


Process de revue

1. Revue dans Gitlab

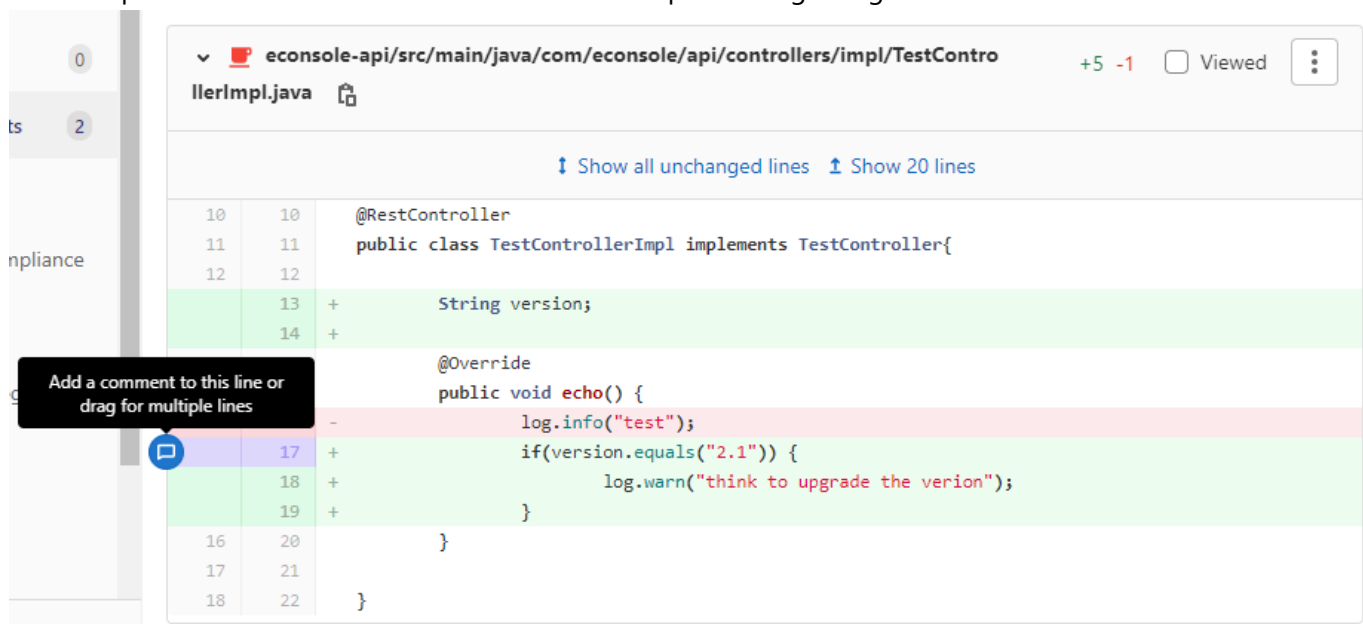
Gitlab fournit des outils intéressants pour les merge requests. La première étape importante consiste à créer une merge request et à affecter le reviewer en tant que **Assignee**

2. Soumettre des informations

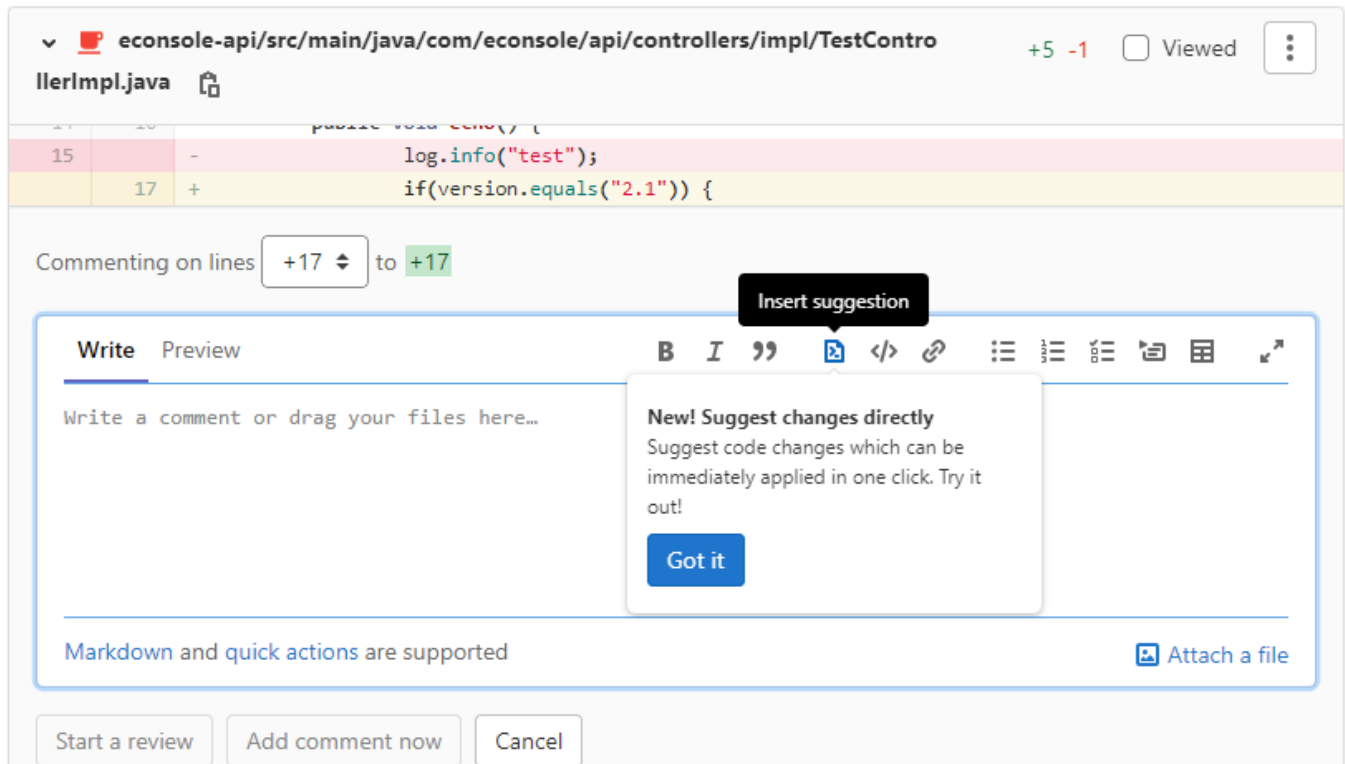
Lorsque le reviewer reçoit la *merge request*, il peut vérifier les changements de code dans l'onglet *Changes*.

 Code Changes

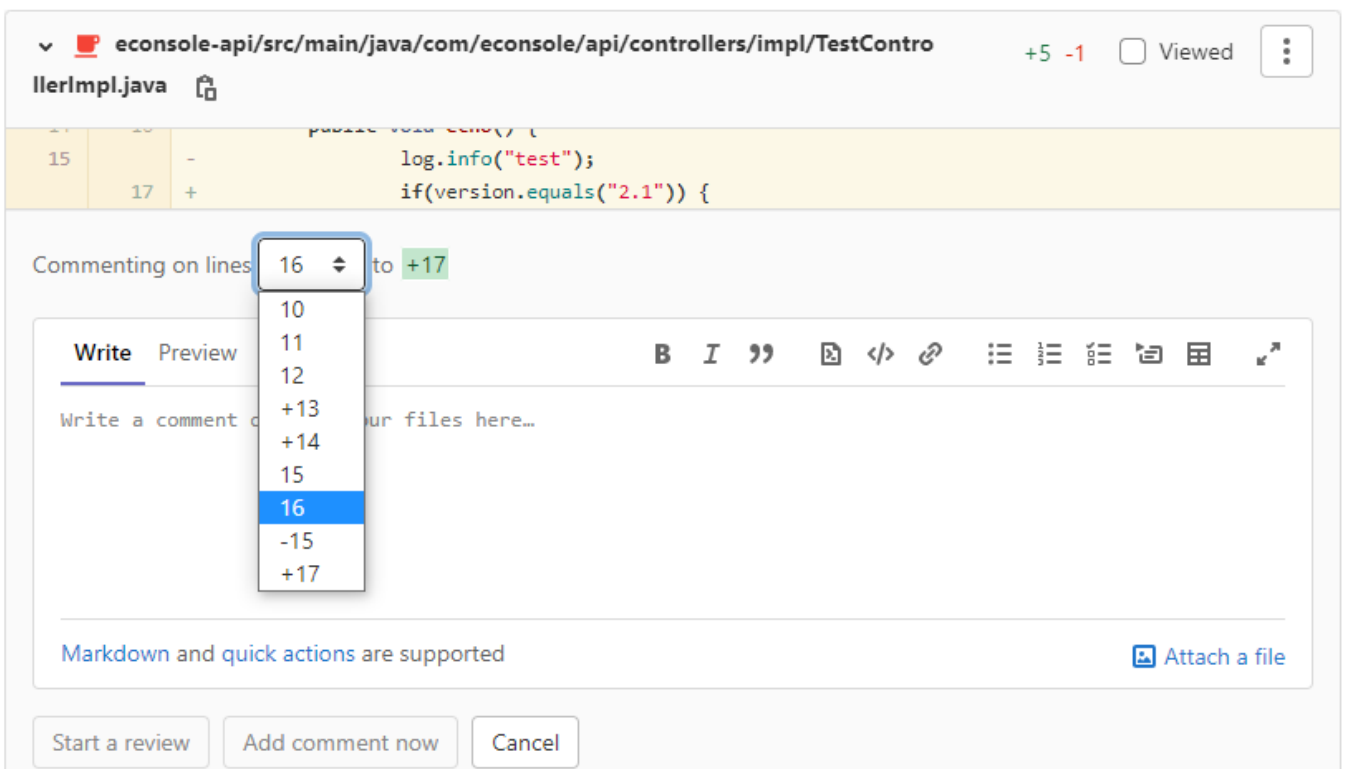
Si une ligne ou une partie de code nécessite plus d'informations, ou si certaines modifications le suggèrent, le reviewer peut mettre des commentaires dans le tooltip à la marge de gauche.



Une popup permet d'écrire une indication en texte intégral ou en markdown. Markdown est utile pour épingler un morceau de code et expliquer pourquoi c'est important et comment le résoudre.



Si le commentaire concerne un bloc de code, il peut cliquer sur la dernière ligne et reprendre les lignes précédentes, ou ajouter des lignes après. Le code pointé sera coloré en **jaune**.



Une fois les commentaires rédigés, le reviewer peut appuyer sur le bouton "**Start a review**" ou "**Add comment now**"

▼ econsole-api/src/main/java/com/econsole/api/controllers/impl/TestContro
llerImpl.java

+5 -1 ☐ Viewed

17 + if(version.equals("2.1")) {

Commenting on lines 16 to +17

Write Preview **B I ”**

In comparaiso statement with "equals", the litteral should be on the left :

```
```java
if("2.1".equals(version)) {
 log.warn("Think to upgrade the version!");
}
```
```

And what about NULL value of version attribute !?

Markdown and quick actions are supported Attach a file

[Start a review](#) [Add comment now](#) [Cancel](#)

Le reviewer peut ajouter des éléments à la revue selon ses besoins pour chaque partie de code nécessitant plus de précisions. Le nombre de commentaires est incrémenté par exemple, ici il n'y a qu'un seul élément.

Overview 0 Commits 1 **Changes 2**

Compare and 2 files +9 -5

▼ econsole-api/src/main/java/com/econsole/api/controllers/impl/TestContro
llerImpl.java

+5 -1 ☐ Viewed

15 - log.info("test");

17 + if(version.equals("2.1")) {

Pending

In comparaiso statement with "equals", the litteral should be on the left :

```
if("2.1".equals(version)) {
    log.warn("Think to upgrade the version!");
}
```

And what about NULL value of version attribute !?

[Submit review 1](#) [Add comment now](#)

18 + log.warn("think to upgrade the verion");

[Pending comments ^](#) [Submit review 1](#)

Lorsque la revue est terminée, le reviewer n'a plus qu'à cliquer sur "Finish review " et les éléments sont soumis dans la merge request.

Afin d'informer le développeur, il est préférable de changer le "**Assignee**" avec la section d'édition de la merge request.

3. Correction des éléments

README.md Le développeur peut vérifier l'état de la merge request. S'il y a des commentaires, une nouvelle section s'affiche avec le libellé " **unresolved threads**"

JIRA-XXX : test pour rédaction du process de revue de code!


Overview 1 Commits 1 Changes 2

1 unresolved thread



Le développeur doit corriger et répondre à chaque question posée dans les commentaires. Puis, il peut commiter et pousser le code dans la branche concernée.

Une fois tous les éléments traités, le développeur peut réaffecter la merge request au reviewer.

 Code Changes comment a bloc of code