Team Rocket

CSCD01 Deliverable #4

April 7th, 2020

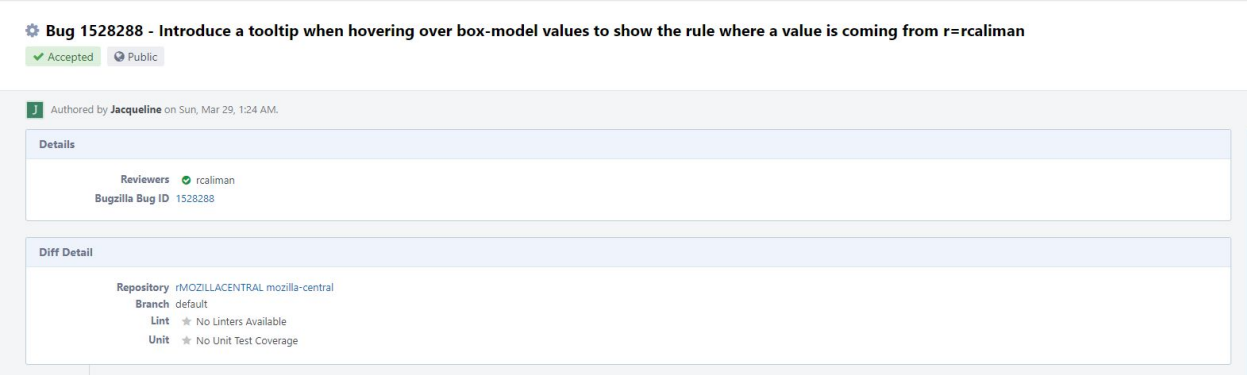
Team Members: Yasir Haque, Adnan Shahid, Patricia Lee,

Sol Han, and Jacqueline Chan

Preface

Our patch is accepted and is pending for deployment. This report outlines the processes we used to develop our successful patch.

https://phabricator.services.mozilla.com/D68708



You can see the exact code changes in that phabricator link. It also has detailed diffs.

To check out our changes for yourself, please use the repo Mercurial with the following instructions:

Checking out on Mercurial

To install mercurial:

On Mac:

- brew install mercurial

On Unix/Linux:

- apt-get install mercurial

Cloning the mercurial repo:

- hg clone https://hg.mozilla.org/mozilla-central

Building:

- run ./mach configure
- run ./mach build
- run ./mach run

Project Feature

To check out our implemented project feature please check out the mozilla central project using mercurial using the steps above and checkout our revisions using the following command:

- hg update -r 520905

Official threads pertaining to our feature are found here:

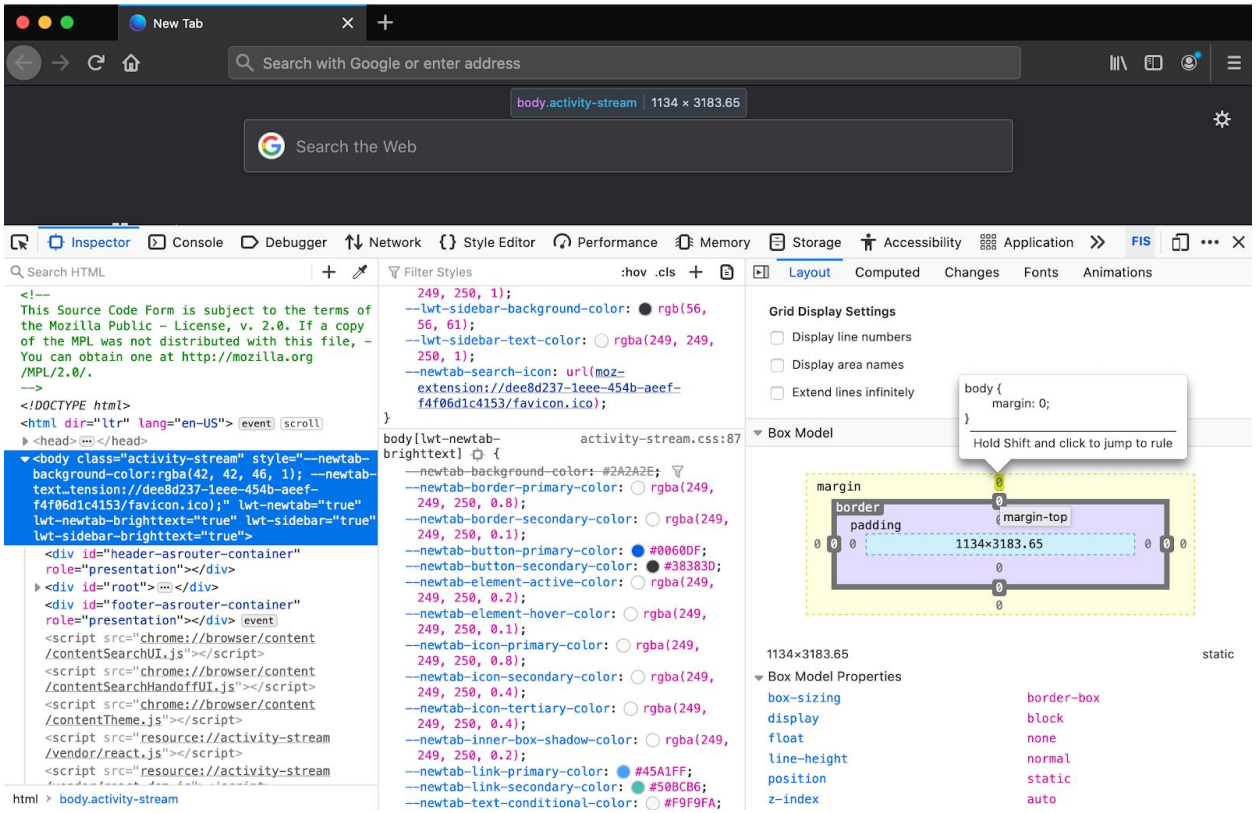
- https://bugzilla.mozilla.org/show_bug.cgi?id=1528288
- https://phabricator.services.mozilla.com/D68708#change-yXY2hZewHp0D

User Guide

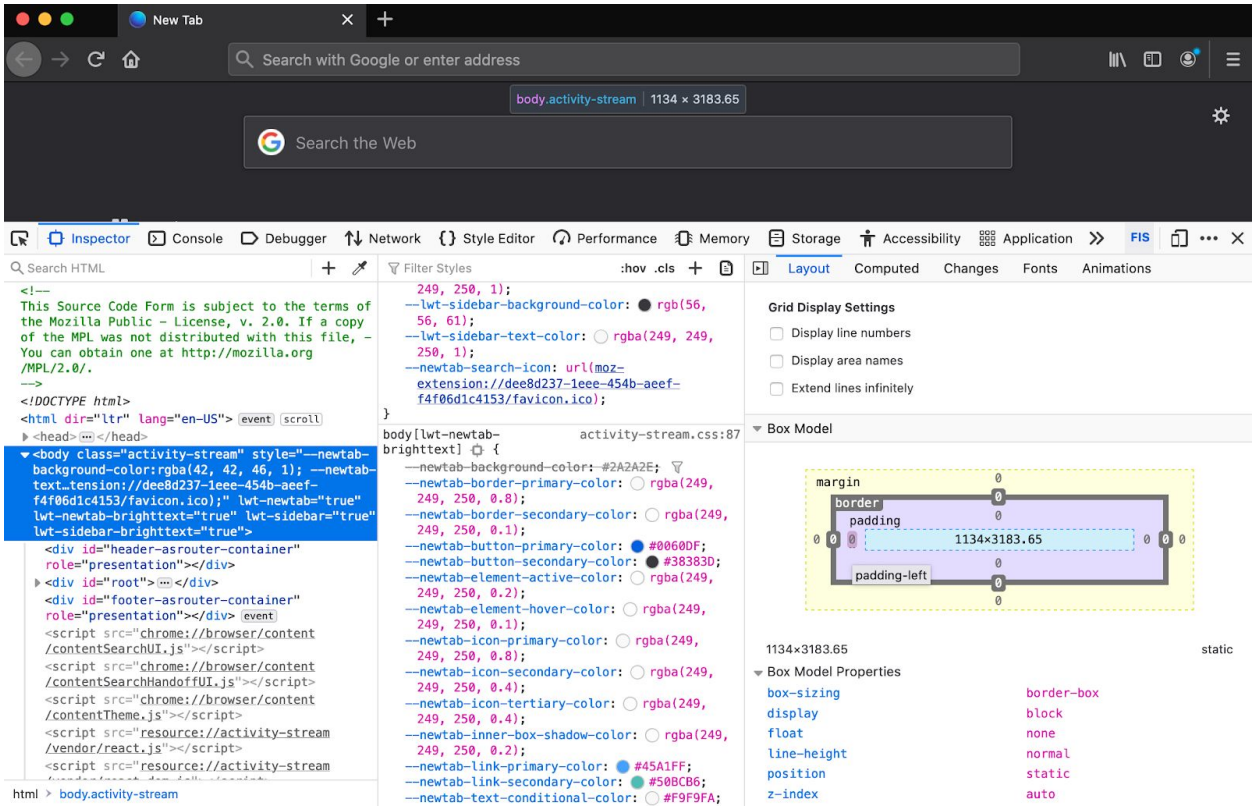
Feature Bugzilla link: https://bugzilla.mozilla.org/show_bug.cgi?id=1528288
Feature Demo: https://i.gyazo.com/432adb57dbe9a1c3927174202b0e0113.gif

This feature involves creating a tooltip that displays a preview of a CSS rule when hovering over a box model value within the Inspector. The CSS rule in the tooltip would only display the declaration associated with the Box Model value being hovered. For example, if you hover over the “padding-left” box model value, the tooltip would only show the declarations that affect “padding-left” and exclude other declarations in the same css rule, if any. To access the feature that was implemented, open the Inspector with F12, go to the Layout tab in the area that looks at the CSS rules.

Hover your mouse over any of the values in the Box Model as shown below:



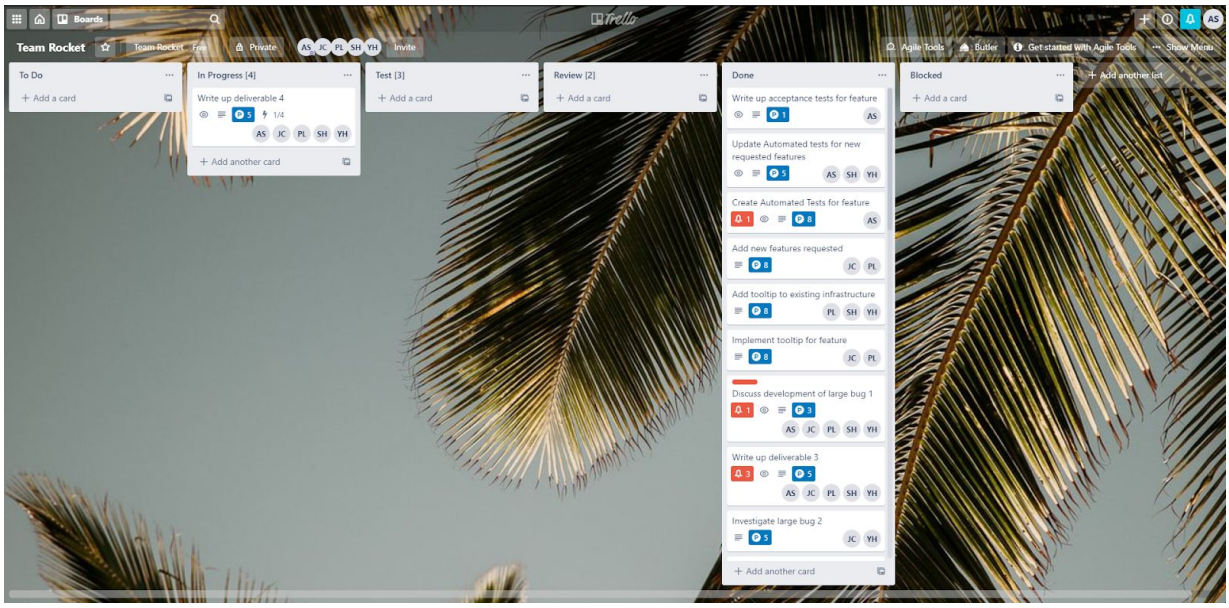
The CSS rule corresponding to the value that was hovered should be displayed in a tooltip. In addition, if a user holds Shift and clicks on the value, the CSS rule should be scrolled to and highlighted in the CSS Rule View panel (to the left of the Box Model). Below is what the behaviour is like when hovering a Box Model value without an associated rule.



Before we implemented the feature, a tooltip would only appear when holding down Shift while hovering over a Box Model value that does not have a CSS rule enforcing it. The tooltip in this case would simply display “No associated rule”, which is not very informative. When implementing the feature, we were asked by the Mozilla developers to remove the “No associated rule” tooltip as well, (and instead don’t show anything).

Software Development Process

Our current Trello board (as of writing this deliverable) is as shown below.



For this deliverable, our workflow mainly consisted of investigating how to develop our feature (a spike), writing acceptance and unit test cases for it and implementing the feature. Our team met up online to do a standup and assigned story points to each of the tasks that needed to be done. Then we pulled the tasks into the corresponding columns as they were being worked on, tested, reviewed and completed. For a couple of days, we were blocked on completing our feature while we waited for a Mozilla developer to get back to us. When he did, he provided more user requirements, and that changed how we were going to implement it. After revisiting the details with the new requirements as a team, we proceeded to develop the feature.

Changes from previous deliverable

From the last deliverable, our team has decided to implement our second feature that we wrote about instead of our first one. This was primarily done because our team received responses from the mozilla community when requesting to develop this feature and have yet to receive a response for our initial feature. In addition, this feature seemed to be more involved as stated by the Mozilla staff member that responded to our request to implement it. Our team chose to develop this feature to make an involved, real contribution to Mozilla that would be easily seen and accessible. Finally, this feature is linked directly to the devtools project, which is the section of Mozilla Firefox that our team had chosen to focus on for the majority of our project.

Acceptance Tests For Our Selected Feature

Test Case 1:

1. Open a new instance of the Firefox browser
2. Press F12 to open an instance of the Inspector
3. In the Inspector, mouse over to the right-most panel with the Layout tab
4. Scroll down the Layout section until you reach the Box Model
5. Hover your mouse over an editable value in the Box Model (i.e. a value for margin, border or padding)
6. Confirm that a tooltip appears that displays the rule for the corresponding CSS property

Test Case 2:

1. Open a new instance of the Firefox browser
2. Press F12 to open an instance of the Inspector
3. In the Inspector, mouse over to the right-most panel with the Layout tab
4. Scroll down the Layout section until you reach the Box Model
5. Hold Shift and click on an editable value in the Box Model (i.e. a value for margin, border or padding)
6. Confirm that the corresponding CSS rule is scrolled to and highlighted in the CSS Rule View Panel

Test Case 3:

1. Open a new instance of the Firefox browser
2. Press F12 to open an instance of the Inspector
3. In the Inspector, mouse over to the right-most panel with the Layout tab
4. Scroll down the Layout section until you reach the Box Model
5. Hover your mouse over each section of the box and confirm that the respective CSS rule is displayed for all Box Model values

Test Case 4:

1. Open a new instance of the Firefox browser
2. Press F12 to open an instance of the Inspector

3. Select and update a CSS rule that is within the Box Model for the current element (i.e. padding, border, or margin)
4. In the Inspector, mouse over to the right-most panel with the Layout tab
5. Scroll down the Layout section until you reach the Box Model
6. Hover your mouse over the changed value of the Box Model
7. Confirm that the displayed CSS rule on the tooltip is correctly updated to reflect the change that was made

Test Case 5:

1. Open a new instance of the firefox browser
2. Open a new tab to get to the **home page**
3. Press F12 to open an instance of the Inspector
4. In the Inspector, mouse over to the right-most panel with the Layout tab
5. Scroll down the layout section until you reach the Box Model
6. Hover your mouse over the padding-left value of the Box Model (this value should not be enforced by a CSS rule)
7. Confirm that no tooltip appears

** We are specifying the home page here because we know that their css has no padding-left value (as of right now), but you can pick any page that you know has no associated rule to it, and test that the tooltip doesn't appear there.

Unit Tests Suite

Mochitest is an automated regression testing framework that utilizes the MochiKit JavaScript library. (The MochiKit library provides tools for handling asynchronous requests AJAX, DOM functionality, and many other features). With JavaScript function calls, Mochitest informs the test harness of any failures/successes. Because Mochitest uses JavaScript function calls to report test successes or failures, it can not be used directly to test some C++ components. However, as all the files we touched were mainly JavaScript files, Mochitest was suitable for us to use to test our feature. Using Mochitest we can imitate things like hovering/clicking and then check that our feature responds to such as actions. Such test cases are described below:

devtools/client/inspector/boxmodel/test/browser_boxmodel_show-tooltip-for-rule.js

- Tests that hovering over a Box Model editable value displays a tooltip with the source CSS rule

devtools/client/inspector/boxmodel/test/browser_boxmodel_jump-to-rule-on-click.js

- Tests that holding Shift while clicking on a Box Model editable value will cause the Rules view panel to jump to the corresponding source CSS rule, and highlight the specific declaration for the clicked value

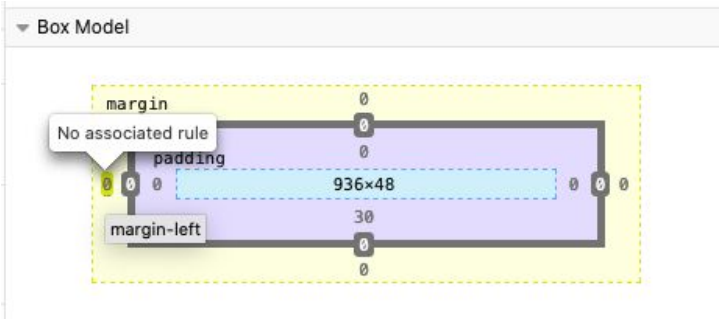
In order to be able to run the test files, we needed to add the new test names to devtools/client/inspector/boxmodel/test/browser.ini

To run our tests, use the following commands:

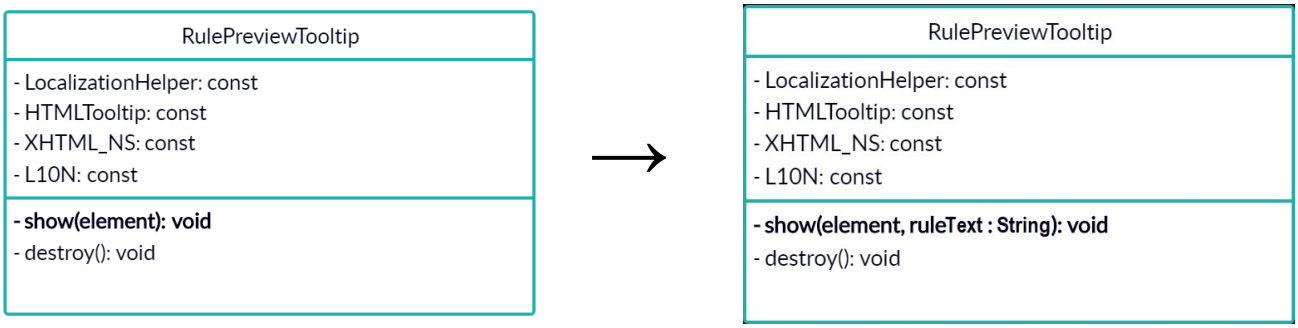
- ./mach test browser_boxmodel_show-tooltip-for-rule.js
- ./mach test browser_boxmodel_jump-to-rule-on-click.js

Code Design

devtools/client/shared/widgets/tooltip/RulePreviewTooltip.js



This component creates and displays the tooltip for the box model through a function called show(). Before our modifications, it could only add tooltips for box model values that had no associated CSS rules with the flavour text “no associated rules” within the tooltip. What we needed to do in this file was change the tooltip message to show the CSS rule and also add a footer in the tooltip saying "Hold Shift and click to jump to rule". Furthermore, with our new feature it was no longer necessary to show a tooltip if no associated rules existed, so we got rid of that as well. We also needed a way to get the CSS rule text in show(), so we added a new parameter to be passed into show(), called “ruleText”. Since the show() function is called in box-model.js, we also needed to modify box-model.js to retrieve the CSS rule text to display.



devtools/client/inspector/boxmodel/box-model.js

In this file, there is a function we needed to modify called onShowRulePreviewTooltip(). Within this function, show() from RulePreviewTooltip is called, so we needed to find a way to retrieve the CSS rule text so it can be passed into show(). Our solution was to create a new function called getApplicableTextProperty() inside the Rule View (rules.js), which returns the TextProperty object associated with a given CSS property name.

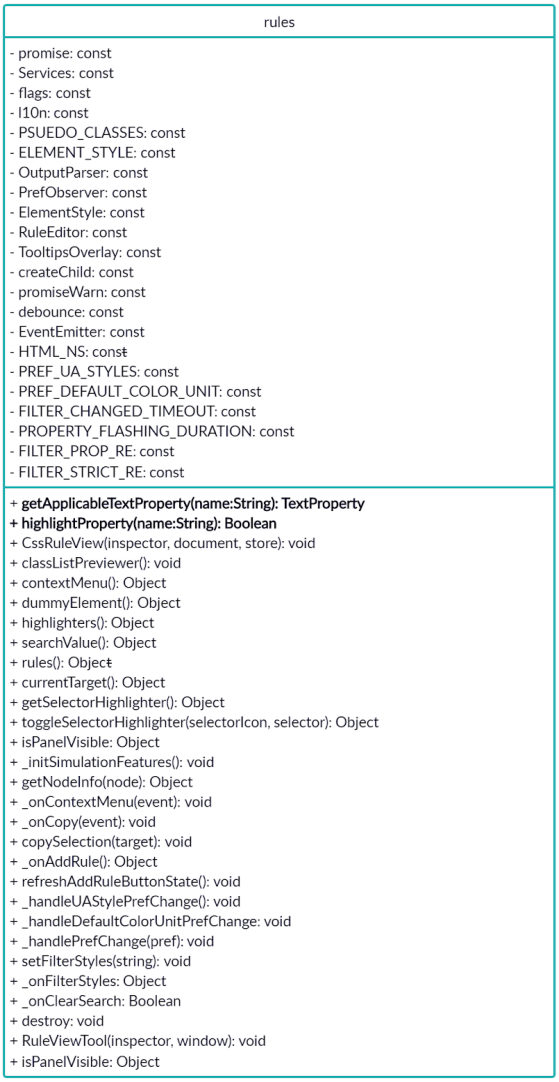
TextProperty objects have a property called “rule” which we can use to get the CSS rule text. We also modified onShowBoxModelEditor(), which is called when a box model editable value is clicked on. In this function, we added code to call an existing function called highlightProperty() from the Rule View (rules.js) when the Shift key is pressed.

devtools/client/inspector/rules/rules.js

This component contains methods relating to the CSS Rule view. It has an existing function highlightProperty() that finds a specified TextProperty in the CSS rule view of the inspector, then jumps to the rule and highlights it. The function was quite complex as it manually finds a TextProperty given a property name, and jumps to the target element for that TextProperty. This function actually had used duplicate code within itself. We refactored the function into 2 simpler functions called getApplicableTextProperty() and highlightProperty(). The function getApplicableTextProperty() finds the TextProperty instance for a given CSS property name. The function highlightProperty() completes the same task as before, but it now makes use of getApplicableTextProperty() to no longer duplicate code. With our new changes, box-model.js can now call getApplicableTextProperty() to retrieve a CSS rule’s text content.

Before:

After:



devtools/client/inspector/boxmodel/components/BoxModelEditable.js

This is a component within the box model that relates to the editable values within the box model. There is a function onMouseOver() that is called when the mouse hovers the editable values. We modified this function so that onShowRulePreviewTooltip() (exists in box-model.js) is called within it.

devtools/client/inspector/rules/models/rule.js

This file contains methods for the Rule object. The stringifyRule() method was modified to now take in a parameter for filtering the declarations in the return string representation of the rule. The new parameter is an optional array of TextProperty objects to filter by. This modification allows for the tooltip to only display the CSS declarations in the rule that is relevant to the property being hovered.

devtools/client/locales/en-US/inspector.properties

In this file, we defined a new localized string "Hold Shift and click to jump to rule", which would be displayed in the tooltip's footer. We also removed the localized string for "No associated rule" as it is no longer needed as per the updated feature requests.

The following test files were deleted:
devtools/client/inspector/boxmodel/test/browser_boxmodel_jump-to-rule-on-hover.js
devtools/client/inspector/boxmodel/test/browser_boxmodel_show-tooltip-for-unassociated-rule.js

They were deleted because the unassociated rule tooltip is now obsolete and the file browser_boxmodel_jump-to-rule-on-hover.js had to be replaced with browser_boxmodel_jump-to-rule-on-click.js.