

# **IPTV SD/HD SetTopBox**

Remote debug

version 1.0

## Contents

Overview.....	3
Environment preparation.....	4
Debugging process.....	5

## **Overview**

This document describes a deployment of the debug environment and the method of remote debug of IPTV-devices. The document assumes some knowledge of system administration and JavaScript.

## Environment preparation

In general the remote debugging can be represented by these parts:

- debug server
- set top box
- desktop browser

The first point from this list requires some preparation described below.

Debug server can be any standalone or virtual computer with Linux or Windows operating system available in the local network. The software required:

1. Node.js (<http://nodejs.org/>)
2. WEb INspector Remote (<https://npmjs.org/package/weinre>)

Node.js can be installed via the provided standard installation package or with the help of OS package manager (apt-get, yum, packman, emerge and so on). It's necessary to make sure that a firewall accept connection to the 8000 port.

When Node.js is ready WEb INspector Remote can be installed with its package manager NPM (this will download the web inspector and all its dependences):

```
npm install weinre
```

To start the debugger run this in the weinre directory:

```
node weinre
```

The output should be similar to this:

```
D:\tmp\node_modules\weinre>node weinre
2013-07-17T12:00:44.563Z weinre: pid:          30144
2013-07-17T12:00:44.567Z weinre: version:     2.0.0-pre-HH0SN197
2013-07-17T12:00:44.568Z weinre: node versions:
2013-07-17T12:00:44.572Z weinre:   http_parser: 1.0
2013-07-17T12:00:44.576Z weinre:   node      : 0.10.12
2013-07-17T12:00:44.581Z weinre:   v8       : 3.14.5.9
2013-07-17T12:00:44.586Z weinre:   ares     : 1.9.0-DEV
2013-07-17T12:00:44.590Z weinre:   uv      : 0.10.11
2013-07-17T12:00:44.597Z weinre:   zlib    : 1.2.3
2013-07-17T12:00:44.601Z weinre:   modules : 11
2013-07-17T12:00:44.606Z weinre:   openssl : 1.0.1e
2013-07-17T12:00:44.614Z weinre: options:
2013-07-17T12:00:44.618Z weinre:   httpPort: 8000
2013-07-17T12:00:44.622Z weinre:   boundHost: 192.168.1.71
2013-07-17T12:00:44.629Z weinre:   verbose: true
2013-07-17T12:00:44.634Z weinre:   debug: true
2013-07-17T12:00:44.638Z weinre:   readTimeout: 3
2013-07-17T12:00:44.645Z weinre:   deathTimeout: 5
2013-07-17T12:00:44.690Z weinre: starting server at http://192.168.1.71:8000
```

This means the web inspector is up and available by the given address.

## Debugging process

The detailed instructions how to setup connection from the set top box can be found on the web inspector start page <http://192.168.1.71:8000>. It's a simple web page that should look like this:

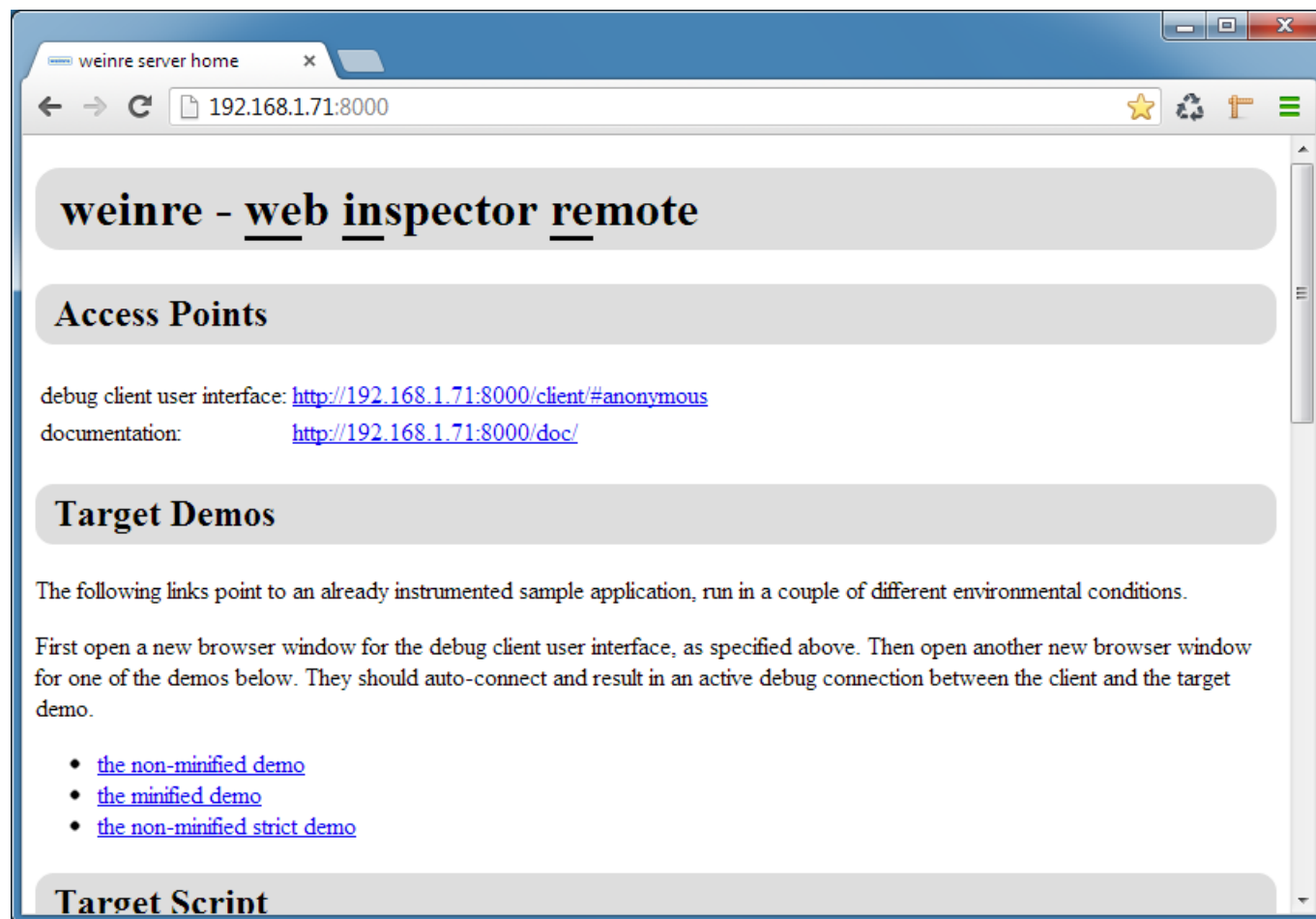


Image 1. Web inspector main page

In order to start debugging it's necessary to add this line to your test page you are starting on the set top box device:

```
<script src="http://192.168.1.71:8000/target/target-script-min.js#anonymous"></script>
```

For example your test file can be like this:

```
<!doctype html>
<html>
  <head>
    <title>test page</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <script type="text/javascript"
src="http://192.168.1.71:8000/target/target-script-
min.js#anonymous"></script>
  </head>
<body></body>
</html>
```

Start it with the command:

```
/usr/share/qt-4.6.0/stbapp -qws -display directfb /home/web/test.html
```

After the start it's possible to connect to it via the web inspector:

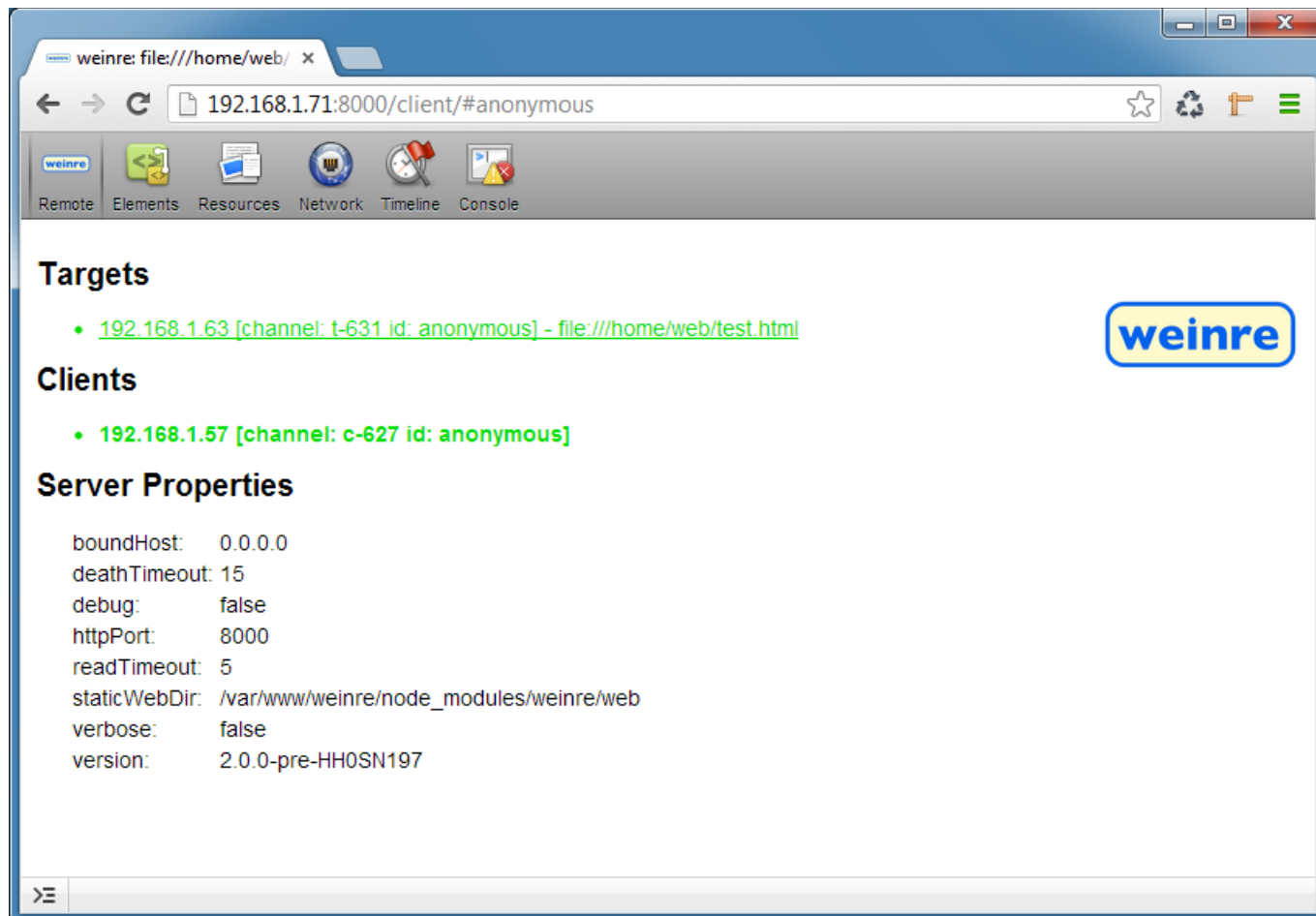


Image 2. Web inspector connection page

Here it's possible to see the HTML structure, available database, local and session resources, network requests with detailed information, timeline data and also direct exec JavaScript code.

At the same time there can be many connection from different std devices with different anchors.

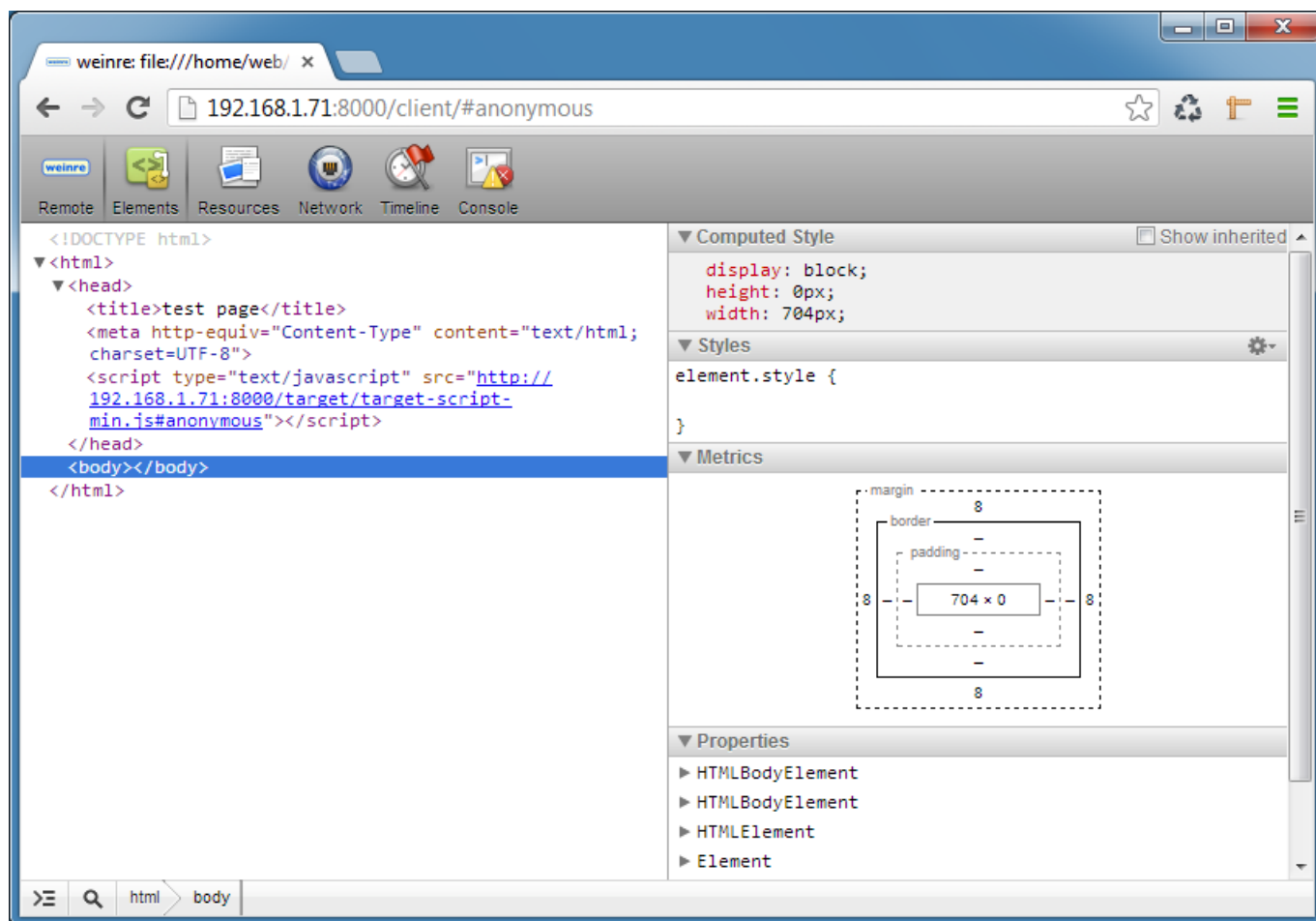


Image 3. Web inspector elements page

This page allows to see all the applied styles, metrics, element properties and listeners with the ability to modify these values.

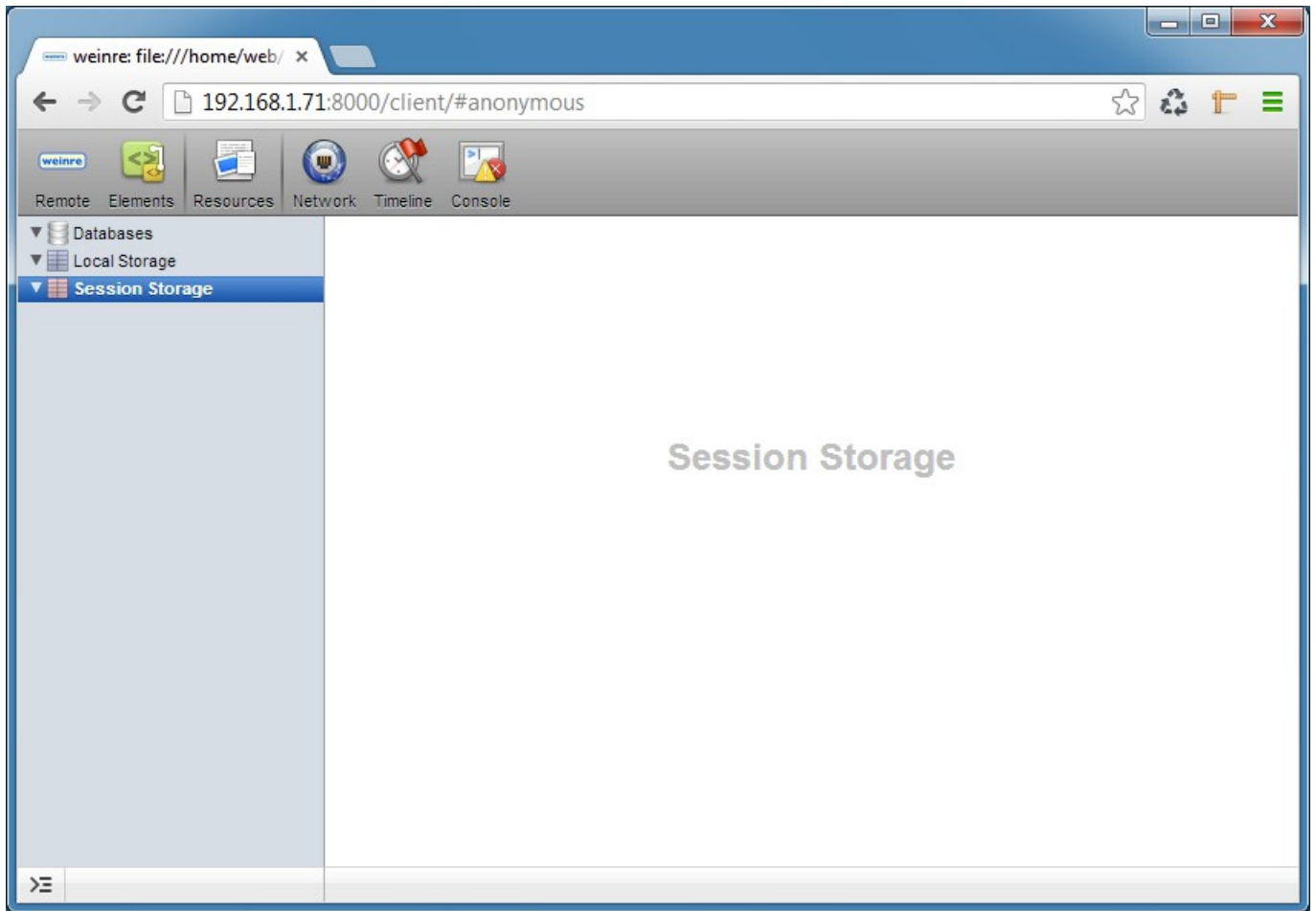


Image 4. Web inspector resources page



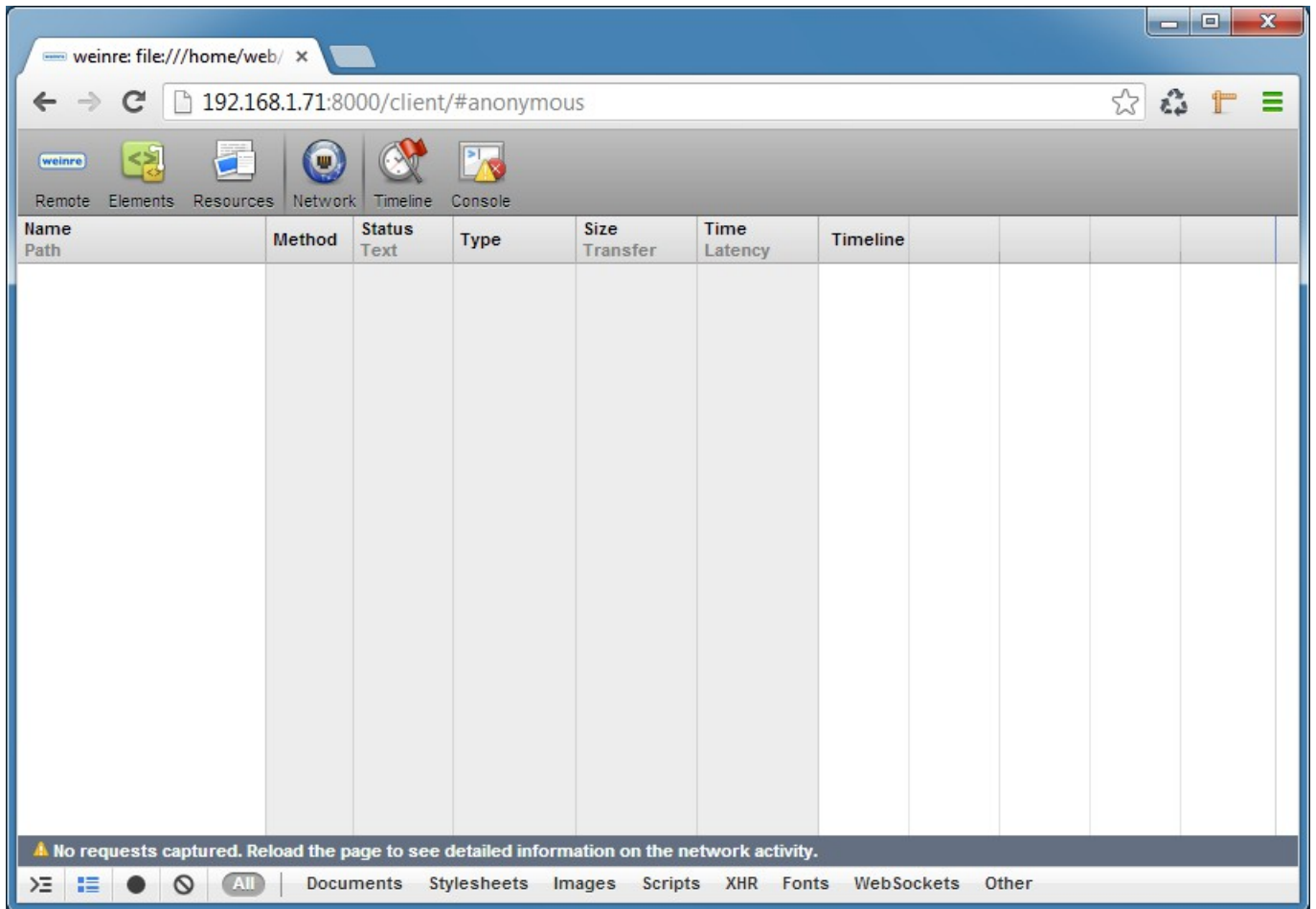


Image 5. Web inspector network page

All the network requests are visible on this page.

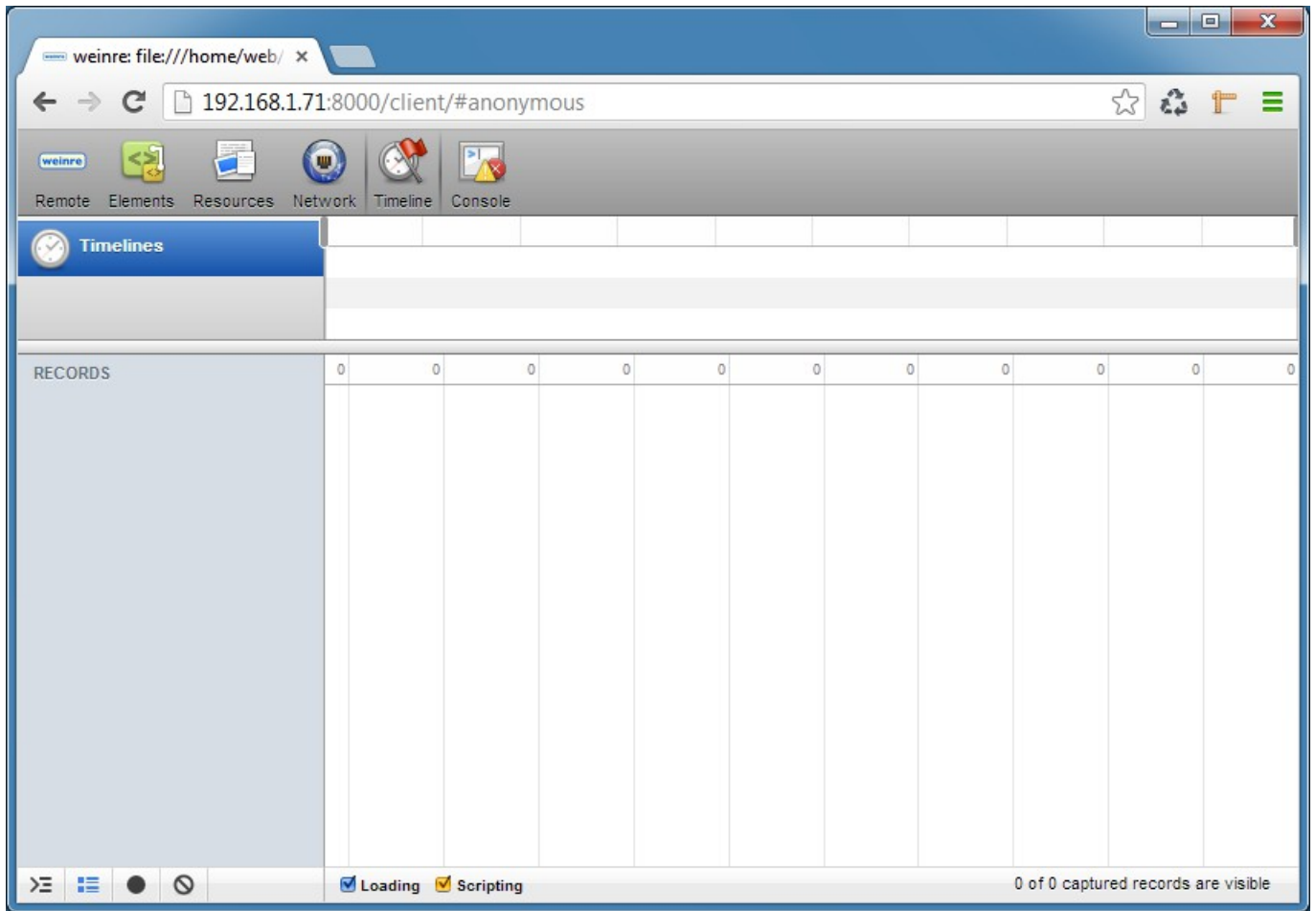


Image 6. Web inspector timeline page

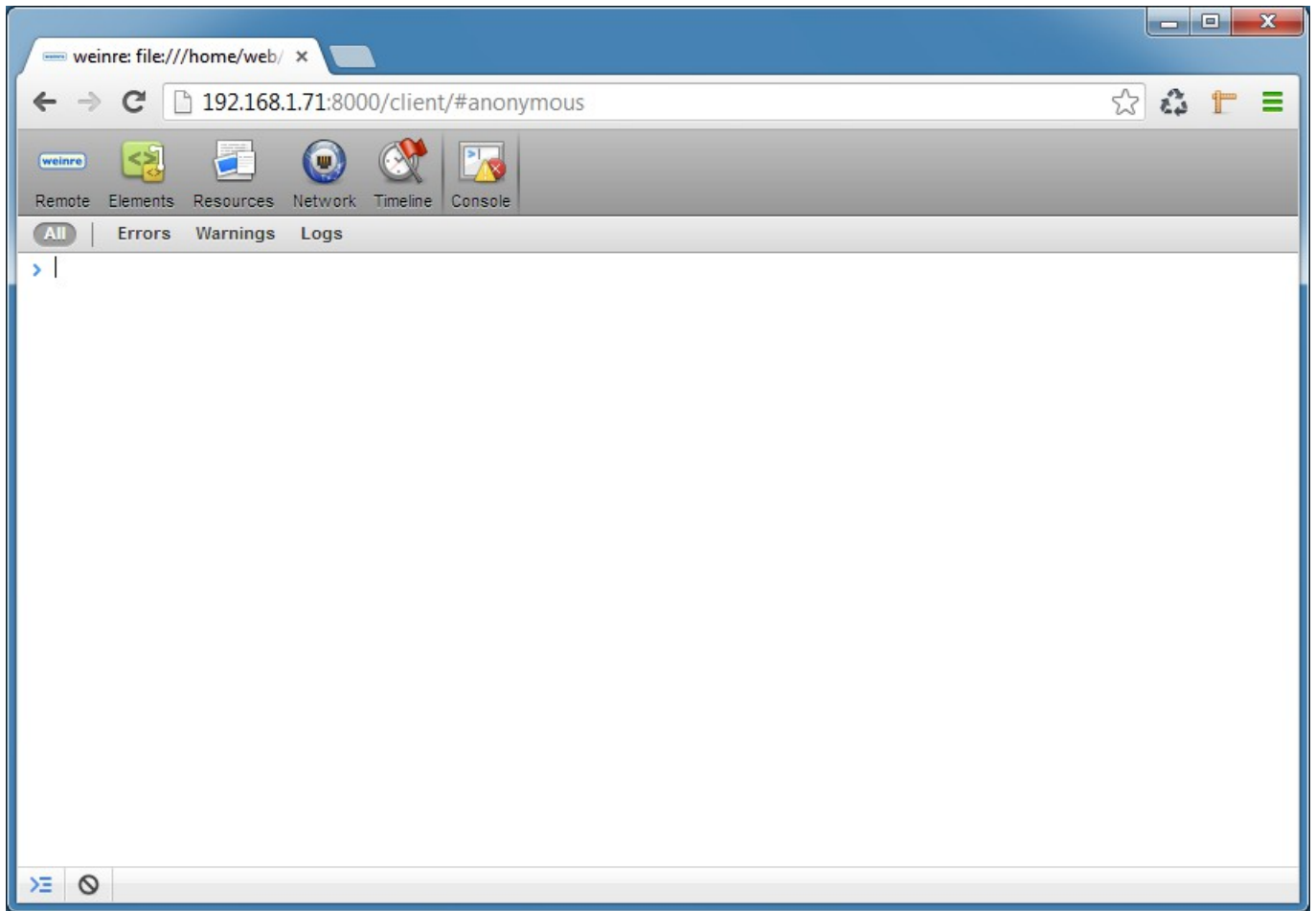


Image 7. Web inspector console page

This page allows to execute any JavaScript code and see the result in real time.