

# Web performance and quality report



https://xarta.co.uk

This report is provided by DareBoost, an online tool for web performance and quality analysis and monitoring.

Don't hesitate to check out our offers or to contact us : contact@dareboost.com

•

# **Summary**











SIMULATED VISITOR:



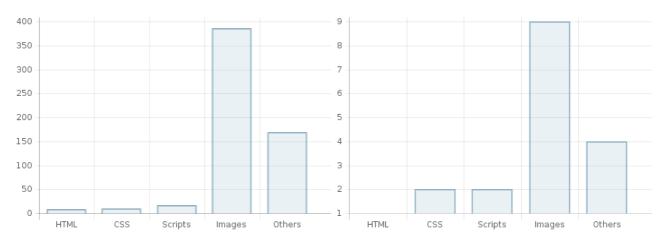


London 10.0/2.0Mbps (Latency: 28 ms)

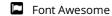


# Weight by resources type

# Requests by resources type



# **Detected technologies**





Google Analytics



Windows Server



# Tips and best practices

Accessibility



# Well done, these best practices are respected

# 100/100

# <noscript> tag detected

#72

This page uses <code>noscript</code> tag. It allows to display a message when JavaScript is disabled by the user.

# 100/100

# ▼ The main title of the page is the first stated title

#2468

Facilitate the work of screen readers by indicating your main title ( <hl> ) before any other title.

We recommend you to keep a coherent hierarchy among your titles (h1, then h2, then h3, etc).

# 100/100

#### ✓ No i> element is empty

#2525

Users should identify easily the number of items in a list. An empty element provides a wrong information about the list content.

The <i> element should not be empty. You must specify a text, or an image with an alternative text to allow all your users to be informed of the contents listed.

#### Example:

```
my first element
```

You should prefer the use of CSS to apply a layout.

# 100/100

# This page defines a lang

#2334

Perfect. Your page defines a lang attribute that will allow screen readers to correctly understand your website.

# 100/100

# ✓ No 'refresh' <meta> tag

#2395

It is not recommended to automatically force a page refresh. This behavior disturbs users who are not able to control it.

This page does not use meta refresh tag. This is a best practice.

# 100/100

# **▼** You specify a consistent label on your links

#2423

A link is more attractive if the text describes what is behind it. You can also take the opportunity to use keywords in these texts, to improve your page's SEO.

Describe the link in your <a> tag, rather than indicating the link itself. Example: <a href="http://mylink.com/">My description</a>

The content of your <a> tags are different from the link itself. That's a best practice.



# Did you know?

# #50 The Slowest Resources

The page load time is 19.4 seconds. Listed below, are your site's slowest loading requests:

- https://xarta.co.uk/ (225ms)
- https://xarta.co.uk/sound/whoop.mp3 (217ms)
- https://xarta.co.uk/js/homepage-15cc61f464f.js (86ms)
- https://xarta.co.uk/js/defer-15cc61f464f.js (74ms)
- https://xarta.co.uk/favicon-32x32.png (68ms)
- https://xarta.co.uk/favicon-16x16.png (62ms)
- https://xarta.co.uk/css/styles-15cc61f464f.css (58ms)

#### Listed below, are your site's 10 slowest requests:

- https://cdn.jsdelivr.net/threejs/0.85.2/three.min.js (3641ms)
- res.cloudinary.com/xarta/image/upload/v[...]500/xarta/moon.png (319ms)
- res.cloudinary.com/xarta/image/upload/v[...]/xarta/8pUBeuw.jpg (302ms)
- maxcdn.bootstrapcdn.com/font-awesome/4.[...]font.woff2?v=4.7.0 (165ms)
- maxcdn.bootstrapcdn.com/font-awesome/4.[...]nt-awesome.min.css (122ms)
- res.cloudinary.com/xarta/image/upload/v[...]/spiral-galaxy.jpg (107ms)
   res.cloudinary.com/xarta/image/upload/v[...]/xarta/fB9BaJJ.jpg (100ms)
- res.cloudinary.com/xarta/image/upload/v[...]/xarta/Eu0MQrk.png (73ms)
- res.cloudinary.com/xarta/image/upload/v[...]-me-at-work256.jpg (46ms)
- res.cloudinary.com/xarta/image/upload/v[...]/xarta/KmwuMPu.jpg (40ms)

Excessively slow response times can be due to any number of reasons, usually because of a heavy resource that's loading or a temporary network issue.

Well done, these best practices are respected



# ▼ You don't execute the same script several times

#2445

A library or an external script are usually intended to be called once per page. However, the use of widgets can lead to several useless executions.

#### **Duplicate scripts**

It is common to see scripts used multiple times within the same page. The most common case is the integration of social networks widgets. It can be useful to find them several times on the page. That is not a bad practice, however you should be careful that those scripts don't slow down your page.

#### What happens when a script is included 2 times in the code? How many times is it loaded? Parsed? Executed?

Most modern web browsers download only once a script included 2 times. An exception persists: Firefox, which will load the resource as many times as mentioned if no effective caching policy is configured.

Apart from this exception, performance issues come during the parsing and execution of the scripts. Indeed, if a script is placed three times in the code, it will be parsed and executed 3 times, on all browsers.

Do not hesitate to read this article on that topic.

#### How to fix it?

There is a solution to use a script several times without parsing and executing more than once. You need to write some JavaScript code that checks if the script is present. If the script is already included, it just uses it, otherwise it injects it and uses it.

Consider the following example with the Facebook widget, described in the article. Whenever you want to integrate this functionality into your page, it is necessary to include the following code:

```
(function(d, s, id){
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) {return;}
  js = d.createElement(s); js.id = id;
  js.src = "//connect.facebook.net/en_US/sdk.js";
  fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'facebook-jssdk'));
```

The bold line checks the presence of the script in the document. So the script is included, parsed and executed only during the first call in the page. Other calls will fall in the case of the bold line, and therefore will just use the script that is already included and executed.

# 100/100

# ✓ Your scripts seem to be injected efficiently

#2460

The script injection through the document.write instruction delays the rendering of your page and/or the interactivity for the user.

#### Do you need to inject a script?

document.write is sometimes used to inject a script. That is a bad practice. Example:

```
document.write('<script src="" + src + "" type="text/javascript"><\/script>');
```

As described in this article, you should prefer the "createElement-insertBefore" pattern:

```
var sNew = document.createElement("script");
sNew.async = true;
sNew.src = "http://ajax.googleapis.com/ajax/libs/jquery/1.5.1/jquery.min.js";
var s0 = document.getElementsByTagName('script')[0];
s0.parentNode.insertBefore(sNew, s0);
```

On this page, no document.write injects a script.

# 100/100

# ▼ The page don't use client-side redirection

#2539

No client-side redirection (window-location, meta refresh...) has been triggered to display the page. Client-side redirections are to be avoided in any way possible.



# ✓ No detected SPOF on this page

#2531

The failure of a third-party content provider could bring an overall breakdown of your website.

A Frontend Single Point Of Failure (SPOF) is a critical dependency on a third-party content, that may block the entire display of your page in case of failure of the content provider.

As an example, if your web page uses a blocking script hosted by Google's servers, then your page is reliant on any failure from this script. Please read our blog post dedicated to SPOF for more information.

#### How to avoid SPOF?

As far as possible, exclude any of these dependencies, even from renowned providers. If you have to use a third-party content, ensure that you choosed an asynchronous integration and that you have a fallback in case of problem.

We are checking if the tested web page depends (in a critical way) on some of most widespread external resources (googleapis, typekit,...). That are known as Frontend SPOF (Single Point Of Failure) cases.

We did not detect any SPOF in this web page.

# 100/100

# ▼ Your JavaScript resources don't block your page loading

#2353

JavaScript can significantly slow down a page display, especially if it is necessary to download an external script.

Defer the use of JavaScript as much as possible to provide a faster start for the page display.

#### How can I fix this?

Use one of the methods below to defer parsing for external JavaScript files:

- use the async attribute;
- use the defer attribute;
- append the script to the DOM in JavaScript during the onload event;
- make sure your scripts are placed at the bottom of the page (ideally at the end of the body).

# 100/100

#### You specify a character set

#2356

Specify the character set used in the Content-Type HTTP header allows the browser to parse immediately the page.

# 100/100

# ✓ You do not use CSS @import

#2370

Using CSS @import allows to add external stylesheet. In fact browsers cannot download them at the same time, this may add a delay to the rendering of the page. It is better to use the link tag. See more information.



# ▼ You do not use a useless "http-equiv" <meta> tag

#2405

HTTP headers are more efficient than the http-equiv meta tags.

#### The <meta http-equiv=""/> tags

The <a href="http-equiv">http-equiv</a> meta tags allow to communicate to the web browser information equivalent to the ones of HTTP headers. For example, the meta <meta http-equiv="content-type"/> will have the same consequences than the HTTP Content-Type header.

Two points don't stimulate the use of http-equiv meta tags:

- Going through the meta requires to interpret the beginning of the HTML page, which is slower than going through the HTTP headers in terms of performance
- If the HTTP header is already present, the meta is ignored

#### In which cases are the <meta http-equiv=""/> useful?

Only one case can justify the presence of these meta tags: if you don't have access to the configuration of your server, and that is to say to the HTTP headers.

However, we advice you to use a configurable server so that you can establish the most efficient site possible.

This page does not contain a http-equiv meta tag. That's a best practice.

# 100/100

# ▼ This page uses an appropriate number of DOM elements

#2416

The number of DOM elements influences the complexity of the webpage and DOM access in JavaScript.

A well-designed webpage can offer rich content while maintaining a reasonable number of DOM elements. Read more about this here.

We recommend creating pages that contain less than 1000 DOM elements.

There are 127 DOM elements on this page.



# Well done, these best practices are respected

# 100/100

# You do not use too long inline scripts

#70

Any script with a significant size should let the browser cached them in order to reduce loading time/improve performance of your returning visitor.

#### Inline scripts / cache policy

"inline" scripts allow to integrate easily small portions of scripts directly in the HTML code. Example:

By doing so, you avoid making a request to the server to retrieve the resource. So inline scripts represent a performance gain if you want to integrate small scripts.

However, once a script has a fairly substantial size, we advise you to outsource it and perform a request to retrieve it. So you will benefit from the cache mechanism.

#### What should I do?

Outsource your scripts with more than 1500 characters in one or more separate files.

# 100/100

# ▼ The cache duration is greater than 48 hours

#2437

Defining several days of cache retention for your static resources will reduce the load on your server.

## The Expires header explained

Some of your resources use the Expires HTTP header to get an effective caching policy—this is a best practice. However, you should consider improving its configuration to make the most of the caching mechanisms. Here is an example of the Expires HTTP header:

Expires: Thu, 25 Dec 2014 20:00:00 GMT



When you **deploy a new version of your website, remember to rename static resources** that have been modified. If you do not change their names, your users will keep resources corresponding to the old versions stored in their caches, and they may find themselves on an unstable version of your page. For example:

myresource.min.20140101.js

Example

Read the Yahoo! guidelines on this subject.

#### **Recommended Expires header setting**

We recommend setting the **Expires** HTTP header so the date is between 2 days and 1 year.

Your resources are cached at least 2 days.

# 100/100

#### ▼ The 'Vary: Accept-Encoding' header is defined

#2352

The Vary: Accept-Encoding header allows to cache two versions of the resource on proxies: one compressed, and one uncompressed. So, the clients who cannot properly decompress the files are able to access your page via a proxy, using the uncompressed version. The other users will get the compressed version.



# ▼ You specify 'Expires' headers

#2430

The Expires header is essential for an efficient caching policy. It will significantly impact on the loading time for returning

#### The Expires header explained

You can set an expiration date for each resource: as long as the date is not exceeded, the browser stores and uses the resource in cache.

The expiry date of resources is set using the **Expires** HTTP header:

Expires: Thu, 25 Dec 2014 20:00:00 GMT

You can set a far expiry date for static resources (1 year maximum), and a closer date for resources that change more frequently (at least 48 hours).

When you deploy a new version of your website, remember to rename static resources that have been modified. If you do not change their names, your users will keep resources corresponding to the old versions stored in their caches, and they may find themselves on an unstable version of your page. For example:

myresource.min.20140101.js

See the Yahoo! guidelines on this subject.

Congratulations! Your resources are cached thank's to the 'Expires' header.

# Well done, these best practices are respected

# 100/100

# ✓ No frameset, frame and noframes tags detected

#80

These tags are obsolete, due to several issues related to the navigation consistency, SEO or browsers' bookmark features

None of these tags is detected on this page.

The use of the iframe tag is prefered.

# 100/100

# No Java applets detected

#82

Java applets are considered obsolete in 2015. HTML5 is powerful and more widely supported. Using Java applets can lead to compatibility issues and may send negative signals to your users (eg the browser indicating that content was blocked because it could be dangerous).

Congratulations, this page doesn't contain Java applets.

# 100/100

# ✓ Do not use <br/>bgsound> tag

#89

No bgsound tag detected. This is a good practice: this element is not a HTML standard. See more information.

Use the audio tag to deliver audio content on your page.

<audio src="my-audio-file.ogg" autoplay> Your browser doesn't support the audio element. </audio>

# 100/100

# ✓ You do not use links to Word documents

.doc and .docx documents do not garantee compatibility with all major operating systems. It's recommended to use PDF documents.

This page contains only standard links.

#### 100/100

# Your characters are encoded in UTF-8

#2454

Your content is readable by the largest number of web users.

# **Encoding of characters**

The encoding of characters indicates to the web browser how to interpret the bytes of the web page in order to convert them in readable characters by the user. Lots of encodings exist so that all the characters specific to each language are represented. Be aware that the supported groups of encoding are different from an engine to another one. If you ever use an encoding too specific, a lot of web users won't be able to accurately interpret the page.

# Why choosing the UTF-8?

The UTF-8 is known for being an encoding supported by almost every web user, and taking into account a large range of characters. Its universality is then particularly well suited to the web environment.



# Silverlight plugin is not used

#2493

Silverlight plugin is considered obsolete in 2015. HTML5 is more powerful and more widely supported. Using Silverlight plugin can lead to compatibility issues and may send negative signals to your users (eg the browser indicating that content was blocked because it could be dangerous).

Congratulations, this page doesn't use Silverlight plugin.

100/100

# ▼ The EOT font format is well declared

#2494

Make sure that the EOT format will be well used for your visitors using Internet Explorer <= 8.

#### Use the EOT format for Internet Explorer <= 8

The web browsers (Firefox, Chrome, Safari, Internet Explorer, Opera ...) do not support the same font formats. Your goal is to use the lightest possible format for each platform. When you use a font, you can specify several formats, via the @fontface CSS instruction: so the browser will use the first supported format.

This is true for all browsers except Internet Explorer <= 8, that misinterpret the @font-face instruction: it will use only the first format declared. And the lighter font format for this kind of browser is the EOT format.

#### How to use EOT format only for IE <= 8?

Only Internet Explorer supports the EOT format. So declare it in first position in the @font-face 's src property: Internet Explorer will be able to understand that it must use the EOT format, and other browsers, that do not support EOT, will use another format. Here is an article about this topic to help you to determine the right order to declare your fonts. TL; DR order to respect for your webfont formats: EOT, WOFF2, WOFF, TTF, and SVG.

All fonts with EOT format are placed in first position in the @font-face instruction.

100/100

#### Your font formats are well used

#2496

Be sure to deliver the most suitable font format for each platform.

# **Declaring several font formats**

The web browsers (Firefox, Chrome, Safari, Internet Explorer, Opera ...) do not support all the same font formats. Your goal is to use the lightest possible format for each platform. When you use a font, you can specify several formats, via the @font-face CSS instruction: so the browser will use the first compatible format that it supports.

# How to declare your font formats?

Check if your formats are well ordered in the offont-face 's src property. Here's an article about this topic to help you to determine the right order to declare your fonts. TL; DR order to respect for your webfont formats: EOT, WOFF2, WOFF, TTF, and SVG.

All fonts are well ordered in the <a>@font-face</a> instructions.

100/100

# ✓ No deprecated attributes are detected in the <body> tag

#2382

Some layout attributes are deprecated in HTML 5: alink / background / bgcolor / link / text / vlink

Congratulations, this page doesn't use deprecated attributes in the body tag.

Prefer using CSS instructions instead.



# ✓ No BOM (Byte Order Mark) detected

#2424

Some parsers are not able to interpret a page with a BOM in it.

#### What is the BOM?

The BOM is a hidden character located in the beginning of the page, aiming at helping to determine what encoding the page uses. But the best practices of the web prompt the use of the HTTP Content-Type header in order to define the encoding used by the page. The  $\operatorname{\mathsf{BOM}}$  has no reason to be in this context.

Aside the fact that this kind of indicator is useless on the web, it can lead to a certain number of issues. This is the case for example of the W3C validation that is going to try to interpret the first character which corresponds to the BOM. Then, the document will not be valid.

No resource use a BOM.



# **Your Optimization Priorities**

# 99/100

#### Optimize your images

#2389

Properly formatting and compressing images can save many bytes of data.

Optimize the following images to reduce their size by 1.8KiB (89% reduction).

It appears these files are hosted by a third-party, so they may not be within your control. However, you should consider any alternative to these resources to improve your page performance.

• Losslessly compressing res.cloudinary.com[...].png could save 1.8KiB (89% reduction).

Images may contain data unnecessary for their use on the web. This data can increase their size significantly. Some tools automatically remove this unnecessary data without loss of quality and thus reduce your image sizes.

We recommend removing unnecessary image data using a tool such as jpegtran (JPEG files), OptiPNG (PNG files) or ImageRecycle.

# Did you know?

#2443

# This page does not load too much data (0,59 MB)

A too high page weight slows down the display, especially on low speed connections. This can lead to frustration for users paying for data (see whatdoesmysitecost.com).

#### **Evaluate the Weight of my Web Page**

In February 2016, the average weight of 100 most visited websites in the world was 1,38MB.

#### How to reduce the weight of my page?

You can report to our "Data amount" category to discover the possible optimizations in your case. Images are often involved.

Moreover, make sure to build your web pages in order to load data that is essential to the user experience (rendering optimization of the critical path).

For other contents (social networking plugins, advertising, content at the bottom of the page ...), it is better to delay the loading (asynchronous, lazy-loading ...), so they don't override priority contents.

We have established the weight distribution of the page by resource type:

Images: 65,55% of total weight
Others: 15,50% of total weight
Font: 13,19% of total weight
JavaScript: 2,79% of total weight
CSS: 1,63% of total weight
Texts: 1,35% of total weight

Here is the weight of the 10 heaviest resources over the network, and that are necessary to load the page:

- res.cloudinary.com/xarta/image/uploa[...]0/xarta/moon.png (323 kB)
- https://xarta.co.uk/sound/whoop.mp3 (91 kB)
- maxcdn.bootstrapcdn.com/font-awesome[...]nt.woff2?v=4.7.0 (78 kB)
- res.cloudinary.com/xarta/image/uploa[...]piral-galaxy.jpg (32 kB)
- https://xarta.co.uk/js/homepage-15cc61f464f.js (13 kB)
- maxcdn.bootstrapcdn.com/font-awesome[...]-awesome.min.css (8 kB)
- https://xarta.co.uk/ (8 kB)
- res.cloudinary.com/xarta/image/uploa[...]arta/fB9BaJJ.jpg (8 kB)
- res.cloudinary.com/xarta/image/uploa[...]e-at-work256.jpg (8 kB)
- res.cloudinary.com/xarta/image/uploa[...]arta/8pUBeuw.jpg (6 kB)

Well done, these best practices are respected



# Lazyloading is set (or not required)

#2446

Load images above the foldline first, that means all the images that are visible without any scroll of the page by the visitor.

#### Webpage and images

Images use to represent more than 60% of the total weight of webpages. By loading initially the only images that are visible without page scrolling, you'll reduce bandwith consumption as for your server and your visitors. If an image has to display after any visitor's action (as scrolling), then you'd better load it on demand, when necessary (lazyloading).

#### How to set up lazyloading?

You can get informations concerning the features and plugins proposed by your page's frameworks/CMS. You should find some easy-to-set solutions. Some Javascript libraries could help you to implement lazyloading too.

# 100/100

# ✓ You serve scaled images

#2436

Images must not be delivered larger than they are actually displayed to avoid loading unnecessary data.

Using the width and height attributes in img tags is not recommended to reduce the rendering of an image.

For instance, if your image is set to render at 300px by 300px on a particular page, don't upload the original 1000px by 1000px version of that image to your page. Instead, resize/crop the image to fit the display size and then upload it to your site to decrease the page weight and loading time.

#### Using images with responsive designs or retina screens?

Responsive website designs and retina screens do not justify an image resizing. Even in such cases, some methods exist to deliver your pictures to the right size. We recommend reading the following resources:

- Introduction to responsive images
- Picturefill, to start using the <picture> element
- RICG, group of developers working on responsive images

Good job! No image resize have been detected. This is ideal for reducing page weight and loading time.

# 100/100

# 4 images use the PNG format in the right way

#2461

The choice of the right format for an image allows to reduce its weight.

#### The PNG format

The PNG image format is intended to the images requiring the transparency, or else to the small images having little details and colours.

#### Prefer the JPEG format...

The main problem of PNG format is to not support quality loss. Indeed, a format such as JPEG offers to "downgrade" the quality of the image without being perceived by the user. Doing so, you can reduce the quality of the image of about 25% whithout the user realizes it.

An image with an consequent weight will be therefore better compressed by using the JPEG format.

#### ...or the PNG-8 format

In the case where your image necessarly requires using the transparency mecanism, not borne by the JPEG format, you should convert your "standard" PNG image, into PNG-8. This format, based upon a 256 color palette maximum allows to decrease the weight of the image whithout significantly affecting its overall quality. Tools such as pngquant or else tinypng will suport you in this procedure.

At last, if the quality given by the PNG-8 format is not appropriate for your image, you can get information on the possible advanced technics to get this behaviour without having an image in PNG format. For example, it is possible to halve your image in 2 JPEG images, one including the transparency data and the other one including the data related to the colours, and to gather the image on the customer's side with a CANVAS element.

For further information, please visit this article about image compression.

The choice of the PNG format on the images of this page doesn't have negative effect.



# ▼ This page do not send cookies > 100kB

#2450

Keep the size of cookies as low as possible to minimize the impact on the loading time.

HTTP cookies are used to track a user to costumize the page according to their profile. They are sent as a HTTP header from the web server to the browser. Then, each time the browser accesses to the server, it sends a request containing the cookie received at the first response. See more information.

# 100/100

# This page does not send too many cookies

#2453

Keep the size of cookies as low as possible to minimize the impact on the loading time.

HTTP cookies are used to track a user to costumize the page according to their profile. They are sent as a HTTP header from the web server to the browser. Then, each time the browser accesses to the server, it sends a request containing the cookie received at the first response. See more information.

# 100/100

# ▼ The favicon is lightweight

#2490

The favicon must be as lightweight as possible.

#### The favicon on the web

This small image allows you to display a logo on the browser tabs for instance.

#### How to fix the issue?

A favicon should be < 10kB.

The favicon isn't too heavy (2kB).

# 100/100

# ✓ Redirects are lightweight

#2486

A HTTP redirect's content is not used by web browsers. So, it provides an unnecessary weight that should be as small as possible.

# Redirects on the web

Redirects can be temporary (302 HTTP code) or permanent (301).

This mechanism is for example used for secondary domains (.net to .com), the language detection mechanisms, etc.

# How to fix the issue?

First ensure that the redirect is unavoidable. In this case, it is necessary to reduce or completely remove the contents of the server responses for redirects.

A redirect should be < 1kB.

There isn't too heavy redirect.



# HTTP error responses are lightweight

#2487

The content of a HTTP error response is not used by web browsers. So, it provides an unnecessary weight that should be as small as possible.

#### Responses in error on the web

The resources that can be reached by the browser return a HTTP code between 400 and 599.

#### How to fix the issue?

First, you have to avoid to request resources that return an error. Moreover, reduce or completely remove the contents of the server error responses.

A response error should be < 10kB.

There isn't too heavy resources in error.

# 100/100

# All your resources are served from a consistent URL

#2325

Resources with identical content should be served from the same URL to avoid duplicate downloads and additional requests.

#### Different URLs for the same resource

If you use different URLs to serve identical content, browsers have to request this content from the server several times, and your page will trigger unnecessary requests for data. If the same URL is used, the browser will only send one request and will use the response wherever the resource is requested. You should also use the same URL to reference the same content on several pages to benefit from the cache mechanism.

#### How to fix this?

Serve each resource from a unique URL. If you have to request the same resource several times, the same URL should be used to retrieve it every time.

# 100/100

# You do not need to minify the HTML resources

#2364

Your HTML is minified. Learn more about minifying HTML.

You get 100/100, while your resources are not minified? It means that we considered that the gains provided by the minification process were not significant enough to be reported.

#### 100/100

# Your JavaScript resources are minified

#2388

Your JavaScript content is minified. Learn more about minifying JavaScript.

There are many tools to minify JavaScript files. You can try YUI Compressor or JSMin, recommended by Google.

# 100/100

#### ✓ Your CSS resources are minified

#2387

Your CSS is minified. Learn more about minifying CSS.

There are many tools to minify CSS files. You can try YUI Compressor or cssmin.js, recommended by Google.

If you want to know more about the CSS minification, do not hesitate to read our article on the subject!



Compression is enabled

#2384

You have compression enabled. Learn more about enabling compression.

100/100

No cookies are sent on static resources

#2414

You send some unnecessary data on the network through cookies on static resources.

#### Cookies and static resources

Cookies are used to customize the content of a page depending on the user profile. There is no need to send this kind of information when requesting resources that deliver always the same content (static resources).

The requests to images, JavaScript or CSS resources do not need to use cookies.

#### How to fix the issue?

Do not use cookies on static resources requests. However, some domain names require the sending of cookies on all resources. In that case, you can create an other domain name that will deliver your static resources. Example: static.mydomain.com

100/100

✓ All your resources have a size < 1MB</p>

#2421

Too heavy files should not be loaded on a web page. Be sure you need this resource to load the page. You should probably consider some of the following actions:



# Did you know?

#2466

# 1 image uses only one color

One request to an image composed of a unique color unnecessarily increases the data amount transmitted on the network.

It is useless to retrieve an image composed of a unique color. The CSS style sheets allow to render the same result at a lower cost.

For instance, to draw a simple circle, you can use the following code :

```
#myElement {
background:#ff0000;
border-radius:50%;
width:160px;
height:160px;
```

1 image of this page is composed of a unique color:

It appears these files are hosted by a third-party, so they may not be within your control. However, you should consider any alternative to these resources to improve your page performance.

• res.cloudinary.com/xarta/image/uploa[...]arta/Eu0MQrk.png (2kB)

#### #2543

#### Resources distribution by domain

This page loads data from 5 domains. This best practice retrieves the following metrics for each of these domains:

- Loading Time (Cumulative): total time spent to load all the resources
- Server Time (Cumulative): total time spent to retrieve the responses from the server (TCP connection + wait for first byte)
- Weight: data amount loaded
- Number of requests

Here is the list of all the domains used by the page:

Domain	Time (ms)	Server Time (ms)	Weight (kB)	Requests
xarta.co.uk	790	605	121	7
res.cloudinary.com	987	425	382	7
maxcdn.bootstrapcdn.com	287	158	86	2
www.google-analytics.com	0	0	0	1
cdn.jsdelivr.net	3641	3604	0	1



# Well done, these best practices are respected

# 100/100

# **Ⅵ** No redirect detected

#2339

The redirects trigger avoidable roundtrips on the network and increase the page loading time.

#### **HTTP** redirects

The HTTP redirects allow to specify that the desired content is accessible from a different URL. They trigger a new HTTP request to retrieve the target resource and return a HTTP code between 300 and 399. See the specifications of HTTP

#### How to solve the issue?

Allow the user to directly access your content without redirects, or determine and improve what causes these excessive loading times on your redirects. See recommendations from Google.

This page does not use redirects before accessing the right content.

# 100/100

# Only reachable resources are requested

#2344

You should avoid requesting unreachable resources.

All of the page's resources are reachable.

These errors can affect content on and behaviors of your website and cause unnecessary network traffic, which affects the loading time of your page.

# 100/100

# ▼ You do not use too much "prefetch" on the links

#2403

No resource is prefetched on this page.

You should limit the number of prefetched requests to avoid a network overload. Identify your users behaviour to prefetch the most requested resources.

Note that some browsers automatically block the number of prefetched resources. For instance, Internet Explorer 11 limits this mechanism to 10 requests.



# **Your Optimization Priorities**

# 95/100

#2511

# 1 CSS selector is duplicated

Using several times the same selector for several declarative blocks within a single CSS file can affect the readability and maintainability of the code. It is also an optimization opportunity: by grouping these elements within a single rule, you will reduce the file size.

#### **CSS Rules**

The CSS rules allow to select elements from the HTML code in order to apply styling properties.

You just have to group the same selectors into the same rule. For instance, the following rules:

```
.myClass {
 margin: 0;
}
. my Class \{\\
 border: 1px solid black;
```

Should be merged into a single one:

```
.myClass {
margin: 0;
 border: 1px solid black;
```

If the duplication is done on purpose (your own architecture, etc.), you can consider using a CSS minification tool able to do automatically this improvement.

You can save 30 characters. The following files define the same selector in different rules:

https://xarta.co.uk/ (inline 0)

• .no-js h1, .no-js h2, .no-js p (2 times)



# 1 CSS property is duplicated

#2527

Using several times the same property within a same CSS rule can affect the readability of the CSS. It is also an optimization opportunity: by removing duplicated properties, you will reduce the file size.

#### **CSS** properties

The CSS properties allow to apply a style to a set of elements. It is unnecessary to define 2 times the same property with the same value in a same rule.

#### How to improve it?

Remove one occurrence of the duplicated property. For example, the following properties:

```
.myClass {
margin: 10px;
margin: 10px;
```

Should be replaced by:

```
.myClass {
margin: 10px;
}
```

The following files define the same property several times (with the same value) in a single rule:

https://xarta.co.uk/css/styles-15cc61f464f.css

• .drawer>li ul>li {float: right} (line 1, col 2704)

# 92/100

# Separate the CSS styles from the HTML tags

#2383

Separating HTML tags and CSS directives improves code readability and promotes factorization.

#### How to define CSS styles

CSS styles are used to format the page. You can use one of three main methods to define them:

- declare styles in a specific CSS file;
- declare "inline" styles (<style> tag in your HTML template);
- declare styles with the "style" attribute of a HTML tag.

# How can I improve my page?

We recommend grouping your CSS styles in <style> tags or in separate files. That way, the HTML is only responsible for providing the structure of the page, and its layout is outsourced. The <style> attribute should only be generated by some JavaScript code (e.g., if you need to know the screen size).

This page uses 1 style attribute(s):

```
<div id="log" class="debug" style="position: absolute; display: none;</pre>
                    z-index: 100; top: 5px; left: 20px;
                    background-color: white; width: 50vw; height: 75vh">
```



# Did you know?

#### #71

### No HTML code is commented

Comments allow you to detail a portion of code and help you navigate more efficiently in the DOM. However, make sure no sensitive information is exposed in your comments.

Well done, none of your comments contains HTML code.

#### #2517

#### 1 The !important declaration is used 8 times

If you abuse of this declaration, you should consider a review of your CSS code. We tolerate 10 occurrences of the !important declaration before penalizing your score.

Here are the !important detected:

https://xarta.co.uk/ (inline 0)

- .no-js {padding: 1rem!important} (line 6, col 17)
- .no-js h1, .no-js h2, .no-js p {position: relative!important... (line 17, col 17)
- .no-js h1, .no-js h2, .no-js p {margin: inherit!important} (line 18, col 17)
- .no-js h1, .no-js h2, .no-js p {padding: inherit!important} (line 19, col 17)

#### https://xarta.co.uk/css/styles-15cc61f464f.css

- .no-js h1, .no-js h2, .no-js p {position: absolute!important... (line 1, col 196)
- .no-js h1, .no-js h2, .no-js p {border: 0!important} (line 1, col 224)
- .no-js h1, .no-js h2, .no-js p {padding: 0!important} (line 1, col 243)
- .no-js h1, .no-js h2, .no-js p {margin: 0!important} (line 1, col 263)

#### #2513

## 1 selectors are superfluous

It's often inefficient to bring too much preciseness in CSS selectors. For example, in the following rule:

```
body div .myClass {}
```

The body element provides no details to select the desired tags. So you should prefer the following selector:

```
div .myClass {}
```

In the same way, you do not need to specify a selector if you specify a behind.

Here are listed the elements considered as potentially redundant: ul li , ol li , table tr , table th , body .

The following files declar some useless selectors:

https://xarta.co.uk/css/styles-15cc61f464f.css

• .drawer>li ul>li (line 1, col 2704)

# Well done, these best practices are respected

# 100/100

# The extensions of your resources are consistent

#2448

The extension of a resource allows to identify easily its content type. You have no reason to indicate a different extension of the actual content type of the resource.

For instance, the file resource.js must use a Content-Type HTTP header equal to application/javascript . You can check the Content-Type HTTP header value using a debug tool like Firebug.



# All resources define their content type

#2449

Each resource should define its content type in order to facilitate their interpretation by web browsers.

You have no reason to hide the type of a resource.

We advise you to set the "Content-Type" HTTP header on every resources of the page.

# 100/100

#### IDs within your HTML code are unique

#2491

Using the same ID on several elements can have side effects, especially during JavaScript executions or when applying CSS rules.

#### IDs explained

Each element of a web page can be identified thanks to the id attribute:

```
<span id="mySpan1"></span>
```

These IDs allow you to manipulate your elements with CSS or JavaScript instructions.

#### How to properly use IDs?

You must ensure identifiers are not duplicated within the page. If you want to share a property or a behavior between multiple items, you have to use the class attribute, which is dedicated to this purpose:

```
<span class="mySpans"></span><span class="mySpans"></span>
```

#### 100/100

# ✓ Your CSS selectors are not too complex

#2516

Using simple CSS selectors improves your code's readability and performance.

#### CSS rules

The CSS rules allow to select elements from the HTML code, and to apply layout properties.

They use selectors (identifiers, classes, tag names, etc.) for selecting the HTML elements. A well-structured page allows writing simple CSS rules, quickly readable and efficient.

# How to improve it?

Simplify your selectors as much as possible. Do not hesitate to introduce new classes in your code to avoid complex selectors.

For instance, the following selector:

```
body td .myClass div .myClass2 span .myClass3 {}
```

Should be replaced by:

#myld .myClass3 {}



# No CSS property is overridden by a shorthand

#2515

CSS has some pitfalls that can cause side effects on your styles.

#### **Override CSS properties**

When a property is used 2 times within a CSS rule, the second one overrides the first.

CSS provides some shorthand properties, that can leads to some mistakes. For instance, the "border" property is applied to all borders of an element, avoiding the use of all "border-top", "border-right", "border-left" and "border-bottom"

For intance:

```
.myClass {
border-color:red:
border:5px solid; // border will take again the default color
```

"border-color" is overridden by the "border" shortand property.

#### **Avoid side effects**

The use of shorthand properties is risky, because it comes with an implicit override. If the resulting style on your page is the desired one, you should delete the overridden property (it will also reduce the file size).

# 100/100

# You provide a favicon

#2379

Well done, you defined a favicon on this page.

Favicon is a small image providing an icon to a website. It's located in the root of your server and the browser will always request it. It is better not to respond with a 404 HTTP code (not found).

Moreover, this file will be asked on every requested web page, so make it cachable: the client will request it only once. See

#### 100/100

# ✓ You do not gzip/deflate PNG images

#2398

Compression has a cost, on the server and the client browser. So you should enable it only if it is effective.

# PNG and compression

PNG images does not support gzip compression. It is useless to perform compression operations on server side, and decompression on client side for this kind of files.

This page does not compress images. This is a best practice.



# Did you know?

#2457

# 1 This page contains 11 links

Two kind of links exist:

- Internal links that refer to pages with the same domain name;
- External links that point to other websites (must be relevant and point towards quality content).

If you reference many links, you can ask the SEO crawlers to consider only some of them, by adding the rel=nofollow attribute to the irrelevant ones (e.g., advertisements).

Here is the distribution of 11 links present in the page:

- 3 internal links (27,27%)
- 8 "follow" external links (72,73%)
- No "nofollow" external link (0,00%)

# Well done, these best practices are respected

# 100/100

# ▼ This page defines <h1> and <h2> tags

#78

We recommend putting page keywords in at least the h1 and h2 tags. Search engines use the h1, h2, and h3 tags for SEO purposes.

This page contains:

- 1 <h1> element(s)
- 1 <h2> element(s)

# 100/100

# ✓ You have defined a <meta> 'description'

#84

The page should define a unique description.

# Description in search engines

The description of the page may be directly displayed in search engine results pages (SERP):

Amazon.com: Online Shopping for Electronics, Apparel, Computers ...

https://www.amazon.com/ - Traduire cette page

Online retailer of books, movies, music and games along with electronics, toys, apparel, sports, tools, groceries and general home and garden items. Region  ${f 1}\dots$ 

It allows you to control at best the entry preview in search engines, and to improve the click rate to your page. Learn more.

# How to define a page's description?

Use <meta name="description" content="page description"> and place it in the <head> tag.

This page defines one <meta> description:

Xarta - David Bayliss's cool self-hosted domain name. Links to work on GitHub, other websites, and, pending, videos



# This page uses only standard image formats

#81

The images that use a non-standard format may not be indexed by search engines.

Only these image formats are considered standard on the web: jpeg, jpg, png, gif, svg, ico, webp. You should consider an alternative to any other format.

Moreover, remember to treat the text around your images: some search engines analyze approximately the 10 words preceding and following the image in order to add a context to the image.

# 100/100

# ▼ This page specifies a <title> tag

#83

The page should define a unique title (using a <title> tag).

#### Use of titles by search engines

Once properly configured, the page title can be displayed in the search engine results page:

Amazon.com: Online Shopping for Electronics, Apparel, Computers ...

https://www.**amazon.com**/ ▼ Traduire cette page
Online retailer of books, movies, music and games along with electronics, toys, apparel, sports, tools, groceries and general home and garden items. Region  ${\bf 1}$ 

Using a suitable title is a major criterion for SEO. It allows you to control at best what is displayed in search results pages, and determine the keywords you want your site pops out.

#### How to define the title of a web page?

The title of the page is specified into the <title> tag, which must be placed into the <head> tag, at the beginning of the code.

This page defines a title the title tag.

Here is the page's title:

xarta

# 100/100

# You do not use query strings in the URL

#90

This URL does not contain any parameter.

A URL should be as readable as possible. For instance, <a href="http://example.com?userld=332&group=MyGroup">http://example.com?userld=332&group=MyGroup</a> is less readable than http://example.com/mygroup/me .

# 100/100

# ✓ Your <title> tag is an appropriate length

#2444

This page defines one title which contains less than 75 characters (see more information):

xarta

The longer your title is, the more your chances are to see the search engines to truncate it or even to select another one from your page content.



# ▼ Your <meta> 'description' has a good length

#2432

This page defines one <meta> description which contains between 20 and 300 characters:

Xarta - David Bayliss's cool self-hosted domain name. Links to work on GitHub, other websites, and, pending, videos

This description is used in search engine results pages (SERP), such as Google results, to indicate a short informative text about the page:

Amazon.com: Online Shopping for Electronics, Apparel, Computers ...

https://www.amazon.com/ ▼ Traduire cette page

Online retailer of books, movies, music and games along with electronics, toys, apparel, sports, tools groceries and general home and garden items. Region 1

Search engines do not necessarily use the description you provide to them. If the description seems irrelevant, they'll use other criteria to describe your page. However, you will not be penalized by Google if your text is too long or too short. See

# 100/100

#### You do not duplicate too much your links

#2456

A same link duplicated several times within a same page may be a bad signal for search engines.

#### Natural links

Search engines check the quality and the relevance of your page. By repeating the same link over and over, you might be penalized.

# 100/100

# None of your titles are empty

#2467

<h1> , <h2> and <h3> tags should contain keywords related to the content.

The titles included on this page provide a content.

# 100/100

# v robots.txt file is well defined

#2503

Indicate to web crawlers which URLs should be explored on your website.

#### The robots.txt file

Place your robots.txt file in the root of the website. It will be interpreted by the robots in charge of your SEO. It delivers instructions to specify the pages to explore by robots, like Google bot.

Note that these directives are indicative only. A lambda robot will not be blocked by the restrictions specified by the file.

We detected a robots.txt file on this website.



# The words are well separated in this URL

#2345

You should prefer the use of dashes in the URL.

#### **URL and SEO**

The words in the URL are among the many factors impacting the SEO: if you search the words web performance in a search engine, one criterion used by Google will be to check if the URL contains the words performance and web.

But for Google, underscores are not word separators: If your URL contains web\_performance it will not help to highlight the page on the query web performance ( web\_performance is regarded as a unique word).

This behavior is however not common to all search engines. For example, Bing does not differentiate dashes and underscores.

Please note that Google does not penalize you for using an underscore in the URL.

#### How to resolve the issue?

We recommend using - instead of \_ on your new web pages.

This is more complex fix the issue on existing web pages, such as the one analyzed here, because you can't just rename the URL (you will lose all your SEO efforts). It is then necessary to set up a permanent redirect (HTTP code 301) retaining the old URL, that forwards to the new URL. Beware: many redirects on your site can also affect the visibility of your web pages. So do not use redirects if the number of relevant pages remains limited.

This URL respects the tip.

# 100/100

#### Your site uses Open Graph properties

#2399

You can help social networks understand information related to the page by using Open Graph properties.

#### The Open Graph properties explained

Several properties allow social networks to learn more about the page's content. We recommend using at least the required properties:

•	<meta content="The title" property="og:title"/>	Example
•	<meta content="The type" property="og:type"/>	Example
•	<meta content="http://url.com/" property="og:url"/>	Example
•	<meta content="http://image.jpg" property="og:image"/>	Example

This information is used to improve links between your page and various social networks, including Facebook. Read more about Open Graph here.

This page provides information to social networks.



# **Your Optimization Priorities**

# 90/100

# Disable the auto detection of resource type

#2485

Protect yourself from malicious exploitation via MIME sniffing.

#### MIME-Type sniffing explained

Internet Explorer and Chrome browsers have a feature called "MIME-Type sniffing" that automatically detects a web resource's type. This means, for example, that a resource identified as an image can be read as a script if its content is a

This property allows a malicious person to send a file to your website to inject malicious code. We advise you to disable the MIME-Type sniffing to limit such activity.

#### How to prevent MIME-Type sniffing

Configure a "X-Content-Type-Options" HTTP header. Add the "X-Content-Type-Options" HTTP header in the responses of each resource, associated to the "nosniff" value. It allows you to guard against such misinterpretations of your resources.

On this page, you should configure the following resources, that risk being misinterpreted:

It appears these files are hosted by a third-party, so they may not be within your control. However, you should consider any alternative to these resources to improve your page performance.

- https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css
- maxcdn.bootstrapcdn.com/font-awesome/4.7.0/fonts[...]me-webfont.woff2?v=4.7.0
- https://res.cloudinary.com/xarta/image/upload/v1497350431/xarta/8pUBeuw.jpg
- res.cloudinary.com/xarta/image/upload/v149735043[...]/xarta/spiral-galaxy.jpg
- res.cloudinary.com/xarta/image/upload/v149700706[...]a/2014-me-at-work256.jpg
- https://res.cloudinary.com/xarta/image/upload/v1497350431/xarta/KmwuMPu.jpg
- https://res.cloudinary.com/xarta/image/upload/v1497350431/xarta/fB9BaJJ.jpg
- https://res.cloudinary.com/xarta/image/upload/v1496407928/xarta/Eu0MQrk.png
- https://res.cloudinary.com/xarta/image/upload/v1496588500/xarta/moon.png

# Did you know?

#2463

# i Your server should be able to communicate with HTTP while it uses a HTTPS connection

Take precautionary measures against attacks like "man in the middle" by making sure to only communicate in HTTPS with the server.

# The HTTP Strict Transport Security (HSTS) Header

When you communicate with a server through a secured connection, every sent request towards this server should use the HTTPS protocol. The HTTP HSTS header allows to indicate to the browser that all the requests sent to the domain concerned must be done via HTTPS. If the URL is presented under "http://...", the web browser is automatically going to replace it by "https://...".

However, we advise you to not set this header unless your entire website serves its resources in HTTPS.

No HSTS header has been detected on this page.



#### #2474

#### 1 9 resources on this page are for public use

By default, the browser accepts to perform AJAX requests, or to retrieve web fonts, only on the same domain name of the page. So a font provided by toto.com can only be used by the pages of toto.com. This prevents misuse of your resources by

Some resources are public, and explicitly want to be available to everyone (eg Google Fonts). In this case, the HTTP header Access-Control-Allow-Origin can be used with the value "\*". You should, however, use this property if your resource has aimed to be used by the greatest number. Otherwise, we recommend that you keep the default, or set a specific domain name in the "Access-Control-Allow-Origin" HTTP header.

You should be aware of the following resources, that use a Access-Control-Allow-Origin: \* HTTP header. Make sure they are actually intended to be used by pages from all domain names:

It appears these files are hosted by a third-party, so they may not be within your control. However, you should consider any alternative to these resources to improve your page performance.

- https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css
- maxcdn.bootstrapcdn.com/font-awesome/4.7.0/fonts[...]me-webfont.woff2?v=4.7.0
- https://res.cloudinary.com/xarta/image/upload/v1497350431/xarta/8pUBeuw.jpg
- res.cloudinary.com/xarta/image/upload/v149735043[...]/xarta/spiral-galaxy.jpg
- res.cloudinary.com/xarta/image/upload/v149700706[...]a/2014-me-at-work256.jpg
- https://res.cloudinary.com/xarta/image/upload/v1497350431/xarta/KmwuMPu.jpg
- https://res.cloudinary.com/xarta/image/upload/v1497350431/xarta/fB9BaJJ.jpg
- https://res.cloudinary.com/xarta/image/upload/v1496407928/xarta/Eu0MQrk.png
- https://res.cloudinary.com/xarta/image/upload/v1496588500/xarta/moon.png

#### #2526

# Do "target=\_blank" links introduce a security leak on this page?

Using the target blank attribute is rarely recommended. Nevertheless, if you need to use this attribute, note that a security leak could cause harm to your visitors, particularly if your site is open to visitor contributions.

It allows the targeted page to manipulate the window.opener.location property, and thus to perform a redirect within the parent tab. When the user gets back to the parent tab, he can be facing a malicious website (phishing, etc).

We recommend you to add the rel=noreferrer attribute when using a target = blank to an external website. This will block access to "window.opener"

If your website allows users to publish contributive content (eg comments, customer reviews, etc.), be sure to automate the addition of this protection. Otherwise, a user could easily exploit this breach.

The following links may be exposed to this vulnerability:

<a href="#" target="\_blank"> <i class="fa fa-question-circle"></i> <span>About</span> </a>

# #2552

# Specify which SSL certificates can be accepted by the browser

Provide the list of valid public cryptographic keys to the web browser in order to avoid some Man-In-The-Middle attacks.

# **Public Key Pinning**

The HTTP Public Key Pinning header allows to define all the public cryptographic keys supported to access your domain via SSL:

Public-Key-Pins: pin-sha256="base64=="; max-age=expireTime [; includeSubdomains][; report-uri="reportURI"]



The browser stores this list. During the second visit, it will compare this list with the key provided by the certificate authority (CA). If they don't match, the browser will consider the certificate as corrupted and will block the access to the page.

Please, be careful while configuring this header: if the provided certificates are not up to date, your website will be unreachable (see a detailed feedback). Do specify several certificates, including (at least) one fallback certificate. It will allow to deal with the expiration or invalidation cases of a certificate.

Note: We advise you to deploy it progressively by using the report-only mode (Public-Key-Pins-Report-Only header) and limited cache validation times.

Read more about HPKP.

This page doesn't send a Public Key Pinning header.



#2553

# 1 Do all third parties resources deliver the right content?

This page loads data from third parties, you should ensure their integrity.

# SubResource Integrity (SRI)

Use SRI to ensure that a third party resource has not been tampered. Add the integrity attribute to <script> and tags loading this kind of resource. Example:

<script src="https://exemple.com/exemple-framework.js"</pre> integrity="sha384-oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQIGYl1kPzQho1wx4JwY8wC" crossorigin="anonymous"> </script>

The integrity attribute value is equal to the base64-encoded hash (SHA) of the resource. The browser compares this hash with the downloaded content in order to determine if the resource matches the expected content.

You can create the SHA thanks to several tool. In command line, you can use openssl. You can also test some online tools, as srihash.org or report-uri.io. Learn more about SubResource Integrity.

# Well done, these best practices are respected

# 100/100

# **▼** You defined a Content Security Policy

#2433

It is critical to restrict the origin of the contents of your webpage to protect your website from cross-site scripting attacks (XSS).

#### XSS attacks explained

An XSS attack aims to inject content into a page.

You can protect your pages against these attacks by implementing a content security policy that tells the web browser which servers are allowed to deliver resources on each page. If the browser makes a request to an unauthorized server, it must inform the user.

# How can I prevent an XSS attack?

Set up a "Content-Security-Policy" (CSP) HTTP header. To specify a security policy on the source of your resources, configure your server so the response of the first resource contains the "Content-Security-Policy" HTTP header.

Here's an example:

Content-Security-Policy: script-src 'self' https://apis.google.com

In this case, the page loads correctly provided that all the scripts come from the current host or https://apis.google.com.

Read more about the CSP HTTP header. You can also look at the CSP directives.

Please, be careful, if the header is misconfigured, some of your content, scripts, or styles may be blocked. That could cause unwanted side effects. Moreover, the restrictions apply to all pages of the website. We recommend you test the different pages of your website before deploying this header in your production environment.

This page defines a CSP, which allows it to control the origin of its resources.



# ▼ This page is not exposed to "clickjacking" type attacks

#2480

Keep malicious people from integrating your pages into their websites.

#### Clickjacking explained

This kind of attack happens when your page gets integrated with a malicious website via <frame> or <iframe> tags. By doing this, attackers can persuade users that they are on your own page when they are not. The unsuspecting user may enter personal information that is visible on and thus vulnerable to the malicious website.

To avoid this, always indicate which domains have permission to integrate your pages.

#### How to prevent clickjacking?

Configure a "X-Frame-Options" HTTP header. Configure your server so the main resource response includes the "X-Frame-Options" HTTP header.

Three values may be defined:

- DENY to prevent any frame or iframe from integrating the page;
- SAMEORIGIN to authorize only frames from the same domain name;
- ALLOW-FROM uri to indicate the domains allowed to integrate a page into frame (however is not compatible with some browsers)

The "X-Frame-Options" HTTP header is configured on this page; you are not exposed to clickjacking.

# 100/100

# You block access to the entire page when an XSS attack is suspected

#2482

Make sure that the user's browser does all it can to prevent an XSS-type attack.

#### XSS attacks

An XSS-type attack (XSS stands for Cross-Site Scripting) aims at injecting content into the page.

Recent browsers have an integrated protection against XSS attacks. However, this protection can be disabled. To prevent any harm to the user, we recommend that you force the activation of the XSS Protection, and should an XSS attack be detected, block access to any of the page content.

#### Solution: configure an "X-XSS-Protection" HTTP header

Add the "X-XSS-Protection" HTTP header with "1; mode=block" as value (1 to indicate the activation, and mode=block to indicate that the entire page must be blocked if a problem occurs).

The XSS protection is enabled on this page.

# 100/100

# The secure version is used systematically

#2509

Redirect users using the HTTP version of the page to the HTTPs version.

#### HTTPs redirect

This page uses HTTPS, however your visitors can still browse the HTTP version. You should perform a redirect to secure the browsing of your users accessing the page using the HTTP protocol.

Your users are automatically redirected to the secure version of the page.

# 100/100

# This page is not exposed to the Mixed Content issue

#2397

This page uses the HTTPS protocol, , and all resources are using HTTPS protocol. So there is no Mixed Content vulnerability.

For further information on Mixed Content, you can read our article on this subject.



# Well done, these best practices are respected

# 100/100

# ▼ This page uses Universal Analytics

#2348

Since April 2014, Google provides the new Google Analytics version, called Universal Analytics. This tool allows a better understanding of how visitors interact with your online content.

This page uses the Universal Analytics properties.

Note that it is perfectly safe to include both ga.js/dc.js (old version) and analytics.js (new version) snippets on the same page.

