

# Tech Support & Bug Report Guidelines

Version 1.3, 2018-06-15

Thomas Pronk, [pronkthomas@gmail.com](mailto:pronkthomas@gmail.com)

## Introduction

This manual contains technical support and bug report guidelines for studies that are conducted online. The remainder of the manual consists of two sections:

- **If a Participant Reports a Problem.** This section covers the kind of bug reporting we can hope to expect from participants. It also has a sub-section on basic tech support.
- **If You Discover a Problem.** This section offers more advanced bug reporting that makes use of browser development features. If you discover a problem and manage to reproduce it, but the developer does not, then you might want to deliver this information to help her out.

## If a Participant Reports a Problem

What to do when a participant reports that the software for administering your study malfunctioned? In many cases, a malfunctioning experiment may simply due to a faulty Internet connection or client computer; see “Quick Fixes” for solutions to common problems. In some cases, the malfunction may be due to a programming error, a so-called ‘bug’. Bugs in running experiments are more challenging to find and solve for two reasons:

1. They occur in unknown circumstances, making them more difficult to pinpoint
2. We may initially need to rely on bug reports provided by non-expert participants.

Good reporting can be very valuable in helping to fix such post-deployment bugs, but what makes a good bug report? This document aims to provide some guidelines; when a bug is suspected, concrete and detailed information is valuable to your technician for finding out what is wrong.

### Be concrete

A useful bug report describes the actions and experiences of the participant in very concrete terms. For instance “I clicked the ‘Submit’ button with the mouse” is better than “I submitted the form”. “I saw a red text that said ‘Page error #219’” is better than “the page reported an error”.

### Provide a lot of details

Some specifics that can be most valuable:

- At what time did it happen? How long was the participant taking part before the bug occurred?
- What software did the participant use? Operating system & Browser, what version numbers?
- Was there anything unusual before it went wrong?
- What exactly did the participant do just before it went wrong?
- What did the participant see just before and after when it went wrong?
- Any e-mails, screenshots, etc. received from the software

## Quick Fixes

In most cases, a malfunctioning experiment may be solved by applying a simple workaround:

1. Try again sometime later
2. Try a different web-browser (if using Chrome, try Firefox, and vice versa)
3. Try a different computer

## If You Discover a Problem

This section is about producing the types of logs your technician would inspect to debug the website. These logs are quite detailed, so they can get big quite quickly. Hence, this section is split in sub-sections on what to do before and after triggering the problem. It's important to take these steps just before and after the problem, so that the logs don't contain unnecessary information.

NB - All the examples are based on the browser Chrome. If you can only reproduce the problem in a different browser, then please get in touch with your technician for additional instructions.

### Before Triggering the Problem

We will obtain the logs via the developer tools, which can be opened by pressing Ctrl + Shift + I or F12. It can also be opened by navigating to the menu option shown in Figure 1. Note that the developer tools need to be open *before* you trigger the problem in order to work. In the developer tools, you'll see a list of tabs at the top, named "Elements Console Source etc.", which each opens a differently looking screen. We're mainly interested in the Console and Network screens.

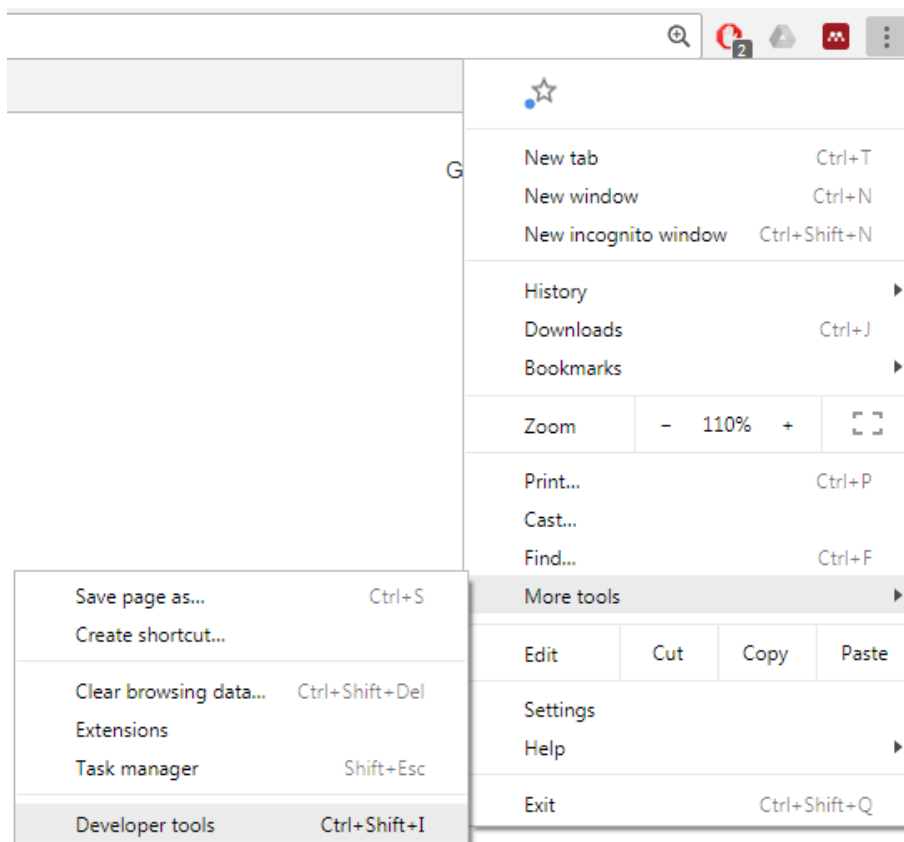


Figure 1. Opening developer tools via the menu

## Preparing the Console Screen

The Console screen (Figure 2) reports Info, Warnings, and Errors, about the web-page that is being displayed. Info is displayed in black, and consists of debug messages generated by the applications running on the web-page. Warnings and Errors are displayed in red, and indicate possible problems detected by the web-browser. *In the Console screen, be sure that **Preserve log** is enabled. To accomplish this, open the Console settings by clicking the blue cog, and ensure that there is a tick mark at the option “Preserve log”. Next, clear the log (by clicking the “no parking” button) just before triggering the problem.*

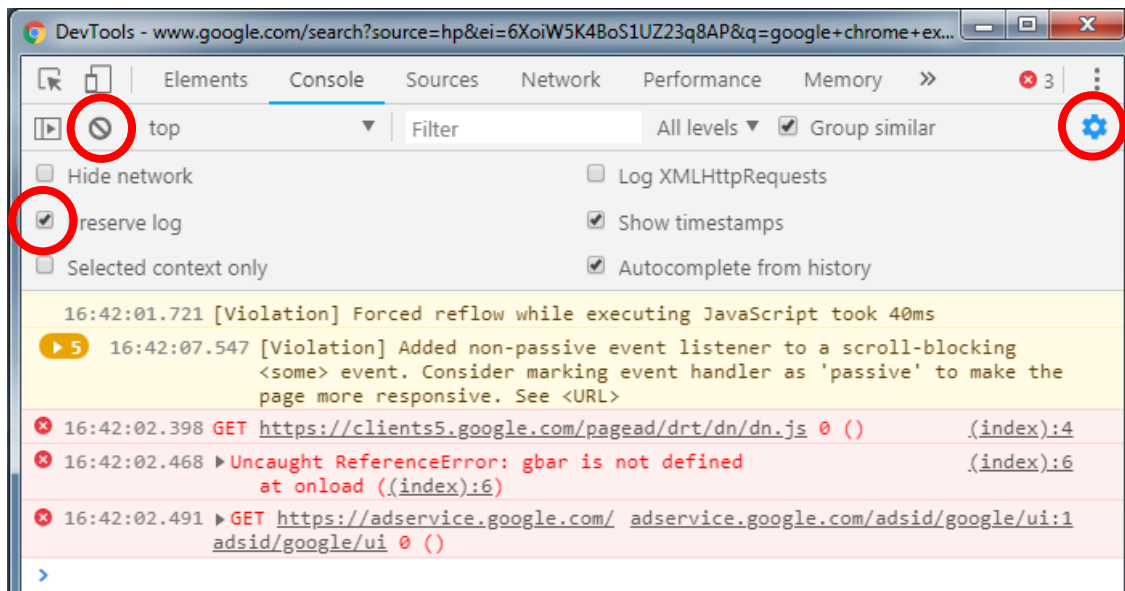


Figure 2. An example console screen. The “console settings”, “clear logs”, and “preserve logs” buttons, are marked with red circles.

## Preparing the Network Screen

The Network screen (Figure 3) reports the data that was sent back-and-forth between your computer and internet. This data may be text or pictures, or packages of abstract data, which is generally in the form of JSON. Similar to the Console, info is displayed in black and errors/warnings in red. *In the Console screen, be sure that **Preserve log** is enabled (which then is shown with a tick mark), and that you’ve cleared the log (by clicking the “no parking” button) just before triggering the problem.*

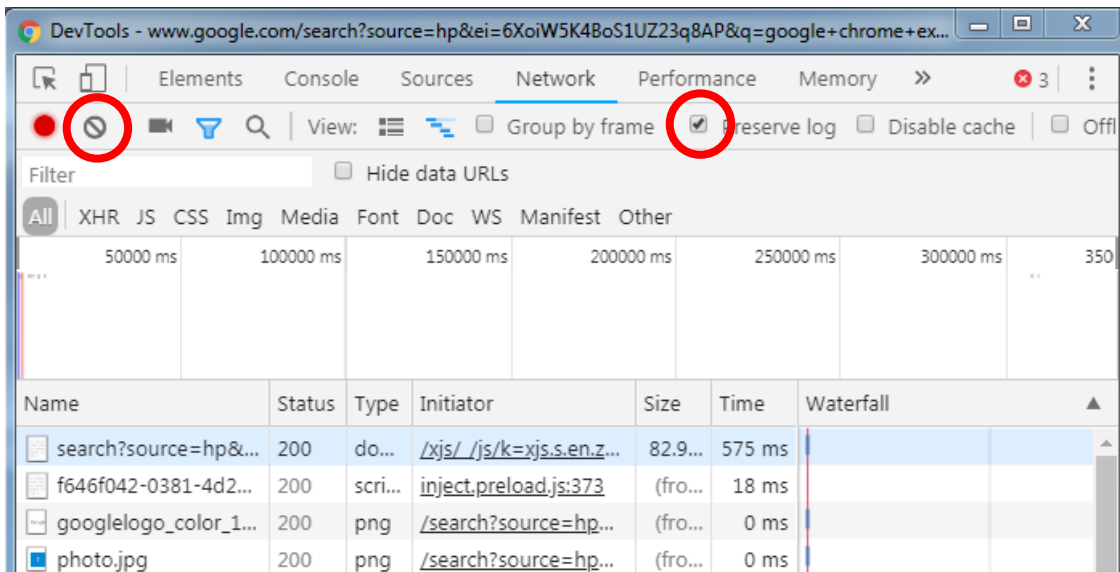


Figure 3. An example network Screen. The “clear logs” and “preserve logs” buttons are marked with red circles.

### After Triggering the Problem

After triggering the problem, directly go to the Developer tools in order to export the logs of the Console and Network:

- **Console.** In the console screen, right-click anywhere and pick **Save as** from the context-menu that appears
- **Network.** In the network screen, right-click on a log entry and pick **Copy -> Copy all as HAR** from the context-menu that appears (Figure 4). Next, open a new text file via a plain text editor such as Notepad (Kladblok), but don't use Word please. Paste (Ctrl + V) the contents and save the file.

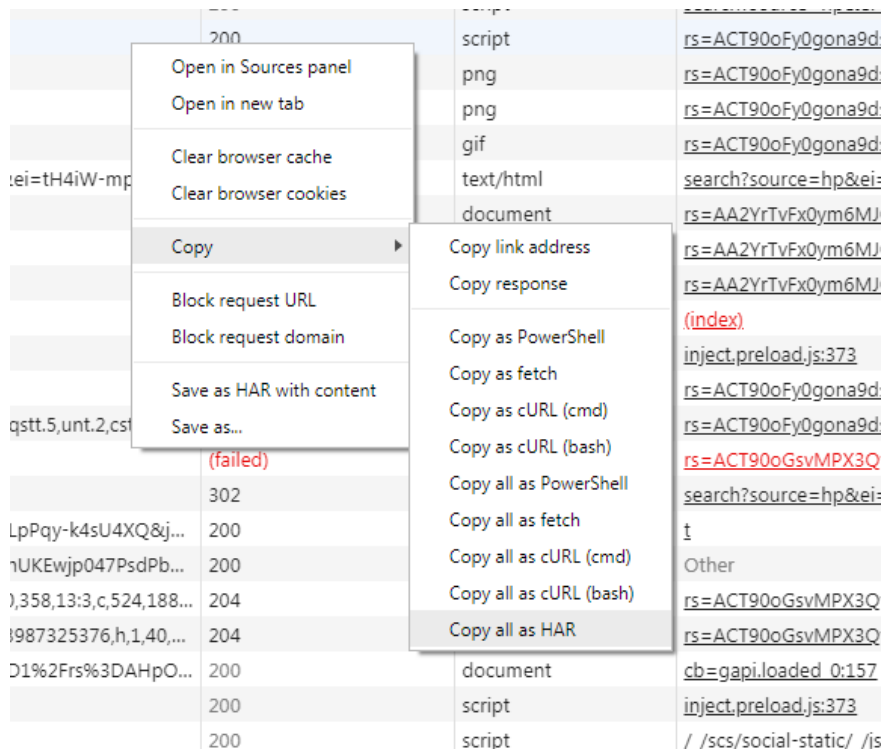


Figure 4. Saving the network log

## Version History

- **v1.0 & v1.1.** Initial version and an update before I tracked version history 😊
- **v1.2.** Added “If You Discover a Problem”, which covers Console and Network logs. Rewrote introduction.
- **v1.3.** Fixed language errors and added explanation on Console settings