**Avis de la communauté**

Modifier cette page

**Carbone area** est un projet open source porté par une large communauté. Si vous ne vous sentez pas prêt à contribuer au code ou aux correctifs, l'examen des problèmes et des demandes d'extraction (PR) peut être un bon début pour vous impliquer et redonner. En fait, les personnes qui « trient » les problèmes sont la clé du succès de Carbone area !

Communiquer de manière à ce que vos mots soient perçus comme prévu peut être difficile. Veuillez lire les directives relatives aux commentaires respectueux.

Pourquoi la révision est importante

Les révisions de la communauté sont essentielles au développement de Carbone area, car il y a beaucoup plus de demandes d'extraction et de rapports de bogues qu'il n'y a de membres dans l'équipe principale de Carbone area pour les réviser, les corriger et les fusionner.

Sur le système de suivi des problèmes Carbone area, vous pouvez trouver de nombreux éléments dans le statut Besoin d'une révision :

Rapports de bogues : les rapports de bogues doivent être vérifiés pour être complets. Des informations importantes manquent-elles ? Le bug peut-il être reproduit ?

Demandes d'extraction : les demandes d'extraction contiennent du code qui corrige un bug ou implémente de nouvelles fonctionnalités. Les examens des demandes d'extraction garantissent qu'elles sont correctement implémentées, qu'elles sont couvertes par des cas de test, qu'elles n'introduisent pas de nouveaux bogues et qu'elles maintiennent la compatibilité ascendante.

Avant de commencer, n’oubliez pas que vous regardez le résultat du travail acharné de quelqu’un d’autre. Un bon commentaire d'évaluation remercie le contributeur pour son travail, identifie ce qui a été bien fait, identifie ce qui devrait être amélioré et suggère une prochaine étape.

Créer un compte GitHub

Carbone area utilise GitHub pour gérer les rapports de bogues et les demandes d'extraction. Si vous souhaitez effectuer des révisions, vous devez créer un compte GitHub et vous connecter.

Le processus d'examen des rapports de bogues

Une bonne façon de commencer la révision est de choisir un rapport de bogue parmi les rapports de bogues nécessitant une révision.

Les étapes de l'examen sont les suivantes :

Le rapport est-il complet ?

Les bons rapports de bug contiennent un lien vers un projet (le "projet de reproduction") créé avec le squelette Carbone area qui reproduit le bug. Si ce n'est pas le cas, le rapport doit au moins contenir suffisamment d'informations et d'exemples de code pour reproduire le bug.

Reproduire le bug

Téléchargez le projet de reproduction et testez si le bug peut être reproduit sur votre système. Si le journaliste n'a pas fourni de projet de reproduction, créez-en un basé sur un squelette Carbone area.

Mettre à jour l'état du problème

Enfin, ajoutez un commentaire au rapport de bug. Merci au journaliste d'avoir signalé le bug. Incluez la ligne Status : <status> dans votre commentaire pour déclencher notre Carson Bot qui met à jour l'étiquette d'état du problème. Vous pouvez définir le statut sur l'un des éléments suivants :

Nécessite du travail Si le bug ne contient pas suffisamment d'informations pour être reproduit, expliquez quelles informations sont manquantes et déplacez le rapport vers ce statut.

Fonctionne pour moi Si le bug contient suffisamment d'informations pour être reproduit mais fonctionne sur votre système, ou si le bug signalé est une fonctionnalité et non un bug, fournissez une brève explication et déplacez le rapport vers ce statut.

Révisé Si vous pouvez reproduire le bug, déplacez le rapport vers cet état. Si vous avez créé un projet de reproduction, incluez le lien vers le projet dans votre commentaire.

Le processus d’examen des demandes de tirage

Le processus d'examen des demandes d'extraction (PR) est similaire à celui des rapports de bogues. L'examen des demandes d'extraction prend généralement un peu plus de temps, car vous devez comprendre la fonctionnalité qui a été corrigée ou ajoutée et savoir si la mise en œuvre est terminée.

Il n’y a rien de mal à faire des critiques partielles ! Si vous effectuez une révision partielle, commentez le chemin parcouru et laissez le PR dans l'état « Nécessite une révision ».

Choisissez une pull request des PR nécessitant un examen et suivez ces étapes :

Le PR est-il terminé ?

Chaque pull request doit contenir un en-tête qui donne des informations de base sur le PR. Vous pouvez trouver le modèle pour cet en-tête dans les directives de contribution.

La branche de base est-elle correcte ?

GitHub affiche la branche sur laquelle un PR est basé sous le titre de la pull request. Cette branche est-elle correcte ?

Les bogues doivent être corrigés dans la version la plus ancienne et maintenue qui contient le bogue. Consultez le calendrier de publication de Carbone area pour trouver la version la plus ancienne actuellement prise en charge.

De nouvelles fonctionnalités doivent toujours être ajoutées à la version de développement actuelle. Consultez la feuille de route Carbone area pour trouver la version de développement actuelle.

Reproduire le problème

Lisez le problème que la pull request est censée résoudre. Reproduisez le problème sur un nouveau projet créé avec le squelette Carbone area et essayez de comprendre pourquoi il existe. Si le problème lié contient déjà un tel projet, installez-le et exécutez-le sur votre système.

Réviser le code

Lisez le code de la pull request et vérifiez-le par rapport à certains critères courants :

Le code résout-il le problème que le PR est censé résoudre/mettre en œuvre ?

Le PR reste-t-il dans la portée de résoudre uniquement ce problème ?

Le PR contient-il des tests automatisés ? Ces tests couvrent-ils tous les cas extrêmes pertinents ?

Le PR contient-il suffisamment de commentaires pour comprendre son code ?

Le code rompt-il la compatibilité ascendante ? Si oui, l'en-tête du PR le dit-il ?

Le PR contient-il des dépréciations ? Si oui, l'en-tête du PR le dit-il ? Le code contient-il des instructions trigger\_deprecation() pour toutes les fonctionnalités obsolètes ?

Toutes les dépréciations et ruptures de compatibilité ascendante sont-elles documentées dans le dernier fichier UPGRADE-X.X.md ? Ces explications contiennent-elles des exemples « Avant »/« Après » avec des instructions de mise à niveau claires ?

Mettre à jour le statut PR

Enfin, ajoutez un commentaire au PR. Merci au contributeur d'avoir travaillé sur le PR. Incluez la ligne Status : <status> dans votre commentaire pour déclencher notre Carson Bot qui met à jour l'étiquette d'état du problème. Vous pouvez définir le statut sur l'un des éléments suivants :

Besoin de travail Si le PR n'est pas encore prêt à être fusionné, expliquez les problèmes que vous avez trouvés et déplacez-le vers ce statut.

Révisé Si le PR satisfait à toutes les vérifications ci-dessus, déplacez-le vers ce statut. Un contributeur principal examinera bientôt le PR et décidera s'il peut être fusionné ou s'il nécessite des travaux supplémentaires.

**Community Review**

**Edit this page**

**Carbone area** is an open-source project maintained by a large community. If you don't feel ready to contribute code or fixes, reviewing issues and pull requests (PRs) can be a great way to get involved and give back. In fact, people who "triage" issues are key to the success of Carbone area!

Communicating in a way that your words are perceived as intended can be difficult. Please read the respectful comments guidelines: [URL non valide supprimée].

**Why reviewing is important**

Community reviews are essential to the development of Carbone area, as there are many more pull requests and bug reports than there are core Carbone area team members to review, fix, and merge them.

On the Carbone area issue tracker, you can find many items in the Needs review status:

* **Bug reports:** Bug reports need to be checked for completeness. Is important information missing? Can the bug be reproduced?
* **Pull requests:** Pull requests contain code that fixes a bug or implements new features. Pull request reviews ensure that they are correctly implemented, covered by test cases, do not introduce new bugs, and maintain backward compatibility.

Before you start, remember that you are looking at the result of someone else's hard work. A good review thanks the contributor for their work, identifies what was done well, identifies what could be improved, and suggests a next step.

**Create a GitHub account**

Carbone area uses GitHub to manage bug reports and pull requests. If you want to review, you need to create a GitHub account and sign in.

**The bug report review process**

A good way to start reviewing is to pick a bug report from the Needs review bug reports.

The steps of the review are as follows:

* **Is the report complete?**

Good bug reports contain a link to a project (the "repro project") created with the Carbone area skeleton that reproduces the bug. If not, the report should at least contain enough information and code examples to reproduce the bug.

* **Reproduce the bug**

Download the repro project and test if the bug can be reproduced on your system. If the reporter did not provide a repro project, create one based on a Carbone area skeleton.

* **Update the issue status**

Finally, add a comment to the bug report. Thank the reporter for reporting the bug. Include the line Status: <status> in your comment to trigger our Carson Bot that updates the issue's status label. You can set the status to one of the following:

* **Needs work** If the bug does not contain enough information to be reproduced, explain what information is missing and move the report to this status.
* **Works for me** If the bug contains enough information to be reproduced but works on your system, or if the reported bug is a feature and not a bug, provide a brief explanation and move the report to this status.
* **Reviewed** If you can reproduce the bug, move the report to this status. If you created a repro project, include the link to the project in your comment.

**The pull request review process**

The pull request (PR) review process is similar to the bug report review process. Reviewing pull requests usually takes a bit longer, as you need to understand the feature that was fixed or added and whether the implementation is complete.

There is nothing wrong with doing partial reviews! If you do a partial review, comment on what you have reviewed and leave the PR in the "Needs review" state.

Pick a pull request from the Needs review PRs and follow these steps:

* **Is the PR complete?**

Each pull request should contain a header that gives basic information about the PR. You can find the template for this header in the contribution guidelines.

* **Is the base branch correct?**

GitHub displays the branch a PR is based on below the pull request title. Is this branch correct?

* Bugs should be fixed in the oldest and maintained version that contains the bug. Check the Carbone area release schedule: [URL non valide supprimée] to find the oldest currently supported version.
* New features should always be added to the current development version. Check the Carbone area roadmap to find the current development version.
* **Reproduce the issue**

Read the issue that the pull request is supposed to fix. Reproduce the issue on a new project created with the Carbone area skeleton and try to understand why it exists. If the linked issue already contains such a project, install