Technical and Programmatic SEO

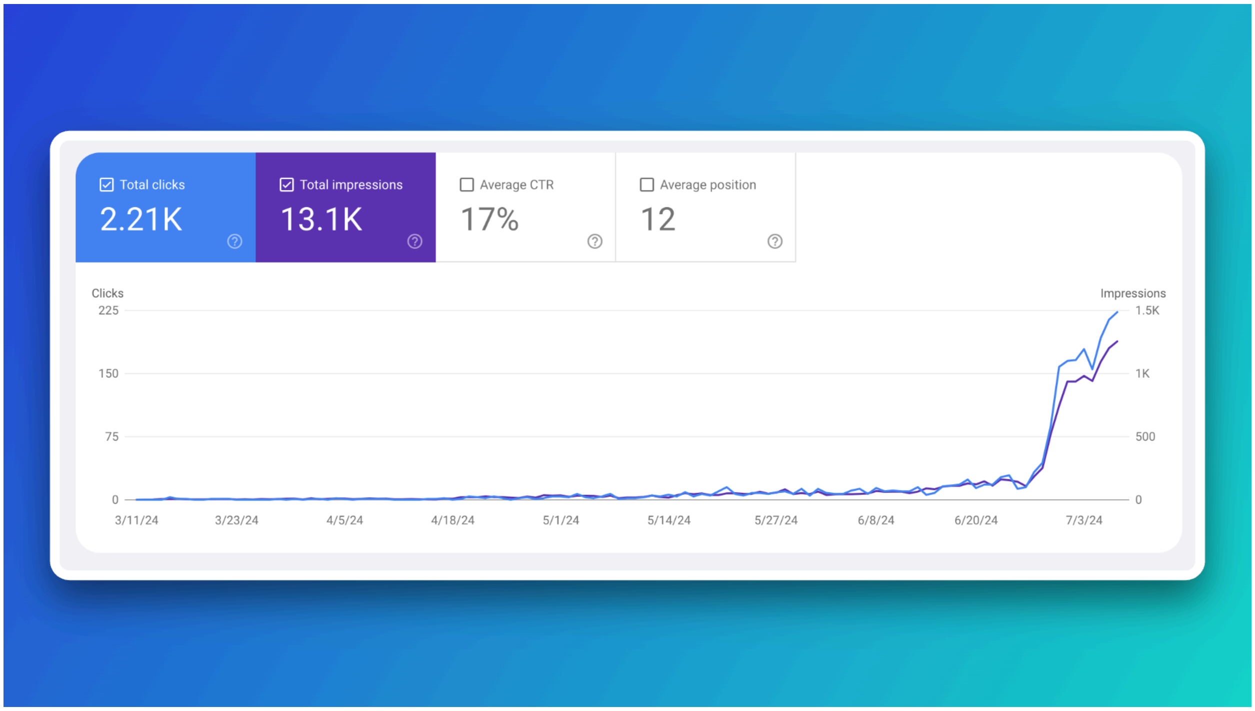
The SEO Guide for Developers Who Hate Marketing

Table of contents

Introduction

Let's start with a fact: SEO is the best way to get a stable, organic traffic for your website, without having to work on your marketing every day.

Imagine spending two days (even one if you're used to it) on your SEO, not having to do it ever again, not having to do marketing at all, to have this chart after a few weeks:



☑ Total clicks 2.21K

☑ Total impressions 13.1K

□ Average CTR 17%

□ Average position 12

Well, this screenshot comes from one of my own websites, that went from about 10 clicks a day to more than 200, two months after working on (technical and programmatic) SEO - in just one day of work.

As you may imagine, the revenue of this product increased as well, going from 20€ a month to 100€ a week.

This is no bs, **technical and programmatic SEO really works**.

Interested to learn more about it? This guide is for you.

I've been working on SEO and programmatic SEO for years, I have now quite some knowledge to share. This document will be the summary of this knowledge.

Of course, I can't promise you'll reach the same result with technical SEO. I can't even promise you any increase in traffic. But it's only one or two days of work, without any marketing - one not try it?

About me

Before getting to the heart of the matter, let me introduce myself (and talk about my SEO products).

I'm Alexandre, web & mobile developer for more than 11 years now. I've also written more than 150 SEO friendly tech articles for clients.

I'm an indie hacker and have been working on dozens of projects over the last years.

Two years ago, I discovered programmatic SEO, and created my own PSEO techniques. It worked so well, that I decided to sell my own implementation of it, and create other products related to SEO. I'll talk about it at then end of this guide, but to summarize, here are my SEO products:

- *Code Driven SEO: online guide + all code provided of my programmatic SEO technique, to x20 your Google impressions and clicks like I did*
- *Rank on Bing: a Saas to automatically request Bing to crawl and (eventually) index your web pages*
- *Free AI sitemap.xml generator: an online tool to generate a sitemap file directly from your code, for free*
- *SpeedMeter: a free Chrome extension to test your website loading speed and tell you what HTTP requests you can improve*

Now that the ad part is done, it's time to talk about technical SEO!

The 3 main components of SEO

First things first, we need a quick introduction to SEO.

In my opinion, your optimization on search engines mainly depends on three factors:

1. **Keywords/text optimization**
2. **Website authority**
3. **Technical website quality**

Let's elaborate on these.

1. Keywords and Text Optimization

You probably know that keywords and text optimization are important. It consists of finding the right keywords depending on your targeted audience, adding them at the right places on your pages, and having good quality text contents.

We're not going to develop this part in this guide, otherwise it would be way too long. If you want to know how to find the right keywords, you'll find some tools on the internet (Google Keyword Planner is a good start)

2. Website Authority

A website with a good authority will rank better on the SERP (search engine result page). But what is a website authority? It is, kind of, the level of trust a search engine gives to your website.

For example, Wikipedia has a high authority, therefore will rank higher and faster, even for new pages. Even if you create a website about a niche topic and create a better content than what's on Wikipedia, it's very likely that the Wikipedia page will still rank higher.

A lot of factors are responsible for a website authority, one of the most important being backlinks. We're not going to talk about it as it's not related to technical SEO, but feel free to do some research about it.

3. Technical Website Quality

Here we are, the part that interests us!

Search engines show to users the websites they want to visit. And users want to visit websites that are **relevant**, that **work properly**, are **fast to load** and **optimized** for their devices.

Therefore, if we're being logical, these are elements that are taken into account by search engines to rank web pages.

If your website is slow to load, doesn't have well formatted HTML elements and doesn't show optimized content for mobile, search engines will not rank it well.

Good thing is, for us developers, it is not hard to do technical SEO. It's even the easiest part of this list, and it can dramatically improve your website's place on SERP.

Let's dive into this.

As you've noticed, I didn't mention programmatic SEO here, because, unlike these three main elements, programmatic SEO is not a mandatory step. It is a technique that can completely skyrocket your website ranking, but a lot of websites rank at the first position without using PSEO (but, they very likely optimized keywords, authority, and technical quality).

Part One: Technical SEO

1 - The importance of Technical SEO

As we already mentioned, a bad technical search engine optimization (such as mobile performance, page load speed, etc.) can lead to low search rankings, which will lead to a limited amount of visitors, which will harm your business.

Therefore, taking care of your technical SEO is an easy and fast way to increase your business.

Enough talking now, let's go to the technical parts!

2 - HTML Structure and Best Practices

a) Title Tags

Title tags are HTML elements that specify the title of a web page. We're not talking about the headings here, but the **unique title of your page**. It is used by search engines to know what your website is about, and it is listed as the clickable headline in SERP.

The Title element is to put in the <head> tag of your HTML page.

Here are a few tips:

- Keep titles **under 60 characters** to ensure they are not truncated in search results.
- Include **relevant keywords at the beginning** of the title.
- Make titles **descriptive but concise**.
- **Avoid keyword stuffing** - better less keywords but clear purpose than the opposite

Example:

```
<head>
    [...]
    <title>Ultimate Guide to Technical SEO for Developers</title>
    [...]
</head>
```

b) Meta Descriptions

Meta descriptions are **short summaries** of a web page’s content. While they don't directly affect rankings, well-written meta descriptions can improve click-through rates, because it might be - it is not true all the time - what is displayed by the search engine on the result page, just under your website title.

The meta tag is also to be included in the <head> element.

A few tips:

- Keep meta descriptions **under 160 characters**.
- **Summarize the page content** accurately.
- Include a **call-to-action if appropriate**.
- Incorporate r**elevant keywords naturally**.

Example:

```
<head>
    [...]
    <meta name="description" content="Learn the essentials of technical SEO for developers, including best practices for HTML structure, website speed, and more.">
    [...]
</head>
```

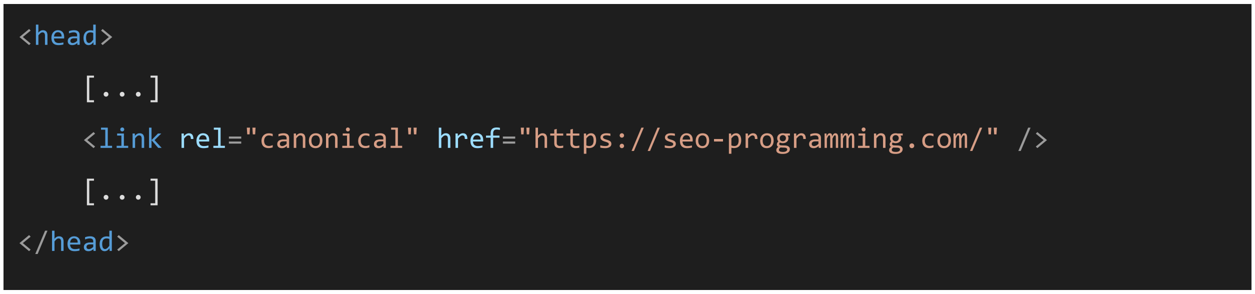
c) Canonical Tags

Canonical tags help prevent duplicate content issues by specifying the preferred version of a web page. They signal to search engines which version of a URL should be indexed. This attribute is not mandatory, but it is advised to add it.

Some tips:

- Use canonical tags to point to the main version of a page.
- Ensure canonical URLs are consistent across different versions of the site (e.g., HTTP vs. HTTPS).

Example:



```
<head>
    [...]
    <link rel="canonical" href="https://seo-programming.com/" />
    [...]
</head>
```

d) Open Graph (OG) Elements

Less famous among developers, OG (Open Graph) elements tags are used to control how URLs are displayed when shared on social media platforms like Facebook, LinkedIn, and Twitter. Although not mandatory, they can increase the click percentage of your website's shared links.

Also to be placed in the <head> element, the og tags have several values:

- `og:title`: The title of your page.
- `og:description`: A brief description of your page.
- `og:image`: URL of an image that represents your page.
- `og:url`: The canonical URL of your page.

Example:

```
<head>
    [...]
     <meta
      property="og:title"
      content="Programmatic SEO: The Developer's Guide to Rapid Ranking"
    />
    <meta
      property="og:description"
      content="The Code Driven SEO Guide is the ultimate documentation for developers who want to improve their websites' SEO but who hate marketing"
    />
    <meta property="og:url" content="https://seo-programming.com/" />
    <meta
      property="og:image"
      content="https://seo-programming.com/img/logofull.png"
    />
    [...]
</head>
```

e) Heading Tags (H1, H2, etc.)

Finally, something you're more likely to know: the heading tags.

Heading tags (H1, H2, H3, etc.) are used to define the structure of your content. They help search engines understand the hierarchy and main topics of your page.

They still matter, and it is important to place them accordingly on your webpage (H1 will have more weight than H2, etc.).

A few tips:

- Use one H1 tag per page, which should contain the main keyword or topic.
- Use H2 tags for subheadings and structure content logically.
- Ensure headings are descriptive and relevant to the content that follows.

Example:

```
<body>
    [...]
    <h1>Ultimate Guide to Technical SEO for Developers</h1>
    [...]
    <h2>Introduction to Technical SEO</h2>
    [...]
    <h3>What is Technical SEO?</h3>
    [...]
</body>
```

3 - HTML Attributes

a) Image Alt Attributes

Alt attributes provide **alternative text for images**, which is used by screen readers and in case the image fails to load. They also help search engines understand the content of the images.

Some tips:

- Describe the image accurately and succinctly.
- Include relevant keywords *if* they are natural to the description.
- Avoid keyword stuffing.

Example:

```
<img src="technical-seo-chart.jpg" alt="Technical SEO performance chart">
```

b) Rel Attributes (nofollow, noreferrer, etc.)

Rel attributes are used on links (<a> tags) to define the relationship between the current document and the linked document. It is also used to prevent search engines from following the link.

Common Rel Attributes:

- `rel="nofollow"`: Prevents search engines from following the link. This attribute is often added by default on social media links (to avoid SEO spamming)
- `rel="noreferrer"`: Prevents the browser from sending the referring URL to the target site.

Example:

```
<a href="https://example.com" rel="nofollow">External Link</a>
```

Note:

Even though `rel="nofollow"` is supposed to prevent search engines from following the link (and therefore, the link is in theory not considered as a backlink), it appears that Google *might* follow the link, but giving it a reduced weight in SEO.

c) Language Attributes

Language attributes specify the language of the content, helping search engines understand the linguistic context of your page.

Some tips:

- Use the `lang` attribute in the `<html>` tag.
- Specify the primary language of the page content.

Example:

```
<html lang="en">
```

d) Data Attributes for SEO

Data attributes (custom attributes that start with `data-`) can be used to store additional information about elements without affecting the visual presentation.

A few tips:

- Use data attributes to store extra information needed for JavaScript or other scripts.
- Avoid using data attributes for information that should be visible or accessible.

Example:

```
<div data-author="Alexandre Grisey"
data-publish-date="2024-07-07">Content</div>
```

Notes:

The weight of these elements is very light for search engines, if not ignored.

4 - Website Performance Optimization

a) Introduction: Importance of Loading Speed

Website loading speed is crucial for both user experience and SEO. Faster websites provide a better user experience, reduce bounce rates, and are favored by search engines.

Key Factors Affecting Loading Speed:

- Server response time
- Image sizes and formats
- Minification of CSS, JavaScript, and HTML
- Use of content delivery networks (CDNs)
- Browser caching

Notes:

Bounce rates can be somehow measured by Google, for example if your visitor goes back to the SERP quickly after clicking on the link leading to your page.

b) Image Optimization

Images often make up a significant portion of a webpage's size. Optimizing images can greatly improve loading speed.

A few tips:

- Use appropriate file formats (JPEG, PNG, etc.).
- **Use .webp** format when possible (it drastically reduces the size of an image, therefore the time it needs to load)
- Compress images without sacrificing quality.

- You could use responsive images with the `srcset` attribute to serve different sizes for different devices.

Example:

```
<img src="image.jpg" srcset="image-small.jpg 480w, image-medium.jpg 768w,
image-large.jpg 1200w" alt="Description of image">
```

Notes:

Some tools exist to help you convert your images and compress them. For example, **ImageMagick**, which is a CLI that allows you to convert/compress all the images in a folder. If you want to compress and convert all your images to .webp, after installing ImageMagick CLI, you could do:

```
magick mogrify -path /images -format webp -quality 80 /images/*.png
/images/*.jpg /images/*.gif
```

c) Minification of CSS, JavaScript, and HTML

Minification removes unnecessary characters from code without changing its functionality, reducing file sizes and improving load times.

Usually, if you use a JavaScript framework, all the sources are automatically minified by pre-installed packages used during build. Just make sure it is, once in production.

d) Lazy Loading

Lazy loading defers the loading of non-critical resources (like images and videos) until they are needed, improving initial load times.

A few tips:

- Use the `loading="lazy"` attribute for images and iframes.
- Implement lazy loading for other elements using JavaScript if necessary.

Example:

```html
<img src="image.jpg" alt="Description of image" loading="lazy">
<iframe src="video.html" loading="lazy"></iframe>
```

Notes:

Most frameworks have their own implementation of lazy loading. Search on their documentation to find out the best way to add it to your project.

e) Browser Caching

Browser caching stores resources locally on users' browsers, reducing load times for subsequent visits.

Best Practices:

- Set appropriate cache headers using HTTP headers.
- Use versioning in file names to force updates when content changes (usually automatically done with JavaScript frameworks builders)

Example (using .htaccess file for Apache server):

```
<IfModule mod_expires.c>
```

```
ExpiresActive On
ExpiresByType image/jpg "access plus 1 year"
ExpiresByType image/jpeg "access plus 1 year"
ExpiresByType image/gif "access plus 1 year"
ExpiresByType image/png "access plus 1 year"
ExpiresByType text/css "access plus 1 month"
ExpiresByType application/pdf "access plus 1 month"
ExpiresByType text/x-javascript "access plus 1 month"
ExpiresByType application/x-shockwave-flash "access plus 1 month"
ExpiresByType image/x-icon "access plus 1 year"
ExpiresDefault "access plus 2 days"
</IfModule>
```

f) Content Delivery Network (CDN) Usage

CDNs distribute content across multiple servers globally, reducing latency and improving load times.

Best Practices:

- Use a CDN to serve static assets (images, CSS, JavaScript).
- Choose a CDN provider that has a wide distribution network.

Popular CDN Providers:

- Cloudflare
- Amazon CloudFront
- Akamai
- Fastly

5 - Mobile-Friendliness

Google uses a "mobile-first" policy to crawl websites. Therefore, having mobile-friendly web pages is mandatory if you want your website to rank well on the SERP.

a) Responsive Design Principles

If you use a CSS framework (Tailwind, Bulma, Bootstrap, etc.), you very likely already use responsive principles even without thinking about it.

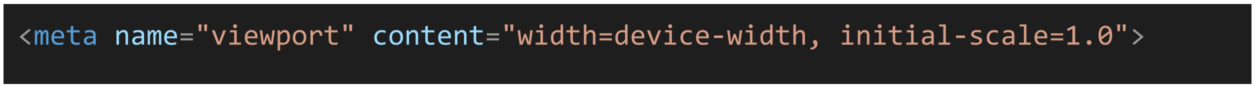
Yet, here is a **summary of good practices**:

- Use flexible grid layouts.
- Use flexible images and media.
- Use CSS media queries to apply different styles for different screen sizes (@media CSS elements)

b) Viewport Configuration

The viewport meta tag controls the layout on mobile browsers. Without it, pages may not display correctly on mobile devices. It is generally automatically added by frameworks or libraries.

Here is what it looks like:



```html
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

6 - Website Security

a) Importance of HTTPS

HTTPS (Hypertext Transfer Protocol Secure) ensures the secure transmission of data between the user’s browser and the server. It’s a ranking factor for Google and essential for protecting user data.

Some tips:

- Obtain an SSL/TLS certificate from a trusted certificate authority (CA). If it's not done automatically, you can generate it via your web server provider
- Configure your server to use HTTPS for all pages.
- Redirect HTTP traffic to HTTPS.
- Make sure that you use HTTPS for every API call/imported script

7 - URL Structure and Management

a) SEO-Friendly URLs

SEO-friendly URLs are clean, descriptive, and easy for both users and search engines to understand.

Few tips:

- Use hyphens to separate words.
- Keep URLs short and relevant; remove empty words ('and', 'or', etc.)
- Avoid using unnecessary parameters and session IDs.
- Use lowercase letters.

- If there is risks of duplicate URLs, add a unique id at the end of the URL, like Medium does (example)

Example:

```
<!-- Incorrect -->
https://example.com/Page.php?ID=123

<!-- Correct -->
https://example.com/technical-seo-guide
```

b) Handling 404 Errors

A 404 error occurs when a page is not found.

Properly handling 404 errors can improve user experience and SEO.

Best Practices:

- Create a custom 404 error page that helps users navigate to other parts of the site.
- Use Google Search Console to identify and fix 404 errors.

Usually, front-end JavaScript frameworks' routers make it easy to handle 404 errors. Read the documentation to know how to do it.

8 - Sitemaps and Crawling

a) XML Sitemaps

A sitemap.xml is a file (in xml format) that lists the pages (URLs) of your website. The goal is to help search engines find your pages by parsing this xml file, to then crawl them, and eventually index them.

Here is what a sitemap.xml can look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
 <url>
   <loc>https://www.example.com/foo.html</loc>
   <lastmod>2022-06-04</lastmod>
 </url>
  <url>
   <loc>https://www.example.com/foo-bar.html</loc>
   <lastmod>2022-06-05</lastmod>
 </url>
</urlset>
```

💡**To know everything about sitemap files**, how to use them and generate them, please read this article I wrote, it explains everything about sitemaps file.

b) Robots.txt Configuration

The `robots.txt` file controls how search engines crawl your site.

What they do:

- Allow search engines to crawl important pages.
- Disallow crawling of sensitive or irrelevant pages (pages you need to be logged in to see, administration pages, etc.)
- Specify the location of your sitemap in the `robots.txt` file.

Example:

```
User-agent: *
Disallow: /admin/
Disallow: /login/
Sitemap: https://example.com/sitemap.xml
```

c) Meta Robots Tags

Meta robots tags control how individual pages are indexed and followed by search engines. Therefore, they have to be added in the <head> HTML element.

Common Directives:

- `index, noindex`: Whether the page should be indexed.
- `follow, nofollow`: Whether links on the page should be followed.

Example:

```html
<head>
    <!-- Prevent indexing but allow following links -->
    <meta name="robots" content="noindex, follow">
</head>
```

Notes:

This tag is far from being mandatory. Furthermore, most JavaScript frameworks only contain one html file (index.html) which makes it harder to add/remove this meta tag.

You can just ignore it.

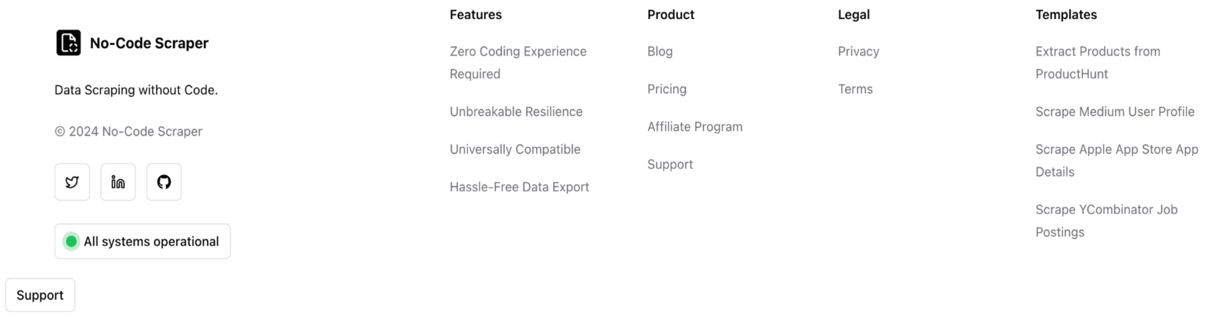
9 - Internal Linking and Navigation

a) Best Practices for Internal Links

Internal links help distribute link equity throughout your site and improve crawlability. Your internal links have to be reachable within a few clicks from the homepage (avoid having internal pages without any link pointing toward it, if you have some, add them to your sitemap.xml file).

Listing your most important links in your footer is a good way to have a strong internal linking strategy.

Here is, for example, a good footer with internal linking:



No-Code Scraper

Data Scraping without Code.

© 2024 No-Code Scraper

● All systems operational

Support

Features

Zero Coding Experience Required

Unbreakable Resilience

Universally Compatible

Hassle-Free Data Export

Product

Blog

Pricing

Affiliate Program

Support

Legal

Privacy

Terms

Templates

Extract Products from ProductHunt

Scrape Medium User Profile

Scrape Apple App Store App Details

Scrape YCombinator Job Postings

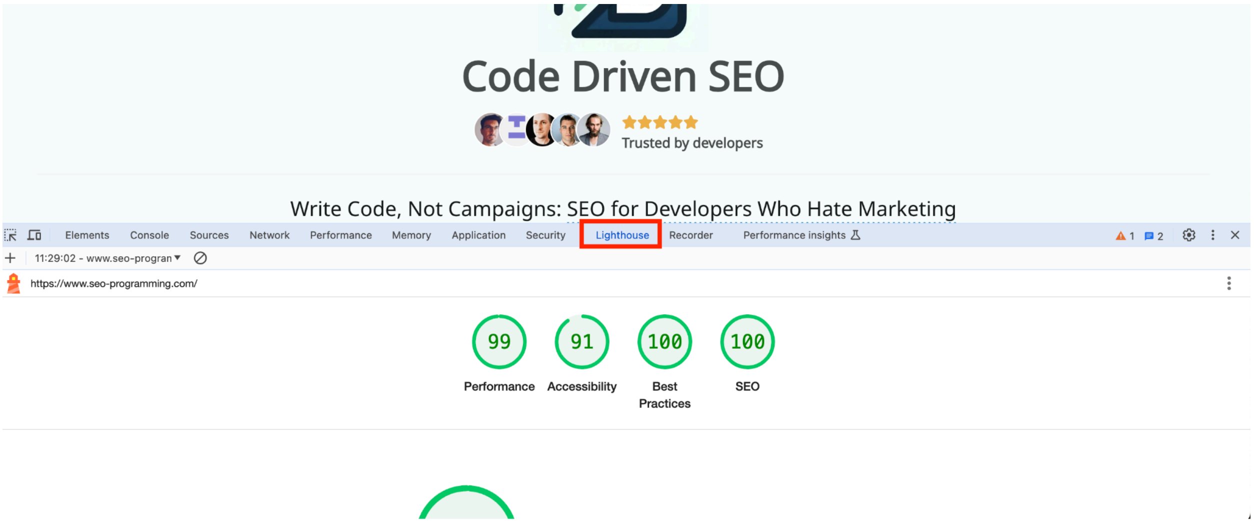
10 - Technical SEO Tools

Now that we've seen most of the technical SEO techniques, it's time to list some useful tools that will help you improve it!

a) Google Lighthouse

Well known by developers, Google Lighthouse is an open-source tool for auditing the performance, accessibility, SEO, and more of web pages.

Accessible through Chrome developer tools, Lighthouse gives you scores on your SEO, accessibility, best practices and performances:



Code Driven SEO

Trusted by developers

Write Code, Not Campaigns: SEO for Developers Who Hate Marketing

11:29:02 - www.seo-progran

https://www.seo-programming.com/

Tips:

- If you use the Lighthouse extension with developer tools, do it in a private navigation window, as some Chrome extensions/features can reduce your website's performance.

b) Google Search Console

Google Search Console helps monitor and maintain your site’s presence in Google Search results.

It is mainly used to watch your statistics, about the number of impressions, clicks, and requests that lead to these clicks.

As explained in the article about sitemap.xml I shared in the corresponding part of this guide, you can submit your sitemap.xml file to Google via the console.

You can also check the crawling status of each of your pages, and individually trigger the crawling if your page has changed/had some crawling issues.

c) Bing Webmaster Tools

Similar to Google Search Console, Bing Webmaster Tools proposes the same features as Google's one: statistics, crawling triggering, sitemap submission, etc.

d) Other Useful SEO Tools

Here is a non exhaustive list of other SEO tools you might find useful:

- Meta Explorer: a free Chrome extension to have reports and advices about a webpage SEO meta tags
- Sprout SEO: a free web browser extension that gives you very detailed data about a webpage SEO
- SpeedMeter: a free Chrome extension to know which requests slow down your website
- Merkle: a helpful list of tools to help you improve your SEO

11 - JavaScript SEO & SPA

a) Server-Side Rendering (SSR) vs Client-Side Rendering (CSR)

In the world of SPA (Single Page Application - which is probably the type of website you work on if you use any JavaScript framework), we separate two main types of rendering: **Server-Side Rendering** (SSR) and **Client-Side Rendering** (CSR).

Best examples of these technologies are **React**, which uses CSR by default, and **Next**, which uses SSR by default. Or **Vue** and **Nuxt**.

What are the impacts on SEO?

While CSR renders all the content on the client side, using JavaScript, the SSR, as its name explains, creates - renders - the whole page on the server side *before* showing it on the web browser.

Therefore, as the page is already created when displayed on the user's screen, it is better for SEO - the search engine bot doesn't have to wait for the JavaScript to load the page to fully crawl it (each crawl has a limited time - Google won't spend five minutes on your webpage waiting for it to load).

That being said, it does not mean a website using only CSR won't rank on Google.

One of my own websites, a simple landing page, makes more than 200 clicks per day without SSR.

So just be aware that SSR has a slight advantage over CSR, but is not mandatory to rank well on search engines.

12 - Last but not Least: Analytics and Monitoring

Monitoring your SEO performance is essential in your quest of a better ranking on SERP. Use the tools already mentioned to track your statistics and to know which keywords work and which don't (and maybe replace the ones that don't work with the ones that do).

You could think of using tools such as Plausible (a Google Analytics competitor) to have a better idea of where your traffic comes from.

To summarize: improving your website SEO with technical elements is mandatory, but tracking the result and adapting your websites is essential too!

Part Two: Programmatic SEO

Now that we've said almost everything about technical SEO, it's time to talk about the other SEO strategy you can, as a developer, easily implement: **programmatic SEO**.

I won't talk that much about it here, as I already created 2 external resources that explain everything:

- An article explaining everything about the programmatic SEO principle
- An online guide + all code provided for **React**, **Angular** and **Vue** to implement it in half a day and have incredible results in a few weeks

But let me summarize it.

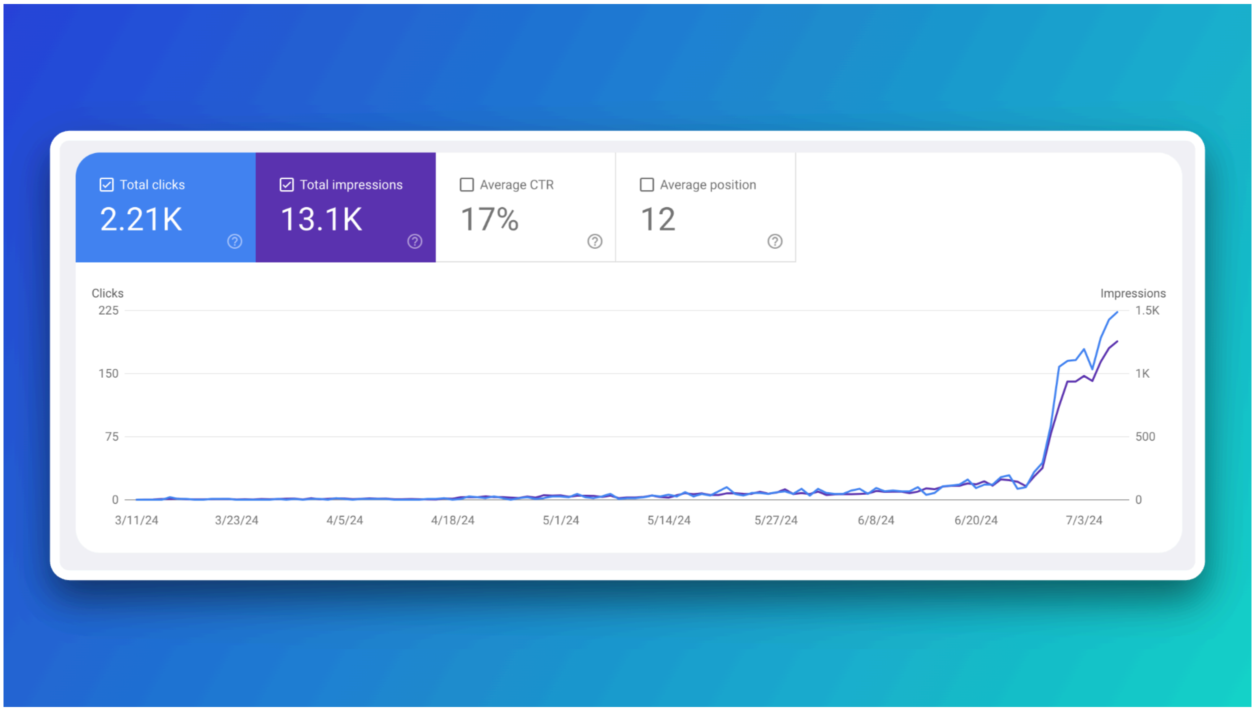
Programmatic SEO allows you to easily create different pages with code, and code only. Basically, you have one or several templates (a view component), and through code, you generate as many pages as you want from this template, by only changing the content - but also the metadata - of it.

With this technique, you can generate 2 or 3 pages with the same template, or more than 30.000 pages. There is no limit.

Of course, if you have to be careful about the dynamic content of your programmatically generated pages.

If you have 30.000 pages with poor content that doesn't match any search, you'll have less visitors than if you create 10 pages with PSEO, but with a high content quality.

The proof of this is my website **AI Photo Forge**, that only has 18 pages, created with programmatic SEO, and that reaches now more than 200 clicks a day:



☑ Total clicks 2.21K

☑ Total impressions 13.1K

□ Average CTR 17%

□ Average position 12

this result is 100% due to my programmatic SEO implementation.

Again, I really advise you to take a look at my article: Programmatic SEO: How to Boost Your Rankings on Google With a Few Lines of Code

Everything is explained there.

Want to go further?

Do you want to go further with technical SEO, and programmatic SEO?

Here is a list of the tools I have created that could speed up your SEO:

Code Driven SEO: the Ultimate Guide to Programmatic SEO

If you want to implement programmatic SEO on your project and, if you do it well, obtain the same result as me, this is the product you need.

Along with this online guide, I provide access to a GitHub repository containing all the code required to successfully use programmatic SEO, for Angular, React and Vue.

For a very fair price, you can see your number of visitors skyrocket quite quickly.

Click here to find the programmatic SEO guide.

Rank on Bing: Rank on Bing Within a Few Days

You might not know it, but Bing represents around 8% of the search engine market share in the US.

8% is an important figure, too important for us to not care about Bing.

Rank on Bing is a SaaS I created to help you rank on Bing.

How does it work? Once your websites are added to your Rank on Bing dashboard, our bots will automatically send your pages to Bing to force the search engine to crawl them, and eventually index them.

Click here to find Rank on Bing, and get 20% off by adding the code **GUIDE20** at the checkout.

SpeedMeter: Test Your Website Velocity in a Minute

As we stated in the guide, having a website offering good performance is important for your SEO.

But testing the velocity of your website is not easy. Lighthouse allows you to do that, but the results are not that easy to read.

This is why I created this free Chrome extension. Once installed, just click on a button to analyze all the web requests your website sends (external, but also images, css files, etc.) sorted by response time. This way, you can easily find out what slows down your website.

The extension is free, **click here to download it**.

AI Sitemap.xml Generator: Generate a Sitemap From Your Code With AI

Having an updated sitemap.xml file is important so the search engines bots crawl all your pages.

This means that each time you add a page, you have to add it to your sitemap file, and making a mistake, or forgetting a page, is easy.

This is why I created this free tool: an online sitemap.xml generator, powered by AI, directly from your code.

Just paste the code of your router file (it supports every JavaScript framework), and receive your sitemap file already formatted by email.

Simple and free! **Click here to have access to the tool**.

Indie Dev Tools: a Directory of Tools for Indie Developers

My last service related to SEO is a directory: Indie Dev Tools.

This website lists all the tools an independent developer (or indie hacker) might need to succeed on their projects.

There are all kinds of tools, from marketing to customer service, and, of course, SEO.

You can even add your own tool, for free.

Find **Indie Dev Tools here**.

+120 SaaS Ideas to Start Your Business Today

And if you're running out of ideas for your next project or SaaS, here is a product of mine that can help you: a list of more than 120 SaaS ideas to start your business today.

Each idea comes with: its target audience, keywords, competitor names, an USP (unique selling proposition), and technologies to use.

Here is the link to the SaaS ideas.

And, that's it!

I hope you found this guide useful. If you did, please don't hesitate to share it, you know you're not the only one that needs it.

Just before letting you go, if you're interested about me, who I am and my work, here few information:

About me

I quickly introduced myself in the introduction. But let me go a bit further, share my work, and a few links that could interest you.

I'm Alex, a web and mobile developer since 2013. I've been an employee for the first 5 years, and then became a freelancer.

Even when I was still a student, I was already creating side projects.

In 2023, I realized that working for clients had become painful, and that I should focus on my side business to make enough money with them - enough not to have to work for clients ever again.

Since then, I've created more than 10 products. You can find most of them on my Linktree page. These projects can be an AI Suit generator, an app to find allergens in food, or a developer tools directory. Very varied, as you can see!

But most of them are about SEO. The reason is simple: it's because I hate marketing, and SEO was the best way I found to generate a stable traffic, and therefore a stable income, without having to speak with people.

I also have **a Medium blog**, on which I share tips and advice about **programmation**, **side projects** and **entrepreneurship**.

Any questions about SEO, my products, or a collaboration? Please reach me out on Telegram or by email at pro@alexandre-grisey.fr!