

# 05. Premiers pas dans une communauté open source

## Objectifs de la séance

- Identifier des projets ouverts aux contributions.
- Découvrir les outils de collaboration (Git, GitHub/GitLab).

## 1. “Good first issue” et autres portes d’entrée

Contribuer à l’open source peut sembler intimidant au départ. Heureusement, de nombreux projets balisent des chemins pour accueillir les nouveaux contributeurs.

### Le label `good first issue`

- Utilisé sur GitHub/GitLab pour signaler des tâches simples et pédagogiques.
- Exemples : corriger une faute dans la documentation, améliorer un test, reformater un fichier de configuration.

### Autres points d’entrée

- `help wanted` : issues où la communauté attend activement des contributions.
- Documentation officielle (souvent une section *Contributing*).
- Canaux de discussion (Discord, Slack, mailing lists) où poser des questions.
- Sites d’agrégation comme [firstcontributions.github.io](https://firstcontributions.github.io) ou [goodfirstissue.dev](https://goodfirstissue.dev).

## 2. Interaction avec une communauté

Contribuer, ce n’est pas seulement écrire du code, c’est **communiquer**.

## Issues

- Ouvrir une issue claire : décrire le bug, le contexte, la manière de le reproduire.
- Éviter les jugements (“ça ne marche pas”), préférer la précision (“Dans la version 2.3.1, l’API renvoie une erreur 500 avec cet input...”).

## Discussions

- Poser des questions dans les forums, les channels Slack/Discord ou GitHub Discussions.
- Toujours vérifier si la question n’a pas déjà été posée.

## RFC (*Request for Comments*)

- Utilisées dans les projets matures (ex. Rust, React) pour proposer de nouvelles fonctionnalités.
- Format structuré : motivation, alternatives, proposition détaillée, impact attendu.

# 3. Bonnes pratiques de communication

- **Clarté et concision** : messages structurés, titres explicites.
- **Respect et bienveillance** : pas d’attaques personnelles, respecter les codes de conduite.
- **Patience** : les mainteneurs sont souvent bénévoles. Les réponses peuvent prendre du temps.
- **Documentation des actions** : toujours laisser une trace écrite (issue, commit message).

Astuce : avant d’ouvrir une issue, lisez le `CONTRIBUTING.md` et cherchez les templates proposés. Cela montre que vous respectez les règles du projet.

## 4. Mini-atelier

1. Explorer GitHub/GitLab avec le filtre `good first issue`.
2. Sélectionner un projet qui vous intéresse (ex. framework, outil, librairie).
3. Lire son `README.md` et son `CONTRIBUTING.md`.
4. Identifier une première tâche faisable et noter les règles de communication spécifiques au projet.

## 5. Livrable de la séance

Sélection d'un projet open source à suivre + description des règles de contribution (CONTRIBUTING.md).

Le document doit contenir :

- Nom et lien vers le projet choisi.
- Motif du choix (intérêt personnel, technologie utilisée, domaine).
- Extrait ou résumé des règles de contribution.
- Identification d'au moins une issue accessible ( `good first issue` ou équivalent).

## 6. Pour aller plus loin

- *GitHub Docs – About issues and pull requests.*
- *First Contributions* : guide interactif pour votre première PR.
- *Good First Issue* (<https://goodfirstissue.dev>) : moteur de recherche d'issues pour débutants.
- Exemples de projets accueillants : **Django**, **Symfony**, **Kubernetes**, **Mozilla Firefox**.

## 7. Mini-FAQ

- **Dois-je être expert pour contribuer ?**

Non. Beaucoup de contributions utiles sont de la documentation, des tests, des corrections mineures.

- **Que faire si je ne comprends pas une issue ?**

Poser des questions poliment, en montrant ce que vous avez déjà cherché ou testé.

- **Que faire si ma PR est refusée ?**

Lire le feedback, améliorer la PR, ou accepter que la direction du projet est différente.  
Dans tous les cas, vous progressez.