Page loading speed is a key indicator of website performance. The faster the page loads, the more likely users are to reach your products or services rather than abandon it. Today we'll talk about Lazy Load and slow image uploads on your site.

What Is Lazy Load

When you website is ready to see the light of day, it’s time to pay attention to the page loading speed. If it takes too long, you can be certain that no one will waste time waiting, especially on mobile devices. This issue is vital if your site has many images that add weight to a page.

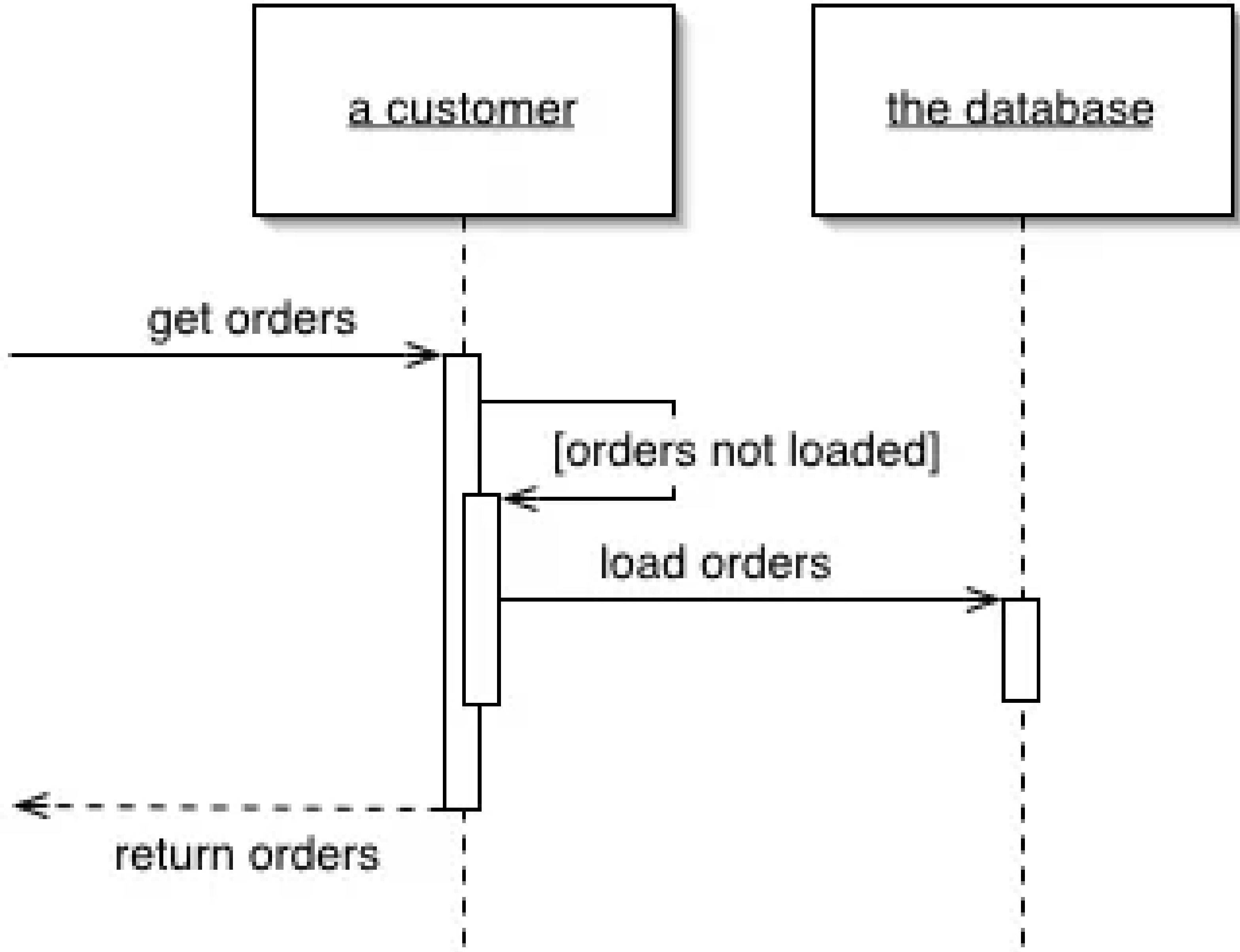
What you need here is the jQuery and Zepto plugin Lazy Load, which slows the downloading of images on your site, opening them only when the user scrolls down the page.

- *You see this effect constantly when you use social networks. Just imagine how long it would take to download your social network page, with 8000 posts on the wall, without Lazy Load.*

Lazy Load has already become an integral part of Web 2.0 sites, not only because it alleviates page weight, but also because it looks cool.

A Couple of Technical Terms

The object does not contain data, but knows where to get it. Have a look at the scheme:



Instead of useless downloading, a marker is put in place indicating that data have not been loaded and should be if needed. As you know, if you're lazy, if you didn't bother doing something that ended up not needing done then you come out on top.

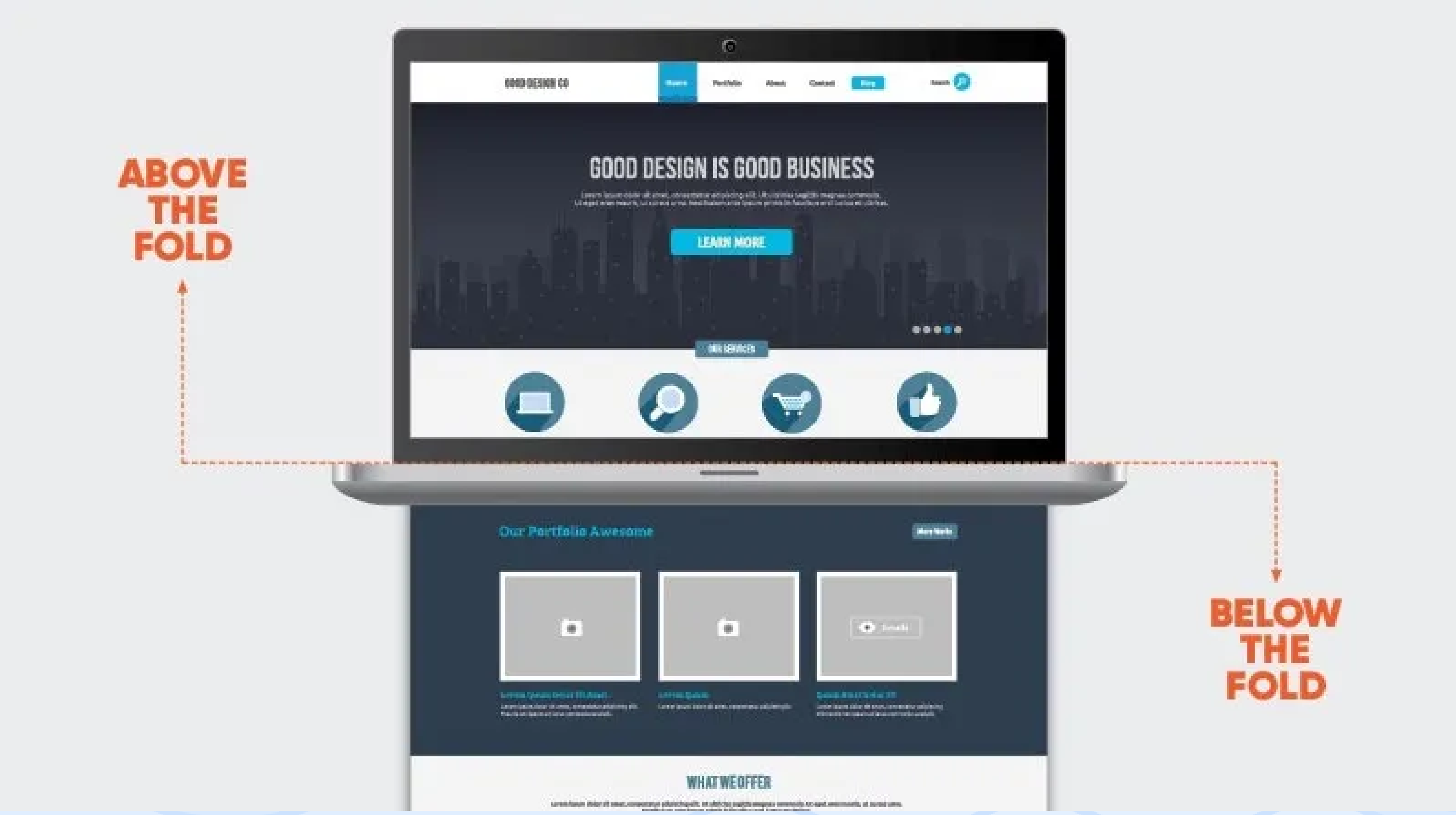
There are four main options for Lazy Load:

1. **Lazy Initialization** uses a special index (usually null) to mark the field as unloaded. Each time the field is accessed, the index value is checked and, if the field value is marked as unloaded, it is

then loaded.

2. **Virtual Proxy** is an object with the same interface as the real object. When you first access an object method, the virtual proxy loads the real object and redirects the execution.
3. **Value Holder** is an object with the getValue method. The client calls the getValue method to get the real object. GetValue calls the load.
4. **Ghost** is an object with no data. At the first call to the method, the ghost loads all the data at once.

<H2> The LazyLoad Feature in Chrome Will Save Mobile Traffic



To make users’ lives easier, the Chrome browser on Android will enable a Blink LazyLoad feature. As explained before, this function stops a web page from downloading until the user personally requests the rest of the page.

Some questions remain about how this will affect users who want to read web pages offline, but for everyone else this can be the next important step in saving mobile traffic.

There are two parts to the system:

- Blink LazyFrames;
- Blink LazyImages.

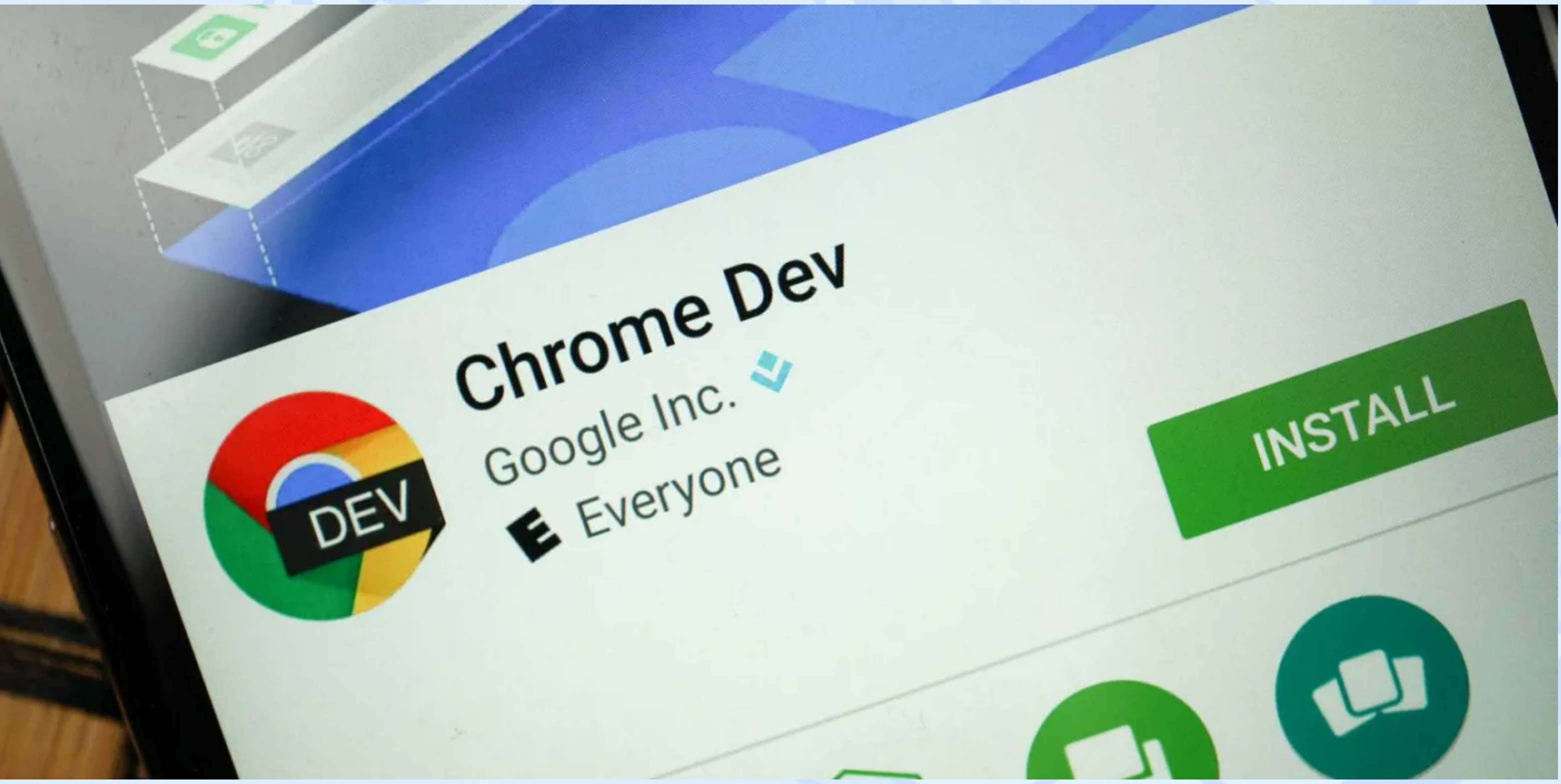
Loading frames is delayed if the frame is a third-party iframe larger than 4 × 4. The system does not delay the loading of third-party frames because they share the Javascript context with the implementation page.

The function first records how much mobile data is deferred to each frame. At this point, the system will write 50 KB per frame and will re-add 50 KB if the frame is loaded later. Delayed images will be calculated by analysing the total size of the resource from the Content-Range response header.

As for the web page design, there's no need to worry - a rectangular filler of an appropriate size will be inserted wherever an image load is postponed. Both the foreground and background images will be deferred until they are called by the user - that is, when the screen is scrolled down.

- *LazyLoad will be implemented primarily on Android. If this implementation is successful, Google will consider other platforms that support the Chrome browser.*

At the moment the plan is to implement LazyLoad as a default function. Users who don't wish to use it will have to switch it off. You will have an opportunity to disable LazyLoad on each separate web page.



Feature-testing will begin in the Chrome Dev app on Google Play if everything goes as planned.

So, with Chrome this problem will be solved - but what about other browsers? You should take care of the download speed of your site. Here are some useful resources and tools:

- Lazyload jQuery plugin. The plugin page says it isn't usable and doesn't work with modern browsers, and that the author has other things to do. But don't despair - find a fresh version here.

Its work is extremely simple and is described on the site.

- YUI 3: ImageLoader. Using YUI Image Loader is a bit more complicated than using Lazyload. Images have to be collected into groups and groups to hang triggers. The group image will not be loaded until the trigger is activated. You can find plenty of written information about it, but not much sense. But there are useful utilities. We recommend digging.
- MooTools LazyLoad. Everything is clear from the title. It's very easy to use.
- lazierLoad — Image Lazy Loader for Prototype. This one is for prototypes.

If the above-mentioned utilities look like gibberish to you, you always can ask our experts - they'll have no trouble explaining how to make your site better and life easier.