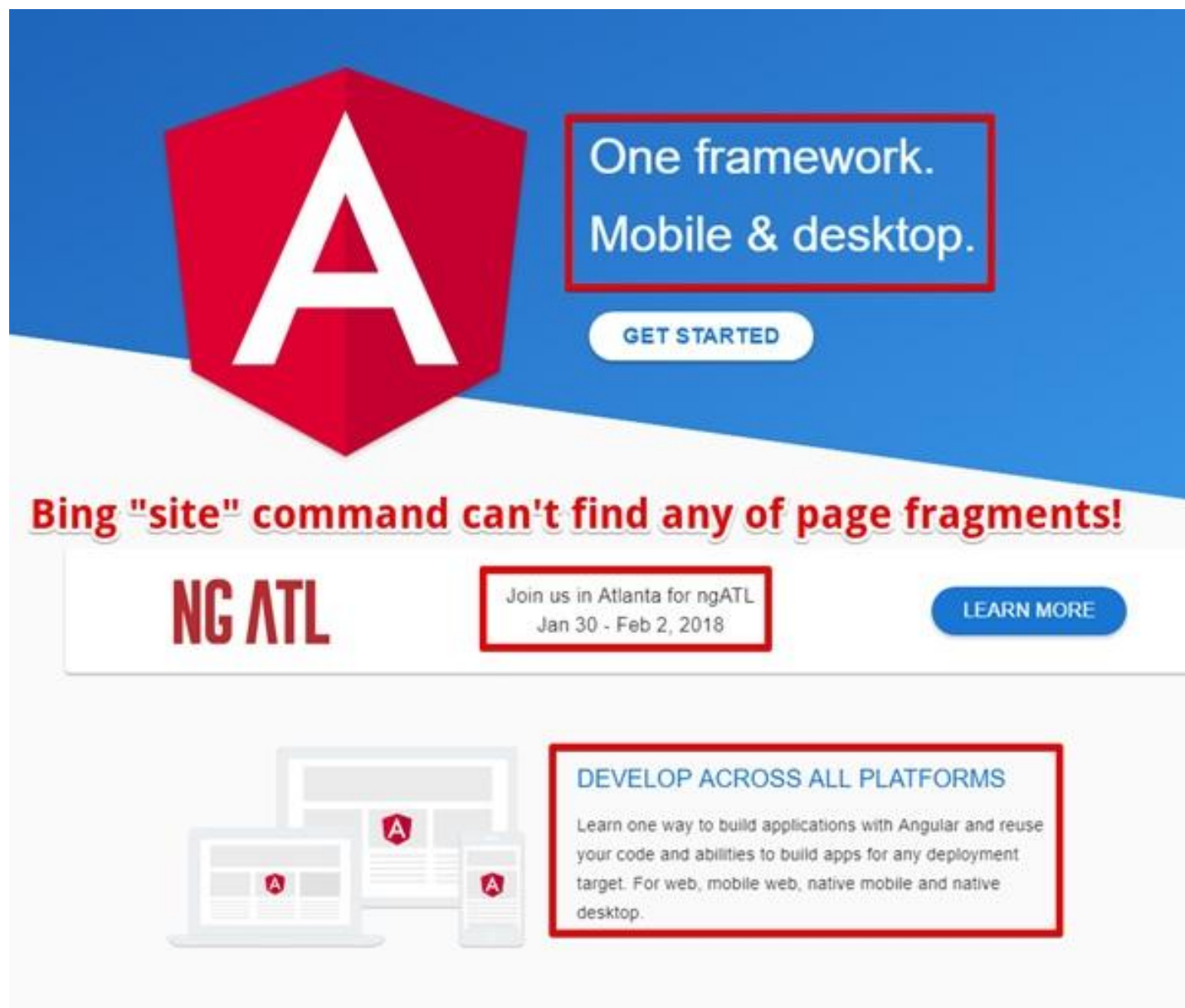


*Bing не видит содержимого сайта Angular.io*

Этот сайт занимает вторую позицию по ключевому слову «Angular» в Bing.

Как насчёт запроса «Angular 4». Снова — вторая позиция, ниже сайта AngularJS.org (это — официальный сайт Angular 1). По запросу «Angular 5» — опять вторая позиция.

Если вам нужны доказательства того, что Bing не может работать с Angular.io — попытайтесь найти какой-нибудь фрагмент текста с этого сайта, воспользовавшись командой `site`. У вас ничего не получится.



The image is a screenshot of the Angular website. It features a large red shield logo with a white 'A' on the left. To the right, a blue box contains the text 'One framework. Mobile & desktop.' and a 'GET STARTED' button. Below this, a red banner reads 'Bing "site" command can't find any of page fragments!'. Under the banner, there is a section for 'NG ATL' with a date 'Jan 30 - Feb 2, 2018' and a 'LEARN MORE' button. At the bottom, there is an illustration of a laptop, desktop monitor, and smartphone, all displaying the Angular logo. To the right of this illustration, a red box contains the text 'DEVELOP ACROSS ALL PLATFORMS' and a paragraph about building applications with Angular.

One framework.  
Mobile & desktop.

GET STARTED

**Bing "site" command can't find any of page fragments!**

**NG ATL**

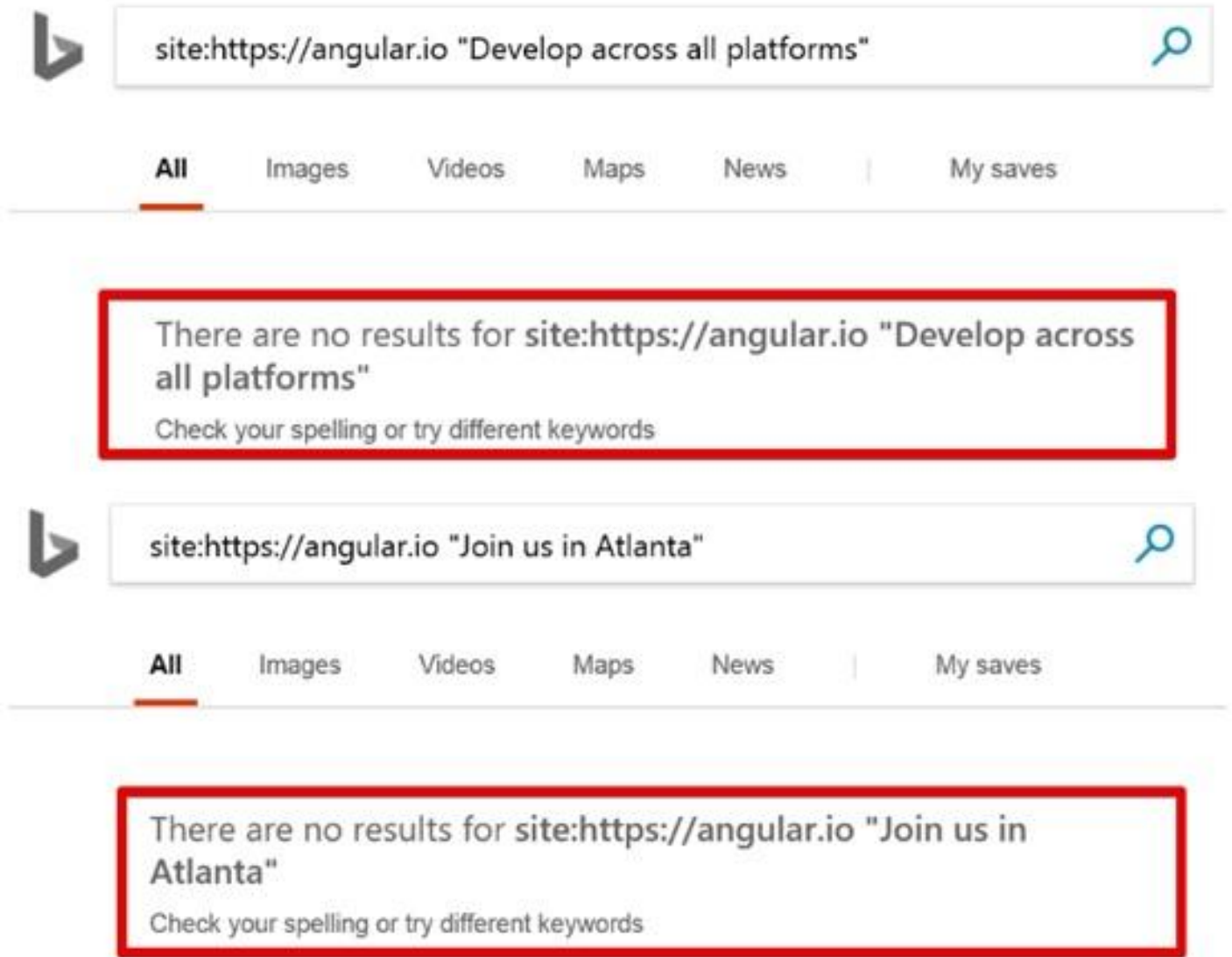
Join us in Atlanta for ngATL  
Jan 30 - Feb 2, 2018

LEARN MORE

**DEVELOP ACROSS ALL PLATFORMS**

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

*Выбор фрагмента текста для проверки*



*Текст, при поиске по сайту, не найден*

По мне, так это странно. Получается, что официальный сайт Angular 2 не может быть нормально просканирован и проиндексирован роботом Bingbot.

А как насчёт Яндекса? Angular.io даже не входит в топ-50 результатов при поиске по слову «Angular» в Яндексе!

## Website about Angular1 - HTML-based - top 1

### AngularJS — Superheroic JavaScript MVW Framework

[Developer Guide](#) [Download](#) [2.0.0-snapshot](#) [FAQ](#)

[angularjs.org](#)

This site and all of its contents are referring to AngularJS (version 1.x), if you are looking for the latest **Angular**, please visit [angular.io](#).

### Angular Docs [angular.io](#)

**Website about Angular2+ (JS-BASED)  
doesn't even rank top 50 for "Angular" in Yandex**

### Начало работы с AngularJS | 1) Загрузить angular.js

[ru.code-maven.com](#) > [getting-started-with-angularjs](#)

Чтобы начать работу с AngularJS, нам нужна HTML страница с тремя вещами: 1) Загрузить [angular.js](#).

### Angular, React and Javascript MV\* Frameworks | MVish

[mvish.com](#)

Angular directives that allow you to build sortable lists with the native HTML5 drag & drop API.

### Angular JS Tutorial for Android Free Download - 9Apps

[angular-js-tutorial.en.9apps.com](#)

9Apps provide best free android apps download. 10,000+ users downloaded Angular JS Tutorial latest version on 9Apps for free every week!

позавчера

### Angular

[userboard.net](#) > [tag/angular](#)

angular element clickedu angular route provider chamillonaire dirty lyrics it toher bookcases covered illinois daycare angular element positioning angular routeprovider...

4 февраля

Вместе с «angular» ищут:

react	react js
node.js	bootstrap
twitter bootstrap	typescript
ember.js	npm
mvc	vue js

В начало 3 4 5 6 7 дальше

Сайт [Angular.io](#) в Яндексе

В Твиттере я [обратился](#) к команде [Angular.io](#), задал им вопрос о том, планируют ли они что-то делать для того, чтобы их сайт могли

индексировать поисковые системы вроде Bing, но они, на момент написания этого материала, пока не ответили мне.

Собственно говоря, из вышесказанного можно сделать вывод о том, что, в стремлении внедрять на свои сайты новые веб-технологии не стоит забывать о Bing и о других поисковиках. Тут можно порекомендовать два подхода: изоморфный JavaScript и предварительный рендеринг.

## Предварительный рендеринг и изоморфный JavaScript

Когда вы замечаете, что Google испытывает сложности с индексацией вашего сайта, рендеринг которого выполняется на клиенте, вы можете рассмотреть возможность использования предварительного рендеринга или преобразования сайта в изоморфное JavaScript-приложение.

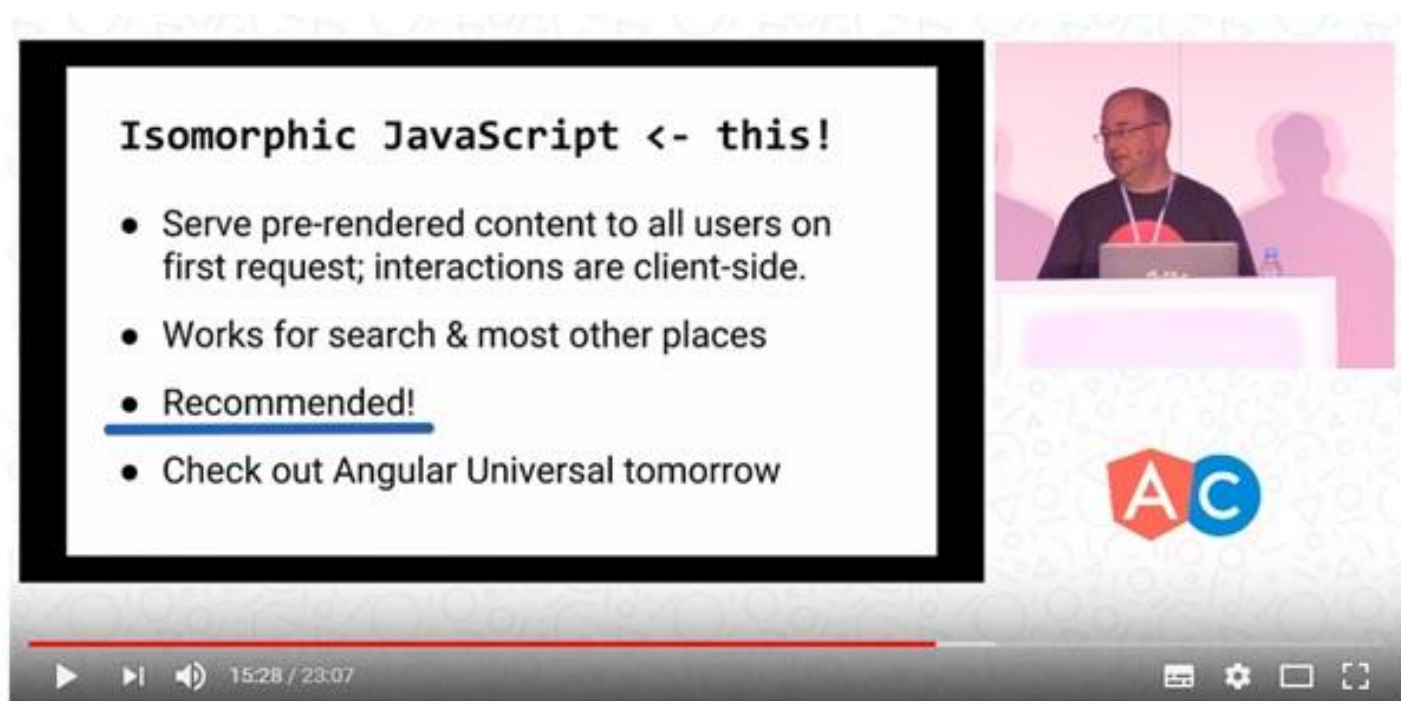
Какой из подходов лучше?

- Предварительный рендеринг применяется в тех случаях, когда вы замечаете, что поисковые роботы не в состоянии правильно сформировать страницы вашего веб-сайта и вы выполняете эту операцию самостоятельно. Когда робот посещает ваш сайт, вы просто отдаёте ему HTML-копии страниц (они не содержат JS-кода). В то же время, пользователи получают версии страниц, оснащённых возможностями JS. Копии страниц используются лишь ботами, но не обычными пользователями. Для предварительного рендеринга страниц вы можете применять внешние сервисы (вроде prerender.io), или использовать



инструменты вроде PhantomJS или Chrome без пользовательского интерфейса на своих серверах.

- **Изоморфный JavaScript** — это ещё один популярный поход. При его применении и поисковая система, и пользователи, при первой загрузке страницы, получают все необходимые данные. Затем загружаются JS-скрипты, которые работают уже с этими, предварительно загруженными, данными. Это хорошо и для обычных пользователей, и для поисковых систем. Данный вариант рекомендуется к использованию, его поддерживает даже Google.



The image shows a video player interface. The main content is a slide with a black border. The slide title is "Isomorphic JavaScript <- this!". Below the title is a bulleted list:

- Serve pre-rendered content to all users on first request; interactions are client-side.
- Works for search & most other places
- Recommended!
- Check out Angular Universal tomorrow

In the background of the slide, there is a video of a man speaking at a podium. To the right of the slide, there is a large, faint watermark of the letters "AC" in a stylized font. The video player controls at the bottom show a progress bar at 15:28 / 23:07 and various icons for play, volume, and settings.

SEO best practices and requirements for modern sites | John Mueller

*Google рекомендует изоморфный JavaScript*

Однако тут есть одна проблема, которая заключается в том, что многие разработчики не могут правильно создавать изоморфные JavaScript-приложения.

Если вас привлекает серверный рендеринг — посмотрите документацию по используемому вами JS-фреймворку. Например, в случае с Angular вы можете применить [Angular Universal](#). Если вы работаете с React — почитайте [документацию](#) и посмотрите этот [учебный курс](#) на Udemu.

React 16 (он вышел в ноябре) принёс в сферу серверного рендеринга [множество улучшений](#). Одно из таких улучшений — функция `RenderToNodeStream`, которая упрощает весь процесс серверного рендеринга.

Если говорить об общем подходе к серверному рендерингу, то хотелось бы отметить одну важную рекомендацию. Она заключается в том, что если нужно, чтобы сайт рендерился на сервере, разработчикам следует избегать использования функций, которые напрямую воздействуют на DOM.

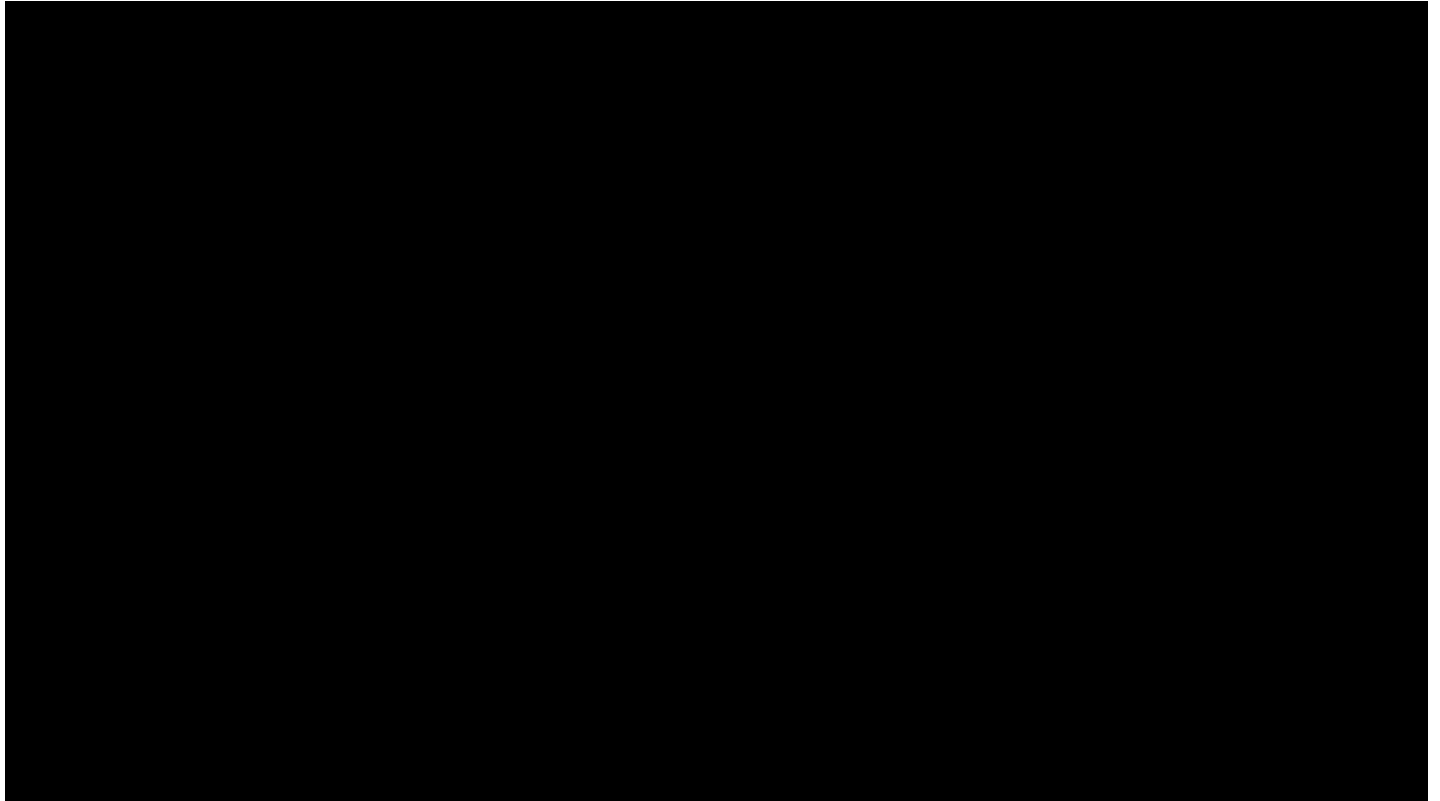
Всегда, когда это возможно, дважды подумайте прежде чем работать напрямую с DOM. Если вам нужно взаимодействие с DOM браузера — воспользуйтесь Angular Renderer или абстракцией рендеринга.

## **Одинаково ли Googlebot обрабатывает HTML и JS-сайты?**

Мы, в Elephate, провели некоторые эксперименты для того, чтобы выяснить, насколько глубоко Googlebot может продвинуться в обнаружении ссылок и в переходах по ним в случае сканирования сайтов, использующих лишь HTML, и сайтов, основанных на

## JavaScript.

Исследование дало поразительные результаты. В случае HTML-сайта Googlebot смог проиндексировать все страницы. Однако, при обработке сайта, основанного на JS, вполне обычной была ситуация, когда Googlebot не добрался даже до его второго уровня. Мы повторили эксперимент на пяти различных доменах, но результат всегда был одним и тем же.



*Эксперимент по индексированию HTML-сайтов и сайтов, основанных на JS*

Бартош Горалевич обратился к Джону Мюллеру из Google, и задал ему вопрос об источниках проблемы. Джон подтвердил, что Google видит ссылки, сгенерированные средствами JavaScript, но «[Googlebot не стремится их сканировать](#)». Он [добавил](#): «Мы не сканируем все URL, или сканируем все их быстро, особенно когда

наш алгоритм не уверен в ценности URL. Ценность контента — неоднозначный аспект тестовых сайтов».

Если вы хотите углубиться в эту тему, рекомендую взглянуть на [этот материал](#).

## Подробности об эксперименте по индексированию тестовых сайтов

Хочу поделиться некоторыми моими идеями относительно вышеописанного эксперимента. Так, нужно отметить, что хотя исследуемые сайты были созданы лишь в экспериментальных целях, к их наполнению применялся один и тот же подход. А именно, их тексты были созданы средствами [Articoolo](#) — интересного генератора контента, основанного на технологиях искусственного интеллекта. Он выдаёт довольно хорошие тексты, которые определённо лучше моей писанины.

Googlebot получал два очень похожих веб-сайта и сканировал лишь один из них, предпочитая HTML-сайт сайту, на котором применяется JS. Почему это так? Выдвину несколько предположений:

- Гипотеза №1. Алгоритмы Google классифицировали оба сайта как тестовые, после чего назначили им фиксированное время выполнения. Скажем, это может быть что-то вроде индексирования 6-ти страниц, или 20 секунд рабочего времени (загрузка всех ресурсов и формирование страниц).

- Гипотеза №2. Googlebot классифицировал оба сайта как тестовые. В случае с JS-сайтом он отметил, что загрузка ресурсов занимает слишком много времени, поэтому просто не стал его сканировать.

На вышеупомянутые веб-сайты имеется множество высококачественных реальных ссылок. Многие люди охотно делились ссылками на них в интернете. Кроме того, на них поступал органический трафик. И тут возникает вопрос о том, как отличить тестовый сайт от реального. Это непростая задача.

Есть немалая вероятность того, что если вы создадите новый сайт, который рендерится на клиенте, вы попадёте в точно такую же ситуацию, как и мы. Googlebot, в таком случае, просто не станет его сканировать. Именно тут заканчивается теория и начинаются реальные проблемы. Кроме того, главная проблема здесь заключается в том, что нет практически ни одного примера из реальной жизни, когда веб-сайт, или интернет-магазин, или страница компании, на которых применяется клиентский рендеринг, занимают высокую позицию в поисковой выдаче. Поэтому я не могу гарантировать того, что ваш сайт, наполненный возможностями JavaScript, получит столь же высокую позицию в поиске, как и его HTML-эквивалент.

Вы можете решить, что, наверняка, большинство SEO-специалистов это и так знают, и большие компании могут направить определённые средства на борьбу с этими проблемами. Однако, как насчёт небольших компаний, у которых нет средств и знаний? Это может представлять реальную опасность для чего-то



вроде сайтов маленьких семейных ресторанов, которые используют визуализацию на стороне клиента.

## Итоги

Подведём краткие итоги, обобщив в следующем списке выводы и рекомендации:

- Google использует Chrome 41 для рендеринга веб-сайтов. Эта версия Chrome была выпущена в 2015-м году, поэтому она не поддерживает все современные возможности JavaScript. Вы можете использовать Chrome 41 для того, чтобы проверить, сможет ли Google правильно обрабатывать ваши страницы. Подробности об использовании Chrome 41 можно найти [здесь](#).
- Обычно недостаточно анализировать лишь исходный HTML-код страниц сайта. Вместо этого следует взглянуть на DOM.
- Google — это единственная поисковая система, которая широко применяет рендеринг JavaScript.
- Не следует использовать кэш Google для проверки того, как Google индексирует содержимое страниц. Анализируя кэш? можно лишь увидеть то, как ваш браузер интерпретирует HTML-данные, которые собрал Googlebot. Это не имеет отношения к тому, как Google обрабатывает эти данные перед индексированием.
- Регулярно используйте инструменты Fetch и Render. Однако не полагайтесь на их тайм-ауты. При индексировании могут применяться совершенно другие тайм-ауты.
- Освойте команду `site`.

- Обеспечьте присутствие пунктов меню в DOM ещё до того, как пользователь вызывает это меню щелчком мыши.
- Алгоритмы Google пытаются определить ценность ресурса с точки зрения целесообразности обработки этого ресурса для формирования страницы. Если, по мнению этих алгоритмов, ресурс ценности не представляет, Googlebot, возможно, не станет его загружать.
- Веб-разработчикам стоит стремиться к тому, чтобы обеспечить скорость выполнения скриптов (множество экспериментов указывают на то, что Google вряд ли станет ждать результатов выполнения скрипта, включая этап загрузки, более 5 секунд). Кроме того, постарайтесь оптимизировать скрипты. Как правило, большинство скриптов можно улучшать и улучшать!
- Если Google не может сформировать страницу, используя JavaScript, он может решить проиндексировать изначально загруженный HTML. Это может очень плохо сказаться на одностраничных приложениях, так как Google, при таком подходе, проиндексирует пустую страницу.
- Нет практически ни одного реального примера сайта, применяющего рендеринг средствами клиента, который занимает высокие позиции в поисковой выдаче. Как вы думаете, почему?

С точки зрения SEO неясно, обрабатывает ли (и ранжирует ли!) Google страницы, основанные на JavaScript так же, как страницы, где используется только HTML. Зная это, очевидно то, что SEO-специалисты и разработчики только начинают подходить к пониманию того, как сделать современные JS-фреймворки такими, чтобы сайты, построенные на их основе, нормально

обрабатывались бы поисковыми роботами. Поэтому важно помнить о том, что в сфере SEO для JavaScript-сайтов нет некоего универсального свода правил. Каждый веб-сайт своеобразен. В итоге же можно сказать, что если вы хотите создать сайт, всюю использующий возможности современного JavaScript, убедитесь в том, что ваши разработчики и специалисты в области SEO знают свою работу.

**Уважаемые читатели!** Сталкивались ли вы, в личной практике, с проблемами индексации сайтов, основанных на JS-фреймворках?

Habrahabr10

Промо-код для скидки в 10% на наши виртуальные сервера

Проголосовать:



+24



Поделиться:



Сохранить:



Комментарии (2)

Похожие публикации

Неявное преобразование типов в JavaScript.  
Сколько будет `!+[]+[]+![]?`

29

ПЕРЕВОД

ru\_vds • 30 января в 14:13

## JavaScript-прокси: и красиво, и полезно

ПЕРЕВОД

ru\_vds • 29 января в 12:23

13

## JavaScript ES8 и переход на async / await

ПЕРЕВОД

ru\_vds • 10 октября 2017 в 14:58

107

## Популярное за сутки

### Наташа — библиотека для извлечения структурированной информации из текстов на русском языке

alexkuku • вчера в 16:12

14

### Unit-тестирование скриншотами: преодолеваем звуковой барьер. Расшифровка доклада

lahmatiy • вчера в 13:05

4

### Люди не хотят чего-то действительно нового — они хотят привычное, но сделанное иначе

ПЕРЕВОД

Smileek • вчера в 10:32

25

## Руководство по SEO JavaScript-сайтов. Часть 2. Проблемы, эксперименты и рекомендации

ПЕРЕВОД

ru\_vds • вчера в 12:04

2

## Как адаптировать игру на Unity под iPhone X к апрелю

P1CACHU • вчера в 16:13

0

## Лучшее на Geektimes

## Стивен Хокинг, автор «Краткой истории времени», умер на 77 году жизни

HostingManager • вчера в 13:49

33

## Обзор рынка моноколес 2018

lozga • вчера в 06:58

70

## «Битва за Telegram»: 35 пользователей подали в суд на ФСБ

alizar • вчера в 15:14

40

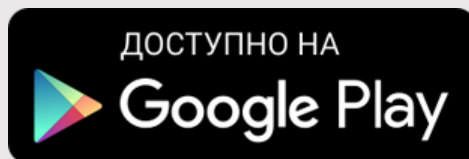
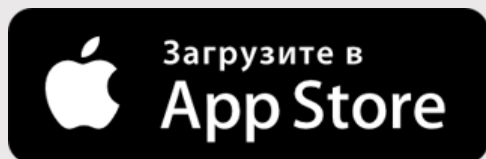
## Стивен Хокинг и его работа — что дал ученый человечеству?

marks • вчера в 14:46

8



Мобильное приложение



Полная версия

2006 – 2018 © TM