AUTOMATED TESTING FOR SEO
with Mike King & Hamlet Batista
Your Speakers

Mike King
Managing Director
at iPullRank

Hamlet Batista
CEO
at RankSense
Has this ever happened to you?
Automated Testing for SEO

https://twitter.com/iPullRank/status/1045435603851063296
Can't imagine much worse than a noindex tag being added to the live site...

6:11 PM - 27 Sep 2018

2 Likes

https://twitter.com/iPullRank/status/1045435603851063296
They used "[name of company] Blog" in an H1 on every blog page as part of the template. The post title was H2.

https://twitter.com/iPullRank/status/1045435603851063296
I got threatened with being sued, a week after a client signed a contact, because their Google traffic went to zero. I hadn’t started as they had asked me to wait a week as their site redesign was finished. That redesign included adding Disallow: / to robots.txt

https://twitter.com/iPullRank/status/1045435603851063296
I also once had a dev let a site update go live without consulting the SEO team, so all site arch., schema, metas, and the whole nine yards were wiped out. That was a NIGHTMARE 😭

6:14 PM - 27 Sep 2018

https://twitter.com/iPullRank/status/1045435603851063296
Multi-million page site having massive de-indexing (millions of pages) due to bad JS on relaunch. Revenue model collapsed.

6:27 PM - 27 Sep 2018 from Kailua, HI

https://twitter.com/iPullRank/status/1045435603851063296
Reverting canonical tags to old versions of URLs. No one noticed until the old URLs started showing up in GA.
What is Automated Testing?
So first, let’s talk just a little bit about modern web development and how it works.
A lot of modern software frameworks follow this design pattern that allows for a separation of concerns between data, logic and frontend.
Why Does This Matter?

In this example the PHP framework Laravel manages the Model and the Controller and the JavaScript framework React acts as the View.
A URL is Converted into a Route

URLs point to controllers which then return the right views featuring data from the right models.

This is why segmented crawls are important to developers being able to implement recommendations.
Waterfall Rarely Makes Sense Anymore

It’s unlikely that the websites you work on will follow a waterfall methodology. Everything is agile now.
Most Dev Teams Follow Agile Methodology

Small pieces of functionality are tackled in short time frames or “sprints.”
Continuous Integration

Often engineering teams are following a Continuous Integration model where teams and individuals are autonomously pushing their code as they complete it.
MOVE FAST AND BREAK THINGS
Deployment Pipelines feature Automated Tests

Deployment pipelines allow for safeguards to be put in place to ensure that everyone is following the same process to get things deployed.
Certain tests are deemed as critical to the infrastructure of the website and will cause the deployment process to fail and the developer has to go back and fix it before they can continue with their work.
MOVE FAST WITH STABLE INFRA
There are three flavors of testing that are most relevant to us as SEOs: Unit Tests, Integration Tests and GUI Tests.
Unit Testing

With Unit tests you are testing an individual function or procedure to ensure that it works.

These are generally small and fast. Codebases can have many of these.

Unit tests can be used to determine the presence of tags and attributes in structure of code to be served to the client, but not the rendered code.
Integration Testing

With integration testing you’re testing how your logic comes together to build the bigger system. It can involve external data such as APIs and databases or how the various pieces of the systems work together.

This is significantly slower than unit testing, but depends on how much of the system needs to be rendered in order to complete the test.

Integration tests can be used to determine tags, features and content of completed HTML source being served to the client. However, there can be overlap with what can be viewed in an UI test.
UI Testing is the testing of the frontend interface of your website. This is largely used for ensuring front end components work as intended.

This is the slowest version of testing because it requires the page to be fully rendered.

UI tests can be used to determine tags, features and content of the rendered DOM being served to the client.
This is What Headless Browsers are Actually For

Certain tests are deemed as critical to the infrastructure of the website and will cause the deployment process to fail and the developer has to go back and fix it before they can continue with their work.
FAQ

Q: Who maintains Puppeteer?

The Chrome DevTools team maintains the library, but we’d love your help and expertise on the project! See Contributing.

Q: What are Puppeteer’s goals and principles?

The goals of the project are:

- Provide a slim, canonical library that highlights the capabilities of the DevTools Protocol.
- Provide a reference implementation for similar testing libraries. Eventually, these other frameworks could adopt Puppeteer as their foundational layer.
- Grow the adoption of headless/automated browser testing.
- Help dogfood new DevTools Protocol features...and catch bugs!
- Learn more about the pain points of automated browser testing and help fill those gaps.

https://github.com/GoogleChrome/puppeteer
Some modern developers follow this methodology wherein they write their tests first and then write their code until the tests pass. Unit tests are primarily used for this, but any type of test could be used. Note: UI Tests will dramatically slow down deployment.
How Does This Apply to SEO?
Last Year I Talked about Using Task Runners

https://searchengineland.com/get-developers-implement-seo-recommendations-280318
I explained the Anderson-Alderson Scale

This is a scale of which the developers you interface with are somewhere between an underachieving known-it-all and overachieving do-it-all.
@BritneyMuller brings up a great point

My goal with that post and talk is to introduce you to another way to integrate developers and ensure we’re all moving towards working better together.

Britney Muller
@BritneyMuller

I'd love to flip this convo:

There's A TON we can learn from developers!

Lack of communication & ownership is often to blame for dev mishaps.

Hope to build content about cultivating healthier dev/SEO relationships very soon.

8:39 PM - 27 Sep 2018

6 Likes
Let’s go back through those horror stories.
A Test Can Catch This

Unit Test Failure Condition – Presence of meta noindex tag on functions that impact the <head>.

Integration Test Failure Condition – Presence of meta noindex on routes that should not have them.

This is a critical issue so this test should fail the build.
Not Sure I’d Test for this One

Integration Test Failure Condition –

H1 tag is the same in the HTML on every accessed page in our test dataset.

UI Test Failure Condition -

H1 tag is the same in the rendered DOM on every accessed page in our test dataset.

I would not recommend having this test fail the build though.
Robots.txt Test

Unit Test Failure Condition –

Integration Test Failure Condition –
Presence of “User-agent: * Disallow: /” on prod in robots.txt URL.

This is a critical issue so this test should fail the build.
SSR Test

Integrated Test/UI Test Failure

Condition -  
Key body content is not available in the HTML version.

This one requires the rendered DOM as a comparison so it technically could be a UI and/or Integrated Test.

This should definitely fail the build.
Rel-canonical

**UI Test Failure Condition** –
URLs that should have a self-referencing canonical do not in the DOM.

**Integration Test Failure Condition** –
URLs that should have a self-referencing canonical do not in the HTML.

**Unit Test Failure Condition** -
Route does not set canonical tag on the server side.

This is a critical issue so this test should fail the build.
Invalid Tag Tests

UI Test Failure Condition –
Rendered DOM features tags in the <head> sub-tree that are invalid.

Integration Test Failure Condition –
HTML features tags in the <head> that are invalid or HTML does not pass W3C validation.

Unit Test Failure Condition –
Function that returns the <head> features tags that don’t fit a list of valid <head> tags.

This is a critical issue so this test should fail the build.
But Testing is Not a Magic Bullet

Sure, we can develop tests for to look for the presence of structured markup and to enforce metadata specifications, but when things go this wrong, you have to put your team member, project management, consultant, management hat on and get to the bottom of what went wrong.
Who Usually Does this Work?

Developer

QA Engineer

https://searchengineland.com/get-developers-implement-seo-recommendations-280318
Doing an SEO Audit is basically doing similar work to that of a QA Engineer.
Crawling HTML is Somewhat like Integration Testing

You’re looking at the results of what the software spits out when all the systems come together.
Crawling JavaScript is more like UI Testing

You’re looking at the results when the user interface is rendered.
...And you should be comparing the differences

John Hogg from iProspect shared his diff comparison Chrome extension that computes the DIFF between HTML and rendered DOM. When it's released I encourage you to check it out.
Your Role is To Help Plan the Tests

You'll want to work with a QA Engineer, a Product Manager or a Developer to develop test cases that can then be built into.
## Here are Few Examples

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>Test Scenario</th>
<th>Test Type</th>
<th>Test Steps</th>
<th>Test Data</th>
<th>Expected Results</th>
<th>Actual Results</th>
<th>Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEO01</td>
<td>Presence of Meta Descriptions</td>
<td>Unit Test</td>
<td>Check for presence of meta description tag in HTML</td>
<td>Page template code</td>
<td>All URLs should have meta description</td>
<td>Product Detail Page is missing meta description</td>
<td>PASS</td>
</tr>
<tr>
<td>SEO02</td>
<td>Viable Internal Links</td>
<td>Functional Test</td>
<td>1. Render pages 2. Open all internal links 3. Review response codes</td>
<td>Crawled URL data</td>
<td>All links return 200 response code</td>
<td>Many links to redirects and 404s</td>
<td>FAIL</td>
</tr>
<tr>
<td>SEO03</td>
<td>Average Page Speed Less than 2 Seconds</td>
<td>Functional/Integration Test</td>
<td>1. Render pages 2. Capture page speed 3. Determine average page speed per page type</td>
<td>Render all page types from URL list</td>
<td>All page types should return an average of 2 seconds load time</td>
<td>Homepage takes 5 seconds to load</td>
<td>FAIL</td>
</tr>
</tbody>
</table>

These examples of tests should be enough to get you started on defining your own potential tests for SEO.
Tools

Some tools you can use to get your automated testing for SEO off the ground
Meet Jenkins

If your developers are doing Continuous Integration or Continuous Delivery, it’s likely that they are using Jenkins.

https://jenkins.io/
Get Failure Alerts in Slack

A Jenkins plugin for posting notifications to a Slack channel.

Install Instructions for Slack

1. Get a Slack account at https://slack.com/
2. Configure the Jenkins integration: https://my.slack.com/services/new/jenkins-ci

https://plugins.jenkins.io/slack
Or in Jenkins’ logs

You could also just regularly the logs that Jenkins provides.
Jenkins logging directly to MySQL database

tags: groovy jenkins mysql

Written on January 27, 2017

There are cases if you like to send the log output from Jenkins directly to a database for better querying or just as a step for further processing them.

As Jenkins is not really admin-friendly with redirecting logs, we have to use the init.d feature of Jenkins (https://wiki.jenkins-ci.org/display/JENKINS/Configuring+Jenkins+upon+startup). Thus you can simply put an init.groovy file into the Jenkins home directory. In this init file you can do a lot of stuff like setting some variables, disabling the sending of usage statistics and so on. And you can hook into the LogManager of Jenkins: https://wiki.jenkins-ci.org/display/JENKINS/Logging

In this example we are using Jenkins 1.651.3

Jenkins is using the java.util.logging package, so first of all we need a handler for sending LogRecords to a MySQL database:

```groovy
import java.util.logging.*;
import java.sql.*;

class MySQLHandler extends Handler {
    def driverString = "com.mysql.jdbc.Driver";
    def insertSQL = "INSERT INTO log (level,logger,message,sequence,sourceClass,sourceMethod,threadID,timestamp) VALUES(?, ?, ?, ?, ?, ?, ?, ?)"
    def connectionString;
    def connection;
    def pStmt;
    def pStmtInsert;

    public MySQLHandler(String connectionString) {
        this.connectionString = connectionString;
        Class.forName(driverString);
        connection = DriverManager.getConnection(connectionString);
        connection.setAutoCommit(true);
        pStmt = connection.prepareStatement(insertSQL);

        pStmt.setString(1, level);
        pStmt.setString(2, logger);
        pStmt.setString(3, message);
        pStmt.setString(4, sequence);
        pStmt.setString(5, sourceClass);
        pStmt.setString(6, sourceMethod);
        pStmt.setString(7, threadID);
        pStmt.setString(8, timestamp);
        pStmt.executeUpdate();

        pStmtInsert.setString(1, level);
        pStmtInsert.setString(2, logger);
        pStmtInsert.setString(3, message);
        pStmtInsert.setString(4, sequence);
        pStmtInsert.setString(5, sourceClass);
        pStmtInsert.setString(6, sourceMethod);
        pStmtInsert.setString(7, threadID);
        pStmtInsert.setString(8, timestamp);
        pStmtInsert.executeUpdate();
    }
}
```

Sending the Jenkins logs into MySQL gives you the opportunity to monitor and visualize failures and successes of builds with respect to SEO in a variety of ways such as through Google Data Studio.
You’ll Also Need a Crawler

https://github.com/yujiosaka/headless-chrome-crawler

https://github.com/antivanov/js-crawler

You’ll want a headless crawler and a text-based crawler to spin up pages and run tests on a fixed list of pages that represent all of the public-facing routes.
You’ll Also Need a Crawler

Building your crawler with AWS Lambda, Azure Functions or Google Clouds Functions may be a great way to enjoy scale without having to manage additional servers.
Shoutout to @vvgomes
Vini’s Blog Post on Serverless Architectures

Event Driven Architectures & Serverless

MONDAY, JULY 23, 2018 - 6 MINS

After a 2.5 years journey building an event-driven services platform for a large financial organization, I’ve recently joined a new team in the middle of their ongoing migration to the public cloud. One of the things we’ve been exploring is the Serverless model in order to release software with close to none operations effort.

Turns out there are significant commonalities between EDAs (Event Driven Architectures) and Serverless which can lead to interesting advantages from the architectural point of view when applied to the appropriate business context. That was the topic of my recent presentation at the North America XConf!

http://vvgomes.com/edas-and-serverless/
Tips On Testing

Make only mission critical SEO tests potentially fail the build.

Most SEO issues will be spotted in your integration tests.

Include your SEO tests in every build.

Keep your testing suites separate.

Do all crawling in parallel.
Things to Play With

Because I never like to leave you without having something you can do
Check out these Video Tutorials on Jenkins

Get started with Jenkins

In these video tutorials I’ll show you how to get started with Jenkins, the open source continuous integration & delivery application. I’ll cover topics like installing Jenkins, importing git source code,

Play with Jenkins:

https://www.youtube.com/playlist?list=PLzvRQMJJHDiSaisKr70nM4FI7JXCDDcmt
Kantu Browser Automation Tool

See how Browser Automation tests work with a chrome extension:
https://chrome.google.com/webstore/detail/kantu-browser-automation/gcbalfbdmfieckjlnblleoemohcganoc?hl=en
If you dabble in PHP or have a WP site you want to play around with, check out Codeception. https://codeception.com
Puppeteer

The Headless Chrome library has tons of scraping use cases.
https://github.com/GoogleChrome/puppeteer
PuPHPeteer

PuPHPeteer

A Puppeteer bridge for PHP, supporting the entire API. Based on Rats, a package to manage Node resources from PHP.

Here are some examples borrowed from Puppeteer's documentation and adapted to PHP's syntax.

Example - navigating to https://example.com and saving a screenshot as example.png:

```php
use Nesk\PuPHPeteer\PuPHPeteer;
use Nesk\PuPHPeteer\helpers\functions;

$puphpeteer = new PuPHPeteer;
Browser = $puphpeteer->launch();
Page = Browser->newPage();
Page->goto('https://example.com');
Page->screenshot(['path' => 'example.png']);
Browser->close();
```

Example - evaluate a script in the context of the page:

```php
use Nesk\PuPHPeteer\PuPHPeteer;
use Nesk\PuPHPeteer\helpers\functions;

$puphpeteer = new PuPHPeteer;
Browser = $puphpeteer->launch();
Page = Browser->newPage();
Page->goto('https://example.com');
// get the "viewport" of the page, as reported by the page.
$dimensions = Page->evaluate('[object Object]');
return [
    'width' => document.documentElement.clientWidth,
    'height' => document.documentElement.clientHeight,
    'devicesPixelRatio' => window.devicePixelRatio
];
```

PHP Wrapper for Puppeteer

https://github.com/nesk/puphpeteer
@Aleyda Mentioned Monitoring Tools

Aleyda Solis @aleyda

If you want to automatically monitor some of the most fundamental SEO related configurations I highly recommend you check out @littlewardenapp, @contentking and @seomonitor 🤘🛠️

MyCool King @iPullRank

Ladies and gentlemen, I'd like to share with you a presentation called "Software Testing for SEO"

bit.ly/save-the-seo...

5:34 AM - 28 Sep 2018
Monitoring is \textit{not the same} as a automated testing.
Thanks @aleyda!

@iPullRank ContentKing's built to tackle most of the situation you've covered in your presentation. Using our API, you can also integrate it in your CI process 🙏

We'd be happy to set up a demo account for you to play around with! 😎
Agenda

- Diagnose SEO and page speed issues related to the incorrect nesting of HTML tags and scripts.
- Learn to use the Chrome JavaScript Debugger to track down serious SEO issues.
- Speed up JavaScript by removing unused code and implementing code splitting when appropriate.
- Leverage service workers and to edge workers for more powerful use cases.
How Misplaced HTML tags Hurt SEO

Let’s see what happens to the canonical tag when we insert a `<div>` in the HTML HEAD.

```
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>This is a test</title>
    <meta name="description" content="This is a test">
    <meta name="author" content="Hamlet Batista">
    <div>This is shouldn't be here</div>
    <link rel="canonical" href="http://dev.ranksense.com/invalid_head_hard-coded.html">
  </head>
  <body>
    <div>This is a test</div>
  </body>
</html>
```
How Misplaced HTML tags Hurt SEO

It ends up inside the HTML BODY.

But why?
**Browsers’ error tolerance**

You never get an “Invalid Syntax” error on an HTML page. Browsers fix any invalid content and go on.

Take this HTML for example:

```html
<html>
  <mytag>
  </mytag>
  </html>
```

I must have violated about a million rules (“mytag” is not a standard tag, wrong nesting of the “p” and “div” elements and more) but the browser still shows it correctly and doesn’t complain. So a lot of the parser code is fixing the HTML author mistakes.

Error handling is quite consistent in browsers, but amazingly enough it hasn’t been part of HTML specifications. Like bookmarking and back/forward buttons it’s just something that developed in browsers over the years. There are known invalid HTML constructs repeated on many sites, and the browsers try to fix them in a way conformant with other browsers.

---

**How Misplaced HTML tags Hurt SEO**

This is the result of browsers error tolerance:

“the element being added is explicitly forbidden inside some outer tag. In this case we should close all tags up to the one which forbids the element, and add it afterwards.”

How Misplaced HTML tags Hurt SEO

Move SEO tags to the top of the HTML HEAD.

Check: The dangers of misplaced third-party scripts

How to fix this

Fortunately, fixing this problem is actually very simple. We have two alternatives. A lazy one:

The proper fix is to track down scripts that insert invalid HTML tags in the head and move them to the HTML body.

The lazy and quickest fix is to move all SEO tags (and other important tags) before any third-party scripts. Preferably, right after the opening <HEAD> tag.

You can see how I did it here.

```html
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>This is a test</title>
<link rel="canonical" href="http://localhost">
<meta name="description" content="This is a test">
<meta name="author" content="Hamlet Batista">
<script>
document.write("<div>This script is misplaced</div>");
</script>
</head>
<body>
</body>
</html>
```
How Misplaced HTML tags Hurt SEO

Does this affect Googlebot?
If it does, does the fix work too?

Let’s see!
How Misplaced HTML tags Hurt SEO

If the page is missing the BODY tag, Google adds it back.

Good.
How Misplaced HTML tags Hurt SEO

If we add a `<DIV>` manually to the HTML HEAD, Google pushes our canonical to the BODY. Same as in the browser.
How Misplaced HTML tags Hurt SEO

If we add a `<DIV>` to the HTML HEAD using a script, the URL Inspection Tool gives an error and the page doesn't get indexed. The browser can handle the page.
How Misplaced HTML tags Hurt SEO

If we move the SEO tags to the top of the HTML and leave the invalid <DIV>, the canonical remains in the HTML HEAD.
How Misplaced HTML tags Hurt SEO

It is properly detected.
Good.
How Misplaced HTML tags Hurt SEO

The optimal solution is to move the invalid scripts and tags to the HML BODY.
• Misplaced HTML tags in the HEAD can push SEO tags to the BODY
• The issue is visible in the browser and Search Console URL Inspection Tool
• Moving SEO tags to the top of the HTML HEAD helps

Share these #SMXInsights on your social channels!
Let's learn to use the Chrome Debugger to identify obscure scripts that override SEO tags. This test page has a canonical and the script linked overrides it.
Let's use the JavaScript Debugger to track down scripts that override SEO tags.
First, we set up a DOM breakpoint to stop JavaScript execution when the attributes of the canonical tag are modified. Next, we hit refresh.
The Powerful JavaScript Debugger
Tracking Down Unused Code
Tracking Down Unused Code

We can track unused code using the Code Coverage tool.
Tracking Down Unused Code

<table>
<thead>
<tr>
<th>URL</th>
<th>Type</th>
<th>Total Bytes</th>
<th>Unused Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://dev.ranksense.com/unused_script.js">http://dev.ranksense.com/unused_script.js</a></td>
<td>JS</td>
<td>000</td>
<td>100 11.0 %</td>
</tr>
<tr>
<td><a href="http://dev.ranksense.com/unused_code.html">http://dev.ranksense.com/unused_code.html</a></td>
<td>JS</td>
<td>23</td>
<td>16 37.2 %</td>
</tr>
</tbody>
</table>

527 B of 852 B (61%) used so far, 125 B unused.
Tracking Down Unused Code

The red vertical bars highlight the unused code.
Tracking Down Unused Code

**TL;DR**

- By default, CSS is treated as a render blocking resource.
- Media types and media queries allow us to mark some CSS resources as non-render blocking.
- The browser downloads all CSS resources, regardless of blocking or non-blocking behavior.
Tracking Down Unused Code
● The JavaScript Debugger can help track down the scripts that override SEO tags and cause performance issues

● The Code Coverage tool helps identify JavaScript and CSS code that is never used so that we can remove it

Share these #SMXInsights on your social channels!
**Automated Testing for SEO**

Let's learn to use the Chrome Debugger to identify obscure scripts that override SEO tags.

This test page has a canonical and the script linked overrides it.

```html
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>This is a test</title>
    <meta name="description" content="This is a test">
    <meta name="author" content="Henley Batista">
    <link rel="canonical" href="http://dev.ranksense.com/override_canonical.html">
  </head>
  <body>
    <div>This is a test</div>
    <script src="canonical_script.js"></script>
  </body>
</html>
```
Wrapping up.

Who am I and where am I from?
I’m #ZORASDAD

First and foremost.
MY NAME IS MIKE KING

Razorfish, Publicis Modem alum
Full Stack Developer
Full Stack Marketer
Moz Associate
We Do These Things

- Content Strategy
- SEO
- Paid Media
- Machine Learning
- Marketing Automation
- Measurement & Optimization
Machine Learning for Marketers Guide

Machine Learning for Marketers
A comprehensive guide to machine learning

Get the PDF:

http://ipullrank.com/machine-learning-guide/
Eagles Super Bowl Champions
Q&A / Thank You!

Twitter: @ipullrankagency
Email: hello@ipullrank.com
Website: https://ipullrank.com

Get in touch if you need some help!

Mike King
Managing Director
at iPullRank

Hamlet Batista
CEO
at RankSense