



General

Request

Browser Steps

Visual Filter Selector

Filters & Triggers

Notifications

Stats

Pro-tips:

- Use the preview page to see your filters and triggers highlighted.
- Some sites use JavaScript to create the content, for this you should [use the Chrome/WebDriver Fetcher](#)

CSS/JSONPath/JQ/XPath Filters

```
/html/body/div[2]/div[4]/div[4]/div[2]/div[5]/ul/li[*]/a
```

One rule per line, *any* rules that matches will be used.

- CSS - Limit text to this CSS rule, only text matching this CSS rule is included.
- JSON - Limit text to this JSON rule, using either [JSONPath](#) or [jq](#) (if installed).
 - JSONPath: Prefix with `json:`, use `json:$` to force re-formatting if required, [test your JSONPath here](#).
 - jq: Prefix with `jq:` and [test your jq here](#). Using `jq` allows for complex filtering and processing of JSON data with built-in functions, regex, filtering, and more. See examples and documentation [here](#).
- XPath - Limit text to this XPath rule, simply start with a forward-slash. To specify XPath to be used explicitly or the XPath rule starts with an XPath function: Prefix with `xpath:`
 - Example: `//*[contains(@class, 'sametext')]` or `xpath:count//*[contains(@class, 'sametext')]`, [test your XPath here](#)
 - Example: Get all titles from an RSS feed `//title/text()`
 - To use XPath1.0: Prefix with `xpath1:`

Please be sure that you thoroughly understand how to write CSS, JSONPath, XPath, or jq selector rules before filing an issue on GitHub! [here for more CSS selector help](#).

Remove elements

```
header  
footer  
nav  
.stockticker
```



Text filtering

Limit trigger/ignore/block/extract to;

- Added lines**
- Replaced/changed lines**
- Removed lines**

Note: Depending on the length and similarity of the text on each line, the algorithm may consider an **addition** instead of **replacement** for example. So it's always better to select **Added+Replaced** when you're interested in new content.

When content is merely moved in a list, it will also trigger an **addition**, consider enabling `Only trigger when unique lines appear`

Sort text alphabetically

Helps reduce changes detected caused by sites shuffling lines around, combine with `check unique lines` below.

Only trigger when unique lines appear

Good for websites that just move the content around, and you want to know when NEW content is added, compares new lines against all history for this watch.

Trigger/wait for text

```
Some text to wait for in a line
/some.regex\d{2}/ for case-INsensitive regex
```

- Text to wait for before triggering a change/notification, all text and regex are tested *case-insensitive*.
- Trigger text is processed from the result-text that comes out of any CSS/JSON Filters for this watch
- Each line is processed separately (think of each line as "OR")
- Note: Wrap in forward slash / to use regex example: `/foo\d/`

Ignore text



- Each line processed separately, any line matching will be ignored (removed before creating the checksum)
- Regular Expression support, wrap the entire line in forward slash `/regex/`
- Changing this will affect the comparison checksum which may trigger an alert
- Use the preview/show current tab to see ignores

Block change-detection while text matches

```
For example: Out of stock  
Sold out  
Not in stock  
Unavailable
```

- Block change-detection while this text is on the page, all text and regex are tested *case-insensitive*, good for waiting for when a product is available again
- Block text is processed from the result-text that comes out of any CSS/JSON Filters for this watch
- All lines here must not exist (think of each line as "OR")
- Note: Wrap in forward slash `/` to use regex example: `/foo\d/`

Extract text

```
/. *gt.* |.*€.*/  
/. *gt.* |.*€.*/
```

- Extracts text in the final output (line by line) after other filters using regular expressions or string match;
 - Regular expression - example `/reports.+?2022/i`
 - Don't forget to consider the white-space at the start of a line `/.+?reports.+?2022/i`
 - Use `/(?aiLmsux)` type flags (more [information here](#))
 - Keyword example - example `Out of stock`
 - Use groups to extract just that text - example `/reports.+?(\\d+)/i` returns a list of years only
 - Example - match lines containing a keyword `/. *icecream.*/`
- One line per regular-expression/string match

