Tab 1

Surface accessibility issues in the Issues panel and the accessibility tree

Attention: Externally visible, non-confidential

Author: yanlingwang@microsoft.com, Changhao Han

Status: Draft

Short Link:http://bit.ly/42gQP3z

Created: 2025-03-27 / **Last Updated**: 2025-05-07

This document is part of a detailed section explaining adding squiggles support for accessibility issues within this core document: DOM Element Semantic Error Adorners. For additional context and comprehensive information, please refer to the core document.

Objective

We propose to integrate Lighthouse accessibility audit results into the Issues panel of DevTools, and to show squiggly line warnings for accessibility errors in the Elements panel's accessibility tree.

Signed off by

Name	Approval status
dsv@chromium.org	Review not started
Benedikt Meurer	Review not started
Michael Hablich	Review not started
...	

Overview

Background

Currently, DevTools does not surface accessibility issues in a meaningful way. While we do report some accessibility issues in Console and as experimental squiggly lines in the DOM tree, they are very scarce, and most of the other accessibility issues are not surfaced because they don’t break the page for non-accessibility use cases. The only holistic and entry-level debugging point for accessibility issues is Lighthouse’s accessibility audit, but the user journey from identification to (proposed) fixes contains some frictions. Tracking, managing, and acting upon the identified accessibility issues is hard in the Lighthouse; when you fix an issue proposed in Lighthouse, the audit result does not refresh, and the user cannot know if their fix actually works or not. In other words, Lighthouse is a good place to audit issues, but not fixing them.

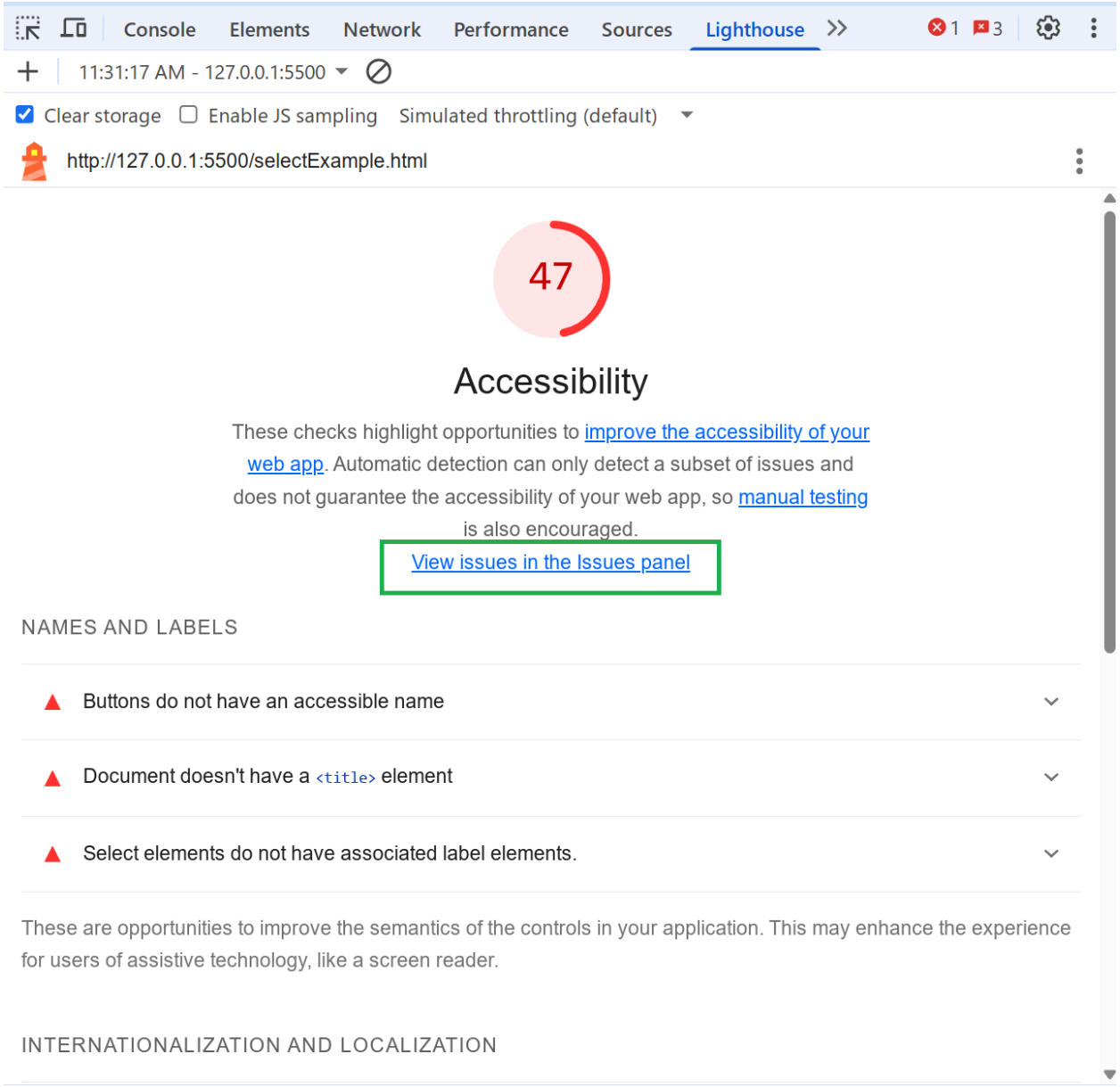
Requirements

This document outlines the design and implementation of two features to solve the friction problem mentioned above. We propose two features:

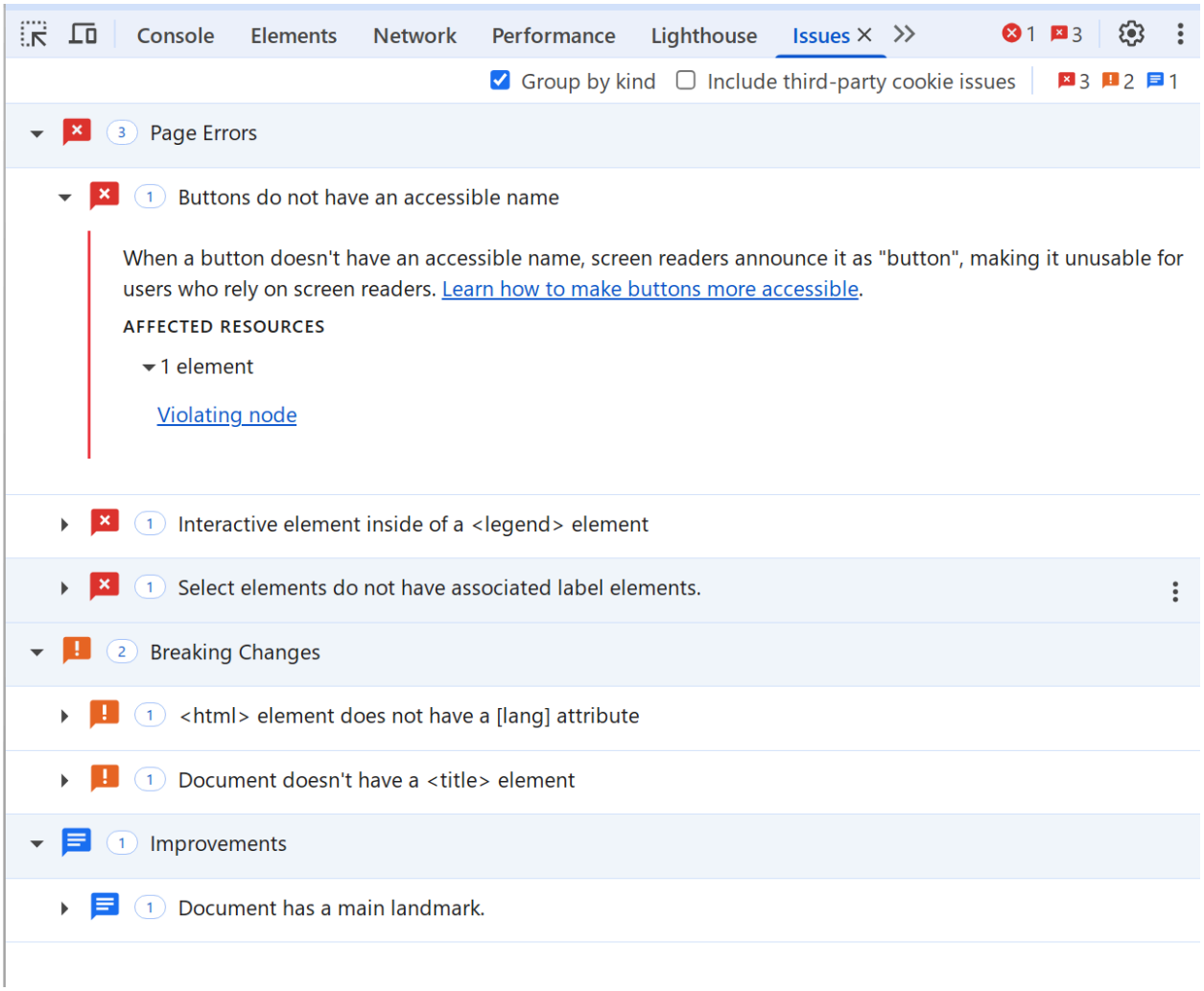
1. Integrate Lighthouse accessibility audit results into the Issues panel of DevTools. The goal is to provide developers with actionable insights into accessibility issues directly within the Issues panel, which is a better place to understand and fix issues. In the long term, when we have platform-surfaced native accessibility issues, they will be funneled to the Issues panel as well, making the Issues panel a central place to understand and fix accessibility issues.
2. Surface accessibility issues via squiggly lines from Lighthouse audit in the accessibility tree. Using the tree facilitates rapid identification of whether you've solved the problem, which has been the go-to way of prototype-fix-verify cycle developers are used to. By enabling this view to present the visual hints that there is an issue, it gives the user the ability to do their analyze-edit-refresh loop focused on their particular area of editing.

Design Sketches

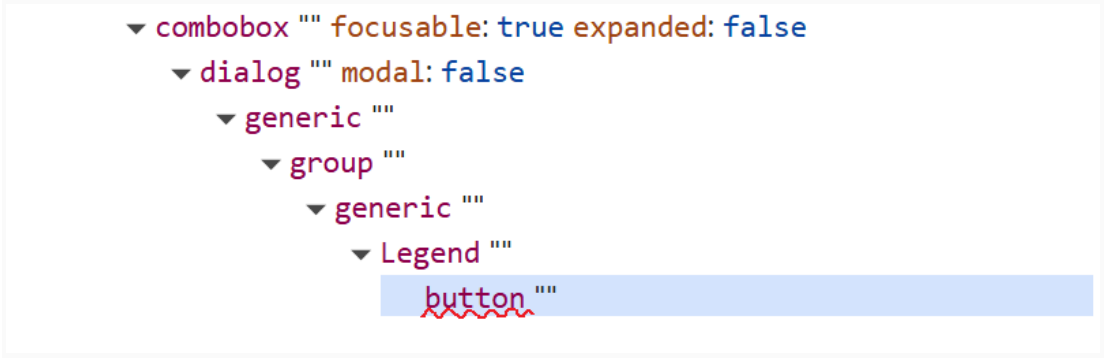
A new link will appear in the Accessibility report, directing users to the Issues panel once Lighthouse results are available. This link is shown only if the user runs a Lighthouse report and accessibility rule violations are detected.


Console
Elements
Network
Performance
Sources
Lighthouse
1
3
11:31:17 AM - 127.0.0.1:5500
Clear storage
Enable JS sampling
Simulated throttling (default)
http://127.0.0.1:5500/selectExample.html
47
Accessibility
These checks highlight opportunities to improve the accessibility of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so manual testing is also encouraged.
View issues in the Issues panel
NAMES AND LABELS
Buttons do not have an accessible name
Document doesn't have a <title> element
Select elements do not have associated label elements.
These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.
INTERNATIONALIZATION AND LOCALIZATION

Accessibility rules that fail during the audit will be flagged as issues and displayed in the Issues panel. Each issue will include detailed information such as the rule that was violated, a description of the issue, its severity, and the specific DOM node that caused the failure.



Adding squiggly line warning support in the AX tree of Elements panel similar to what we have in the DOM tree to improve visibility and usability.



Stretch goal: After the proposed feature is implemented, and based on user feedback, we may consider adding a setting that allows users to toggle the display of squiggly lines in the DOM tree to highlight nodes with accessibility issues.