

CodeDiffVis

Vielen Dank dass du dich bereiterklärst bei der Studie teilzunehmen. Dieser Guide erklärt kurz die Features der Software.

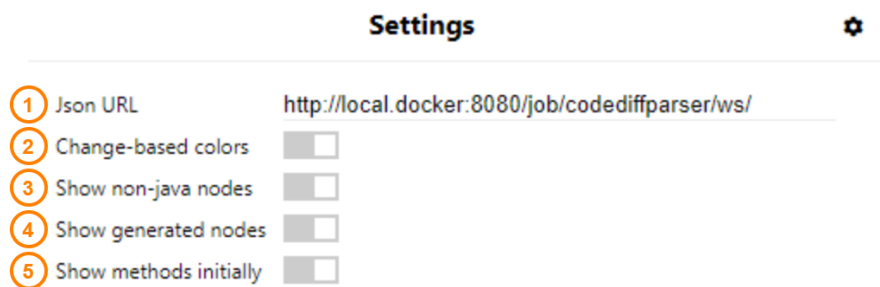
CodeDiffVis verwendet ein sogenanntes ContentScript, also JavaScript, welches in die Webseiten injiziert wird. Sobald man eine Gitlab Merge-Request Seite aufruft, wird das Skript aktiv. Weiter verwendet die Extension den sogenannten «local storage» und speichert so deine Einstellungen.

Beim Aufrufen einer Merge-Request Seite öffnet sich ein zusätzliches Browserfenster mit dem Graf. Dieser ist nur so lange interaktiv, wie das zugehörige Tab im Browser geöffnet ist.

Bei sehr grossen Merge-Requests (>20 Files) kann es zu Verzögerungen beim Laden des Grafs kommen. Dies liegt daran, dass Gitlab bei so grossen Changes generell sehr ressourcenlastig ist.

1 SETTINGS

Wenn du auf einer Merge-Request Seite auf /diff bist, klicke auf das Symbol von CodeDiffVis.

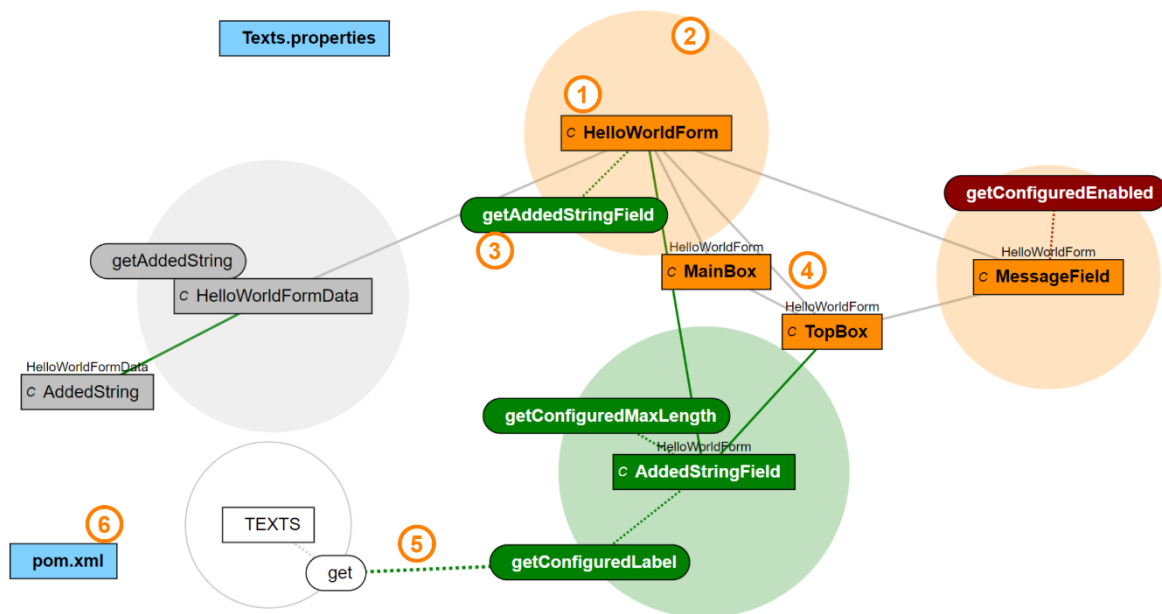


1. Pfad zum Json File. Dieser ist entweder relativ zur Location vom CodeDiffVis auf deinem Computer oder absolut via Web URL. Wird der Filename weggelassen, wird die aktuelle Merge-Request ID verwendet (z.B. 1234.json). Zwei Beispiele:
 - out.json
 - <http://local.docker:8080/job/codediffparser/ws/>
2. Toggle «Change-based colors» färbt die Knoten anhand davon ein, ob sie innerhalb beider Branches geändert haben. Ist der Toggle aus, wird anhand der Java-Packages eingefärbt. Für die change-based Färbung gelten folgende Farbcodes:
 - *Rot*: gelöscht, d.h. der Knoten kommt nur im target Branch vor
 - *Grün*: neu, d.h. der Knoten kommt nur im source Branch vor
 - *Orange*: geändert, d.h. innerhalb vom Knoten gab es Änderungen (Kommentare sind ausgenommen)
 - *Grau*: generierte Knoten mit der Annotation @Generated
 - *Weiss*: referenzierte Knoten, d.h. nicht teil vom Change
 - *Hellblau*: «nicht-Java-Knoten», d.h. JavaScript, Properties, XML's etc.
3. Toggle, ob alle Files als Knoten angezeigt werden sollen oder nur Java Files
4. Java Knoten mit der @Generated Annotation werden ein- bzw. ausgeblendet
5. Steuert, ob Methoden angezeigt werden sollen. Du kannst sie später noch hinzufügen

2 DER GRAF – KURZ ERKLÄRT

Der dargestellte Graf ist eine Mischung aus einem «call graph» und einem «dependency graph». Zum einen werden Methodenaufrufe dargestellt, zum anderen sind einfache Abhängigkeiten, wie z.B. Klassenhierarchien, direkte Importe oder Interface Referenzen ersichtlich.

Wir empfehlen den Graf auf einem separaten Bildschirm anzuzeigen und diesen mit F11 im Vollbildmodus zu öffnen.



1. Eine Java Klasse «HelloWorldForm.java». Innere oder Referenzierte Klassen werden mit einer Linie (Link) verbunden. Das Label **C** bezeichnet dabei eine ordinäre Klasse, **A** steht für eine abstrakte Klasse und **I** für ein Interface
2. Der «Einflussbereich» der Klasse. Innere Methoden bleiben in diesem Kreis gefangen, wogegen alles nicht Zugehörige abgestossen wird
3. Gerundete Ecken eines Knoten stehen für Methoden
4. Die Hauptklasse (Filename) wird als Label oberhalb von inneren Klassen gezeigt
5. Ein Methodenaufruf: die Methode «getConfiguredLabel» der Klasse «AddedStringField» ruft «get» von «TEXTS» auf
6. Knoten, welche keine Java Komponenten sind (hellblau), werden mit vollem Namen angezeigt

3 INTERAKTIONEN

Der eingebettete Graf ist nicht statisch. Du kannst ihn beliebig konfigurieren, sodass er deinen Ansprüchen gerecht und somit möglichst nützlich für dein Code-Review wird.

Zum Beispiel kannst du bereits reviewte Knoten aus dem Graf entfernen, oder nur ein Teil des Grafen anzeigen mit der Hover-Funktion.

<i>Name</i>	<i>Aktion</i>	<i>Beschreibung</i>
<i>Pan</i>	Verschieben vom Graf	Auf eine freie Fläche mittels linker Maustaste haltend den Graf verschieben
<i>Drag</i>	Verschieben von Knoten	Die Position von Knoten lässt sich beliebig verändern. Ziehe dafür den gewünschten Knoten mit der linken Maustaste an die Wunschstelle
<i>Zoom</i>	Zoomen	Mit dem Mausrad hinein und herauszoomen
<i>SVG resize</i>	Vergrössern / verkleinern des SVG	Mit dem Vergrössern und Verkleinern des Fensters lässt sich das SVG vergrössern oder verkleinern
<i>Hover</i>	Verbindungen hervorheben	Mit der Maus über einen Knoten fahren, so werden alle direkten Verbindungen hervorgehoben
<i>Hover reverse</i>	Knoten hervorheben	Anhand der Mausposition über dem Code im Review werden die entsprechenden Knoten im Graf hervorgehoben. Der Graf springt automatisch zur korrekten Stelle
<i>Hover lock</i>	Hervorheben pausieren	Mit CTRL lässt sich das Hervorheben einrasten. Erneutes Drücken löst die Einrastung
<i>Link</i>	Zur Codestelle springen	Mit einem Klick (linke Maustaste) springt man zur ersten Stelle im Code, wo die Methode oder Klasse definiert oder referenziert wird. Der Knoten merkt sich, dass sie bereits angeklickt wurde
<i>Expand</i>	Verknüpfte Knoten hinzufügen	Mit SHIFT + Klick (linke Maustaste) werden verknüpfte Knoten hinzugefügt. Zum Beispiel Methoden der Klasse oder referenzierte Methoden
<i>Remove</i>	Knoten entfernen	Entferne Knoten mit einem Rechtsklick