CodeDiffVis

Thank you for agreeing to participate in this study. This guide briefly explains the features of the software.

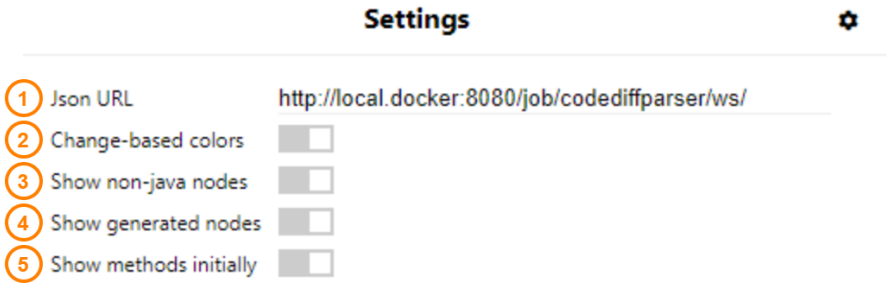
CodeDiffVis uses a so called ContentScript (JavaScript), which will be injected into the webpages. Upon calling a merge request site, the script is activated. The extension uses the «local storage» of Chrome and saves your settings.

If you open a merge request site on GitLab, a separate window will open where the graph will be drawn. The graph is only interactive for as long as you have opened the merge request tab in the browser.

For very big merge requests (>20 files) it is possible that loading the graph takes some time. This is due to the fact, that GitLab is highly resource demanding.

1 SETTINGS

If you are on a merge request site, on «/diff», click on the symbol of CodeDiffVis.

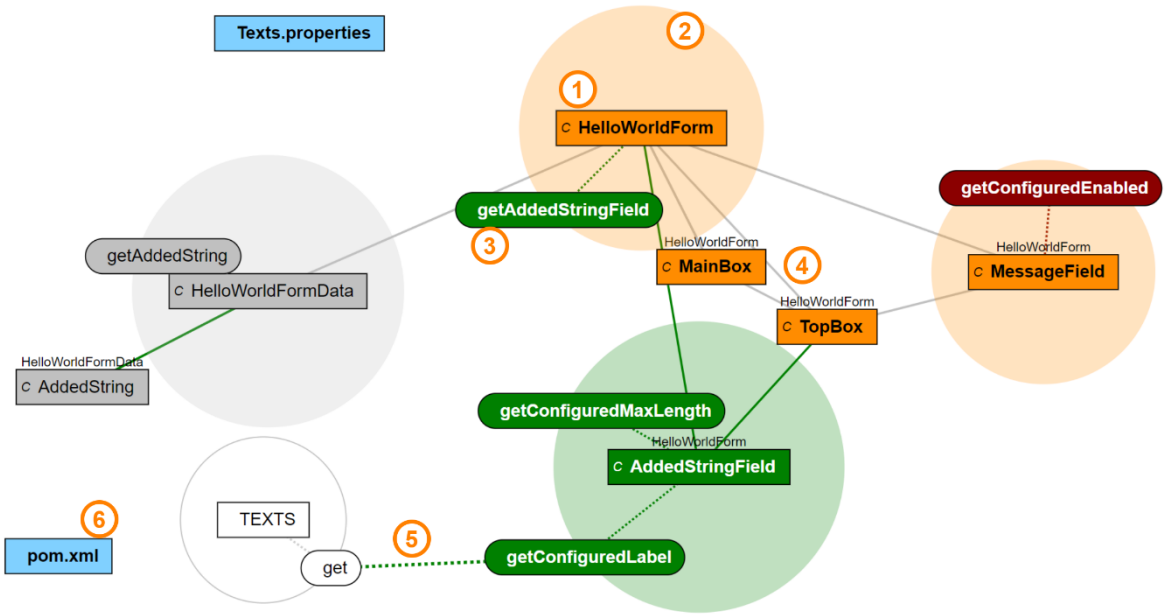


1. Path to Json file. It i seither relative tot he location of CodeDiffVis on your computer or absolute via web url. If no name is specified, the merge request ID is used (e.g. 1234.json). Two examples:
   - out.json
   - http://local.docker:8080/job/codediffparser/ws/
2. Toggle «change-based colors» colors the nodes based on whether they are different in both branches. Disabling this toggle will allow coloring by package. For change-based coloring, the following schema is used:
   - *Ret*: deleted, i.e., this node is only present in the target branch
   - *Green*: new, i.e., this node is only present in the source branch
   - *Orange*: changed, i.e., within this node, there were changes (not counting comments)
   - *Grey*: generated nodes that have the `@Generated` annotation
   - *White*: referenced nodes, i.e., not part of the change
   - *Light blue*: «non-Java nodes », i.e., JavaScript, Properties, XML's etc.
3. Toggle, if all nodes should be drawn or only Java nodes
4. Nodes that have the `@Generated` annotation can be hidden
5. Decides whether methods should be added initially

2 THE GRAPH – BRIEFLY EXPLAINED

The graph is a mix between a «call graph» and a «dependency graph». On the one hand, method calls are drawn and on the other hand dependencies, e.g., class hierarchies, imports or interface references are shown.

We recommend opening the graph on a separate screen and to use the full screen mode (F11).



1. Represents a Java class. Classes are denoted with C, Abstract classes with A and Interfaces with I
2. The circle is added to nodes that contain at least one method in the code change
3. Rounded corners of a node indicate a method
4. Inner classes are denoted with a label on the top with the corresponding class file name
5. Represents a method call. "getConfiguredLabel" of class "AddedStringField" calls "get" from "TEXTS"
6. Nodes that do not represent Java components are shown with their full name

3 INTERACTIONS

The embedded graph is not static. You have a lot of options to configure it to your needs such that it becomes useful for your code review.

For example, you can remove already reviewed nodes from the graph, or you can show only parts of the graph with the hovering function.

Name	*Aktion*	*Beschreibung*
Pan	Move the graph	Hold and click (left) on the free space and move the mouse to move the graph
Drag	Move nodes	Change positions of nodes. Drag a node to your desired location
Zoom	Zooming	Use the mouse wheel to zoom in and out
SVG resize	Enlarge / shrink the view	You can resize the window arbitrary
Hover	Highlight connections	Hover over a node with the mouse; connected nodes will be highlighted
Hover reverse	Highlight nodes	Hover over the source code in the merge request; the node and all connected nodes are highlighted in the graph. The graph automatically centers your node
Hover lock	Pause highlight	With **CTRL** you can lock / unlock the current highlight to prevent flickering
Link	Jump to declaration	On **click** (left) on a node, jump to the first occasion in the code change directly. A node remembers being clicked and changes its appearance to make it easier for you to track which nodes are already reviewed
Expand	Add referenced nodes	With **SHIFT + click** (left) referenced nodes can be added
Remove	Remove nodes	Remove any node with a **click** (right)