

# How to troubleshoot a robot



Written by Henrik Hofmeister  
Updated over a week ago

## How to troubleshoot a robot

This document is your critical resource for troubleshooting unexpected results or execution failures. While our support team is happy to answer questions or offer guidance, check here first for solutions to common issues.

Most problems with crawlers and extractors can be traced to an incorrect CSS selector.

### Robot fails after “Querying dom” message

The execution log shows an entry for *Querying dom:* and immediately after an error message such as *error\_screen|<binary>* appears.

```
Thu May 05 20:07:35 UTC 2016|log|Querying dom: div.review-container:nth-child(8)
> div.review-user > b
Thu May 05 20:07:35 UTC 2016|error_screen|<binary>
Thu May 05 20:07:36 UTC 2016|error_report|<base64>
Thu May 05 20:07:36 UTC 2016|error|Extraction element not found
Thu May 05 20:07:36 UTC 2016|done|false
```

These problems always happen when the scope for a child element in a loop is assigned improperly. Ensure that the selector in the child node takes the parent node's selector into account.

This error usually happens when selecting an element for extraction inside a loop using the *Click to select new element* button located in the step's Edit panel. Be careful when using this button; it will resolve the path to the absolute path (using the document context) and not the relative path (using the loop element context). Most of the time you must delve into the CSS selector to make sure that you're using the right context.

### Solving page iteration problems

When creating a new *Iterate Pages* step, the extractor editor will determine the unique CSS selector representing a page element. However, a common pattern found in paginated Web interface design is that once you navigate to *Page 2*, a *Previous* button appears. This new page element may cause the *Iterate Pages* step's CSS path to become ambiguous or incorrect upon loading *Page 2*. It may even point to the *Previous* button instead of the *Next* button as intended. In practice, this means that the robot will first navigate to *Page 2*, but will then navigate back to *Page 1*. To solve this problem, navigate to *Page 2* in the extractor editor and verify that the CSS selector specified for the *Iterate Pages* step still points to the *Next* button. If it doesn't, modify the selector accordingly. This is often as simple as adding *:last* to the end of the path.

### **A Loop Through Elements step works only on the first page**

The extractor editor tries its best to determine the correct common path for multiple elements, but sometimes it fails to select the optimal choice. If you experience issues where a loop is not working on certain pages - or all pages except the first - investigate the CSS selector of the Loop Through Elements step and ensure it's configured with the correct CSS selector for the list of elements you're looping through.

### **An Element not found error occurs, but I can see the element in the extractor editor**

The extractor editor tries to identify the unique CSS selector for an element, but it can fail. If certain elements aren't found - either in the editor or during your executions - double-check the CSS selectors specified for the relevant steps.

### **No value found for input field**

No input values were configured for the run. Visit the run configuration screen, click the **Inputs** tab, and add input values to the run configuration.

### **No elements found in loop**

The CSS selector specified for the relevant *Loop through elements* step is incorrect. In the extractor editor, edit the step to enter the correct CSS selector in the **Element Path** field.

### **No output available**

The robot failed to extract results, so the *Save Current Output* step had no data to save.

- In the extractor editor, verify the CSS selectors for each suspect extraction step.
- If setting extraction steps' error handling to *\*Ignore & Continue\** throughout an extraction sequence, ensure this setting is applied to the *\*Save Current Output\** step, as well

### **Extraction element not found**

The robot failed to find the page element to extract data from.

- In the extractor editor, verify the CSS selector specified for this extraction step.
- Review the Web page at which the error occurred to ensure the CSS selector pointing to the desired data doesn't differ from the page upon which you based the robot design.

### **Failed to query using CSS selector**

The CSS selector syntax is incorrect.

- In the extractor editor, verify the syntax used in the CSS selector specified for the relevant step.

Did this answer your question?



