

Formation Git

Jeff Martins-Jacquelot



git

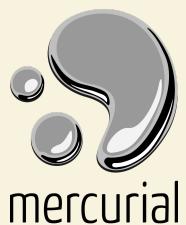
“ Git est un logiciel de gestion de versions décentralisé [...]”

Wikipédia

Quelques informations sur Git

- Première version du logiciel 7 avril 2005 ;
- Écrit par *Linus Torvalds*, auteur du noyau Linux ;
- En 2016 : devient le logiciel le plus populaire ;
- A ne pas confondre avec Github qui lui héberge des dépôts Git (et bien plus encore ...)

Concurrents



**Mais avant tout,
versionner c'est quoi ?**



Mais avant tout, versionner c'est quoi ?

Ça peut se traduire de cette manière ...

Nom	Date de modification	Taille
Couverture v1.pdf	hier à 15:33	57 Ko
Couverture v2.pdf	hier à 15:33	57 Ko
Couverture v3.pdf	hier à 15:33	57 Ko

Pour chaque version :

- Poids du fichier ;
- Date de création (version originale uniquement) ;
- Date de modification ;
- Auteur de la création (version originale uniquement) ;
- Auteur de la modification

Pourquoi utiliser un logiciel de gestion de versions ?

Pourquoi utiliser un logiciel de gestion de versions ?

Ca doit permettre :

- De parcourir facilement les différentes versions ;
- De connaître quoi, quand et par qui un document a été modifié, supprimé ;
- D'identifier les différences sur les documents ;
- De fusionner facilement différentes versions

Installation et configuration

Installation

Disponible sur toutes les plateformes :

Sur ubuntu :

`sudo apt-get install git`

<https://doc.ubuntu-fr.org/git>

Sur Windows :

<https://gitforwindows.org/>

Sur Mac :

`brew install git`

<https://git-scm.com/book/en/v1/Getting-Started-Installing-Git>

Configuration

Configurer son nom et son email :



```
1 git config --global user.name "Jeff Martins-Jacquelot"
2 git config --global user.email "jacquelotjeff@gmail.com"
```

Terminologie

On appelle un **dépôt Git** (repository en anglais) un entrepôt virtuel du projet. Il permet d'enregistrer les versions du code.

Configuration

Ajouter une clé SSH :



```
1 # changer jacquelotjeff@gmail.com par votre adresse email  
2  
3 ssh-keygen -t ed25519 -C "jacquelotjeff@gmail.com"
```

Cette commande va générer 2 fichiers : 1 clé privée et 1 clé publique.

- `~/.ssh/id_ed25519` # Ne jamais communiquer cette clé privée
- `~/.ssh/id_ed25512.pub` # Clé à communiquer aux services tiers

Configuration

Lier sa clé SSH à son compte Github :

The screenshot shows the GitHub desktop application interface. At the top, there's a search bar with placeholder text "Type / to search". Below it, a yellow banner displays a message about failed transaction details. On the left, there's a feedback button and a filter icon with the number "8". A modal window titled "The state 2023" is partially visible. In the center, a sidebar lists "Latest changes" with several items from the past few days. On the right, a user profile dropdown for "jacquelotjeff" is open, showing options like "Set status", "Your profile", "Add account", and a long list of links including "Your repositories", "Your projects", "Your Copilot", etc. A red arrow points to the "Settings" link at the bottom of this list.

- jacquelotjeff
- Jeff Jacquelot
- Set status
- Your profile
- Add account
- Your repositories
- Your projects
- Your Copilot
- Your organizations
- Your enterprises
- Your stars
- Your sponsors
- Your gists
- Upgrade
- Try Enterprise
- Feature preview
- Settings**
- GitHub Docs
- GitHub Support

Configuration

Lier sa clé SSH à son compte Github :

The screenshot shows a GitHub user profile for "Jeff Jacquemet (jacquemetjeff)". The profile picture is a cartoon character with glasses. The name "Jeff Jacquemet" and handle "(jacquemetjeff)" are displayed. Below the name is the text "Your personal account" with a "Switch settings context" link. A red arrow points from the text "Lier sa clé SSH à son compte Github :" to the "SSH and GPG keys" section in the sidebar.

Public profile

- Public profile
- Account
- Appearance
- Accessibility
- Notifications

Access

- Billing and plans
- Emails
- Password and authentication
- Sessions
- SSH and GPG keys

Public profile

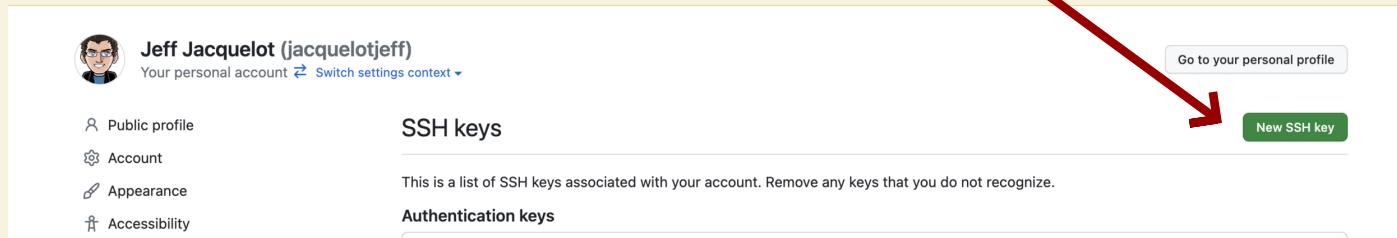
Name
Jeff Jacquemet
Your name may appear here at any time.

Public email
Select a verified email address
You can manage your public email addresses.

Bio
Web Developer
@kubernetes

Configuration

Lier sa clé SSH à son compte Github :



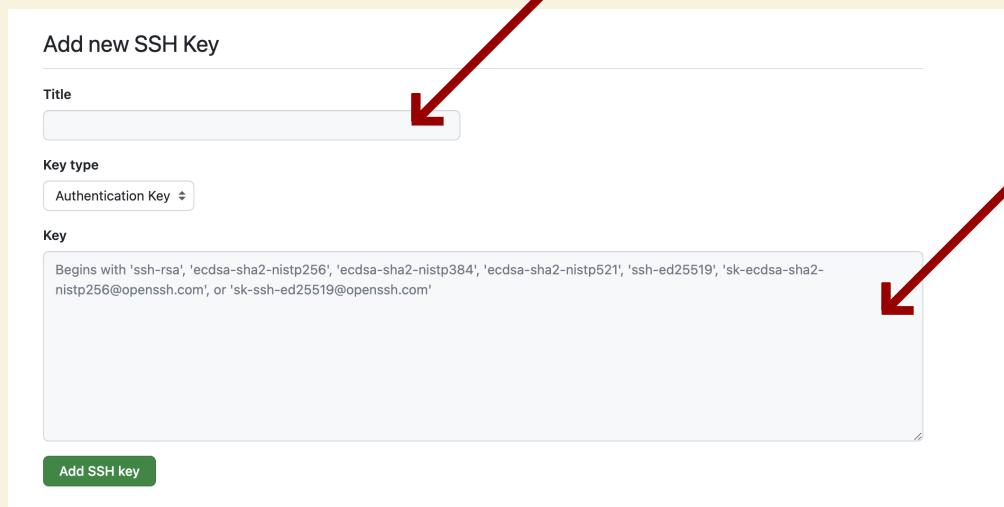
Jeff Jacquelot (jacquelotjeff)
Your personal account Switch settings context

Public profile Account Appearance Accessibility

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

New SSH key



Add new SSH Key

Title

Key type

Key
Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

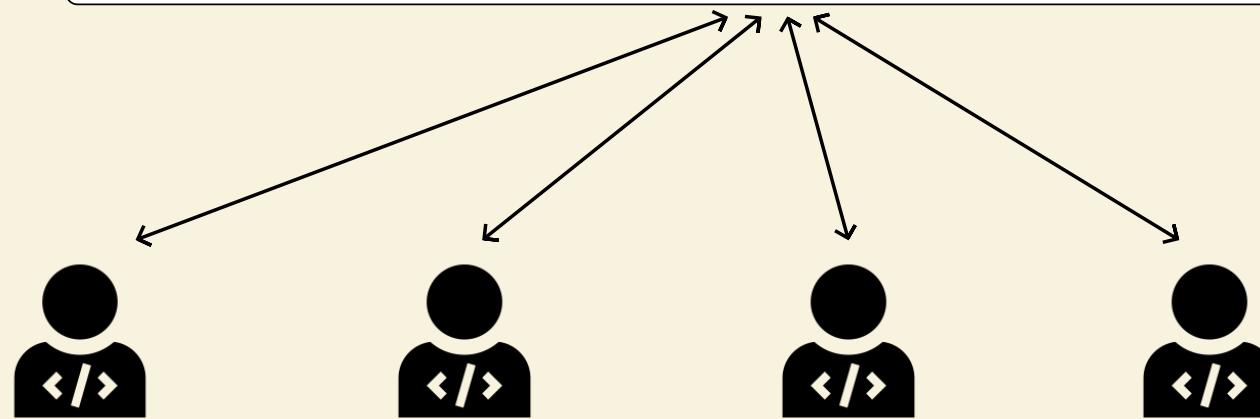
Add SSH key

Nom de l'ordinateur par exemple

Mettre contenu du fichier
~/.ssh/id_ed25512.pub

Fonctionnement "classique" de Git

Le ou les dépôts distants



- Chaque contributeur possède une version du dépôt du projet
- Les contributeurs poussent (push en anglais) les modifications du dépôt et les récupèrent (pull en anglais)

Création d'un nouveau dépôt Git

```
● ● ●  
1 # On ajoute un nouveau dossier :  
2 mkdir monprojet  
3  
4 # On se déplace dans ce dossier :  
5 cd monprojet  
6  
7 # On initialise le nouveau dépôt :  
8 git init
```

Création d'un nouveau dépôt Git

```
● ● ●  
1 # On ajoute un nouveau dossier :  
2 mkdir monprojet  
3  
4 # On se déplace dans ce dossier :  
5 cd monprojet  
6  
7 # On initialise le nouveau dépôt :  
8 git init
```

Ajouter notre premier fichier à Git

```
● ● ●  
1 # Création d'un nouveau fichier  
2 touch monfichier.txt  
3  
4 # Ajoutons le fichier à l'index Git :  
5 git add monfichier.txt  
6  
7 # Note on peut utiliser aussi :  
8 # Permet d'ajouter tous les fichiers dans le dossier actuel à l'index  
9 git add .
```

Enregistrer les modifications apportées au dépôt



```
1 # Le paramètre -m permet d'ajouter un commentaire lié à l'enregistrement des modifications
2 git commit -m "Création d'un nouveau fichier" monfichier.txt
```

Enregistrer les modifications apportées au dépôt



```
1 # Le paramètre -m permet d'ajouter un commentaire lié à l'enregistrement des modifications
2 git commit -m "Création d'un nouveau fichier" monfichier.txt
```

Ajouter une nouvelle modification



```
1 # Editer le fichier
2 echo "Modification 1" > monfichier.txt
3
4 # Enregistrons les nouvelles modifications ...
5 git add monfichier.txt
6 git commit -m "Ajout d'un premier contenu" monfichier.txt
```

Connaître l'état de notre index



```
1 git status
```

```
→ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

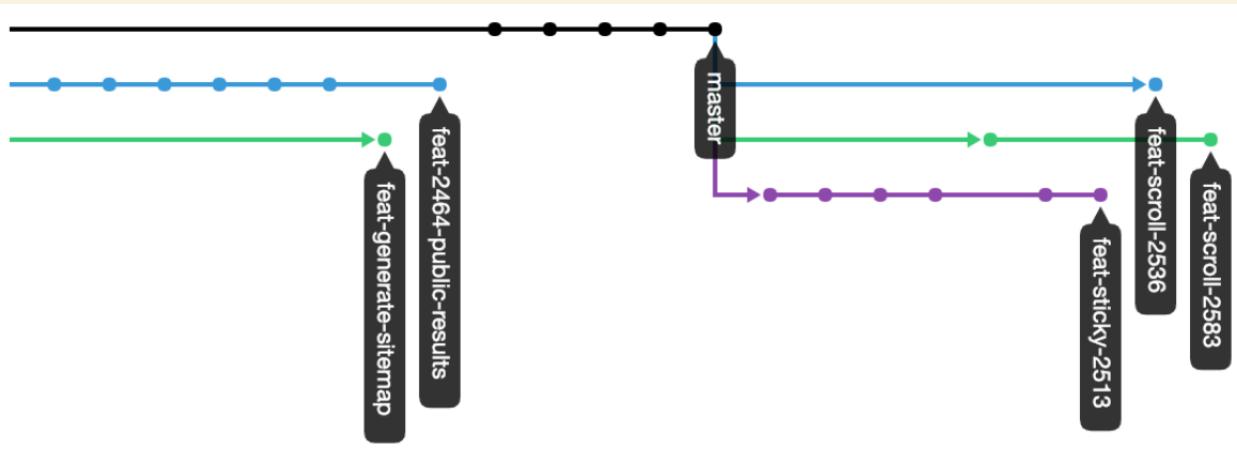
    contact.php
    contacts.php
    db/
    includes/
    index.php
    traitement/

nothing added to commit but untracked files present (use "git add" to track)
```


Travailler avec des branches

```
● ● ●  
1 # Liste des branches actuelles (-a pour voir toutes les branches) :  
2 git branch  
3  
4 # Création d'une nouvelle branche et se connecter dessus :  
5 git checkout -b nom-branche  
6  
7 # Se déplacer sur une branche existante :  
8 git checkout master
```

Voici un exemple de ligne de métros, euh.. de branches pardon !



Travailler avec des branches

Pour réaliser une fonctionnalité, on doit se positionner sur une branche autre que la branche "Master" (parfois nommée "main").

Une fois le travail réalisé et stable, on doit "merge" (fusionner) le travail de la branche sur la branche "Master".

Mettre de côté ses modifications

Utile avant de récupérer des modifications sur le serveur distant

```
● ● ●  
1 # Mettons de côté nos modifications ...  
2 git stash  
3  
4 # Récupération des modifications  
5 git stash pop  
6  
7 # ou ...  
8 git stash apply
```

Suivre une branche distante

```
● ● ●  
1 # Se positionner sur une branche :  
2 git checkout -b NOM_BRANCHE  
3  
4 # Suivre la branche distance  
5 git branch --set-upstream-to origin/NOM_BRANCHE  
6  
7 # Récupération des modifications enregistrées sur le serveur distant :  
8 # (Nous ne sommes plus obligés de préciser le nom du serveur et de la branche  
     derrière la commande.)  
9 git pull  
10  
11 # Envoi des modifications sur le serveur distant  
12 # (Nous ne sommes plus obligés de préciser le nom du serveur et de la branche  
     derrière la commande.)  
13 git push
```

Afficher les x derniers commits



```
1 git log -n 10
```

Afficher les x derniers commits

```
● ● ●
```

```
1 git log -n 10
```

Annuler le dernier commit

```
● ● ●
```

```
1 git reset HEAD~1
```

Afficher les x derniers commits

● ● ●

```
1 git log -n 10
```

Annuler le dernier commit

● ● ●

```
1 git reset HEAD~1
```

Supprimer un fichier de git

● ● ●

```
1 git rm NOM-FICHIER
2
3 # Supprimer un dossier ainsi que les sous fichiers :
4 git rm -r NOM-DOSSIER
```

```
# Création d'une nouvelle branche  
# (et on se déplace dessus)  
git checkout -b ma_branche  
  
# Ajout d'un fichier avec contenu  
touch monfichier3.txt  
echo 'Mon fichier 3' > monfichier3.txt  
  
# Ajout du fichier à l'index et commit  
git add monfichier3.txt  
git commit monfichier3.txt -m "Ajout du fichier3.txt"  
  
# On se déplace sur la branche master/main  
git checkout master  
  
# On affiche les fichiers (on ne voit pas fichier3.txt)  
ls -al  
  
# On récupère les changements de "ma_branche"  
git merge ma_branche  
  
# Affichage des fichiers : on a maintenant fichier3.txt  
ls -al
```

TP :
Travailler avec les branches

Création d'un dépôt distant sur Github

Création d'un dépôt distant sur Github

Puis création d'un dépôt sur Github : <https://github.com/>

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk ().*

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner * MWJeff  /

Great repository names are short and memorable. Need inspiration? How about [turbo-engine](#) ?

Description (optional)

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

Récupération du dépôt sur votre ordinateur



1) Cliquer sur le bouton "Code"

A screenshot of the same GitHub repository page. A modal window titled 'Clone' is open over the repository content. It shows three cloning methods: 'Local' (selected), 'Codespaces', 'HTTPS' (disabled), 'SSH' (selected), and 'GitHub CLI'. The SSH URL 'git@github.com:MWJeff/test-projet.git' is highlighted with a blue border. A note at the bottom says 'Use a password-protected SSH key.' There is also a note above the URL: 'git@github.com:MWJeff/test-projet.git'.

2) Copier le lien du dépôt

```
1 # 3) Récupérer le contenu du dépôt dans un nouveau dossier :  
2 git clone git@github.com:MWJeff/test-projet.git
```


Pousser des modifications sur le serveur



```
1 # origin correspond au nom donné par défaut au serveur distant  
2 # master correspond au nom de la branche par défaut  
3 git push origin master
```

Pousser des modifications sur le serveur



```
1 # origin correspond au nom donné par défaut au serveur distant  
2 # master correspond au nom de la branche par défaut  
3 git push origin master
```

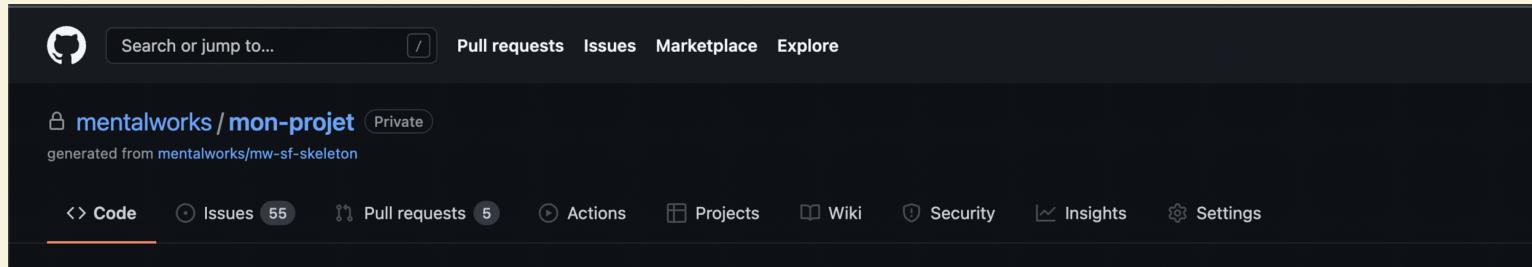
Récupérer des modifications sur le serveur



```
1 # origin correspond au nom donné par défaut au serveur distant  
2 # master correspond au nom de la branche par défaut  
3 git pull origin master
```

Gestion de projet

avec Github et tableaux Kanban



Liste des onglets

- **Code** : affiche l'historique des modifications apportés sur le projet
- **Issues** : affiche les différentes fonctionnalités à réaliser
- **Pull requests** : liste des demandes de fonctionnalités en attente d'approbation
- **Actions** : tâches automatisées
- **Projects** : affichage sous forme de tableau kanban des tâches
- **Wiki** : documentation / annexes du projet

Création d'une nouvelle issue

The screenshot shows the GitHub interface for creating a new issue. At the top, there's a header with the repository name 'mentalworks / mon-projet' (Private), a 'Watch' button (1 watch), and a 'Fork' button (0 forks). Below the header, the navigation bar includes 'Code', 'Issues 65' (the current tab), 'Pull requests 5', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The main area is titled 'Issue: Nouvelle fonctionnalité'. A note says 'Suggérer une nouvelle fonctionnalité sur le projet. If this doesn't look right, choose a different type.' The issue form has a title field with a user icon and a 'Write' tab selected. The 'Write' tab contains a rich text editor with a toolbar and a preview section. Below the editor, there are four sections with bolded headings: '**Contexte**', '**Solution attendue**', '**Développements supplémentaires demandés**', and '**Contexte supplémentaire**'. Each section has a corresponding text input area. At the bottom of the form, there's a note about styling with Markdown, a file attachment section, and a 'Submit new issue' button. To the right of the form, there are sidebar settings for 'Assignees' (None yet), 'Labels' (None yet), 'Projects' (None yet), 'Milestone' (No milestone), 'Development' (Shows branches and pull requests linked to this issue), and 'Helpful resources' (GitHub Community Guidelines). A footer note reminds users to follow GitHub Community Guidelines.

Le langage **Markdown** est utilisé comme texte enrichi.
Guide : <https://www.markdownguide.org/>

Création d'une nouvelle issue

On nomme généralement le titre d'une fonctionnalité en "User story" utilisé notamment dans la **méthodologie Agile**.

“ En tant que <qui>, je veux <quoi> afin de <pourquoi>

Exemple :

**“ En tant qu'utilisateur je veux m'inscrire sur le site
afin de commander un repas**

Découper le projet en ensemble de "tâches"

On liste toutes les tâches sur Github avec les dépendances (temps max par tâche : 4 heures)

Tâche 1 : 4h

Tâche 2 : 2h (dépend de la tâche 1)

Tâche X : ..h

Total : **120** heures

TPS TOTAL / PERSONNE (3 personnes par groupe) : $120/3 = \mathbf{40}$ heures par personne

Durée du projet : **10 semaines**

On prévoit 1 à 2 semaines de "Recette" (tests) + spécifications fonctionnelles

= 8 semaines de disponibles.

TPS DE TRAVAIL / SEMAINE = DUREE TOTALE HEURE / NB SEMAINES

TPS DE TRAVAIL / SEMAINE = $120h / 8$

TPS DE TRAVAIL / SEMAINE = **15 heures de travail par semaine**

TPS DE TRAVAIL PERSONNEL / semaine = $15 / 3$ personnes = **5 heures**

Création d'une nouvelle issue

Associer un collaborateur à l'issue

En tant que collaborateur, je peux accéder à l'historique de mes demandes de congés #2

[Edit](#) [New issue](#)

[Open](#)

MWJeff opened 18 minutes ago

No description provided.

Create sub-issue

MWJeff moved this to Todo in Planning 18 minutes ago

MWJeff added this to Planning 18 minutes ago

Assignees

Assign up to 10 people to this issue

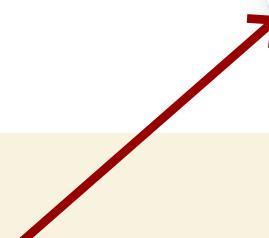
Filter assignees

MWJeff

Projects

Planning

Status Todo



Création d'une nouvelle issue

Associer un tag pour le temps

En tant que collaborateur, je peux accéder à l'historique de mes demandes de congés #2

[Edit](#) [New issue](#) [...](#)

[Open](#)

MWJeff opened 19 minutes ago

No description provided.

[Create sub-issue](#) [...](#)

[MWJeff moved this to Todo in Planning 19 minutes ago](#)

[MWJeff added this to Planning 19 minutes ago](#)

Assignees [Assign yourself](#)

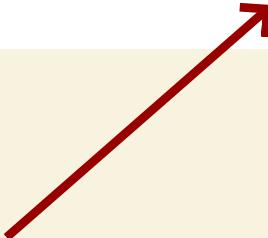
Labels

Apply labels to this issue

Filter labels

Group labels

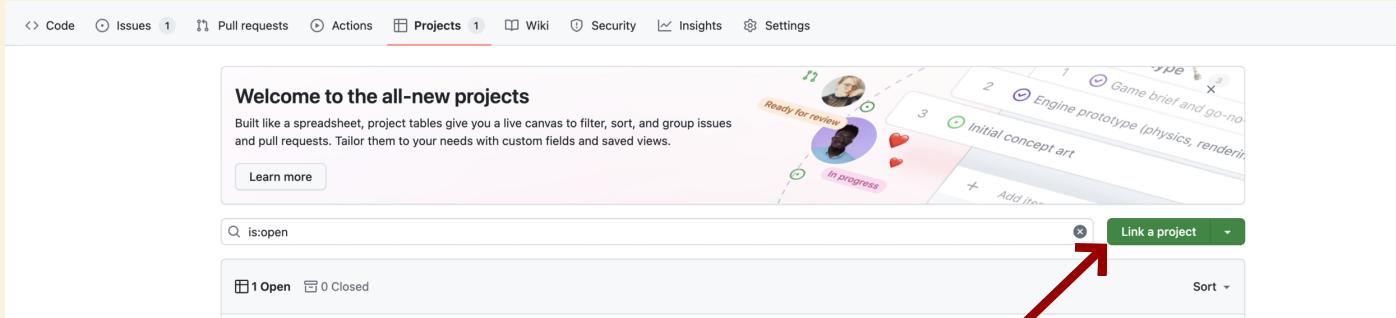
4h



Création d'un projet Kanban



Création d'un projet Kanban



The screenshot shows a project management interface with a header navigation bar including Code, Issues (1), Pull requests, Actions, Projects (1), Wiki, Security, Insights, and Settings. Below the header is a section titled "Welcome to the all-new projects" with a sub-section about project tables. A search bar contains the query "is:open". At the bottom left, there are filters for "1 Open" and "0 Closed". On the right, there is a "Link a project" button with a dropdown arrow. A large red arrow points from the bottom center towards the "Link a project" button.

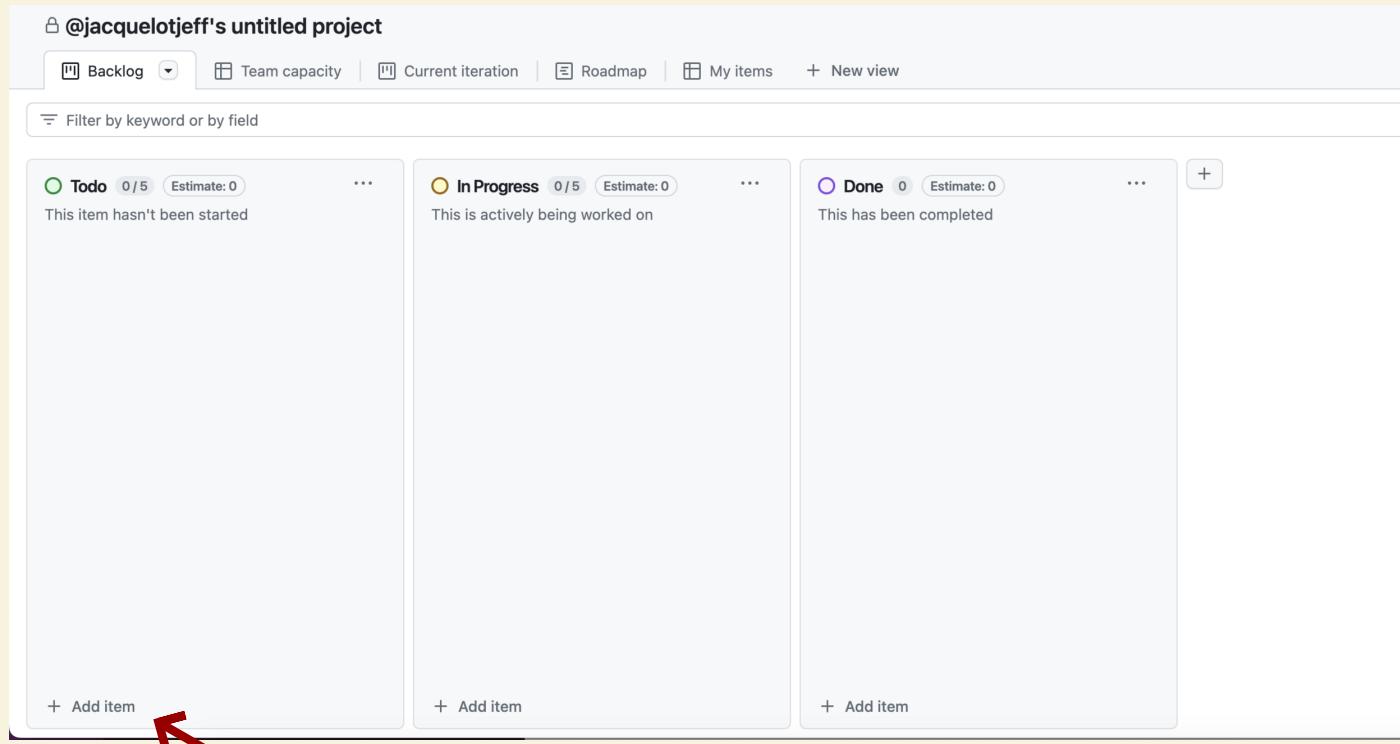
Création d'un projet Kanban

The screenshot shows the Microsoft Project 'Create project' interface. On the left, there's a sidebar with options: 'Create project', 'Project templates', 'Featured' (which is selected), 'Start from scratch', 'Table', 'Board', and 'Roadmap'. The main area displays several project templates:

- Team planning • GitHub**: Manage your team's work items, plan upcoming cycles, and understand team capacity.
- Feature release • GitHub**: Manage your team's prioritized work items when planning for a feature release.
- Kanban**: Visualize the status of your project and track work in progress.

A red arrow points from the 'Team planning' template towards the 'Kanban' template.

Création d'un projet Kanban



The screenshot shows a Kanban board interface with three columns:

- Todo**: 0 / 5 Estimate: 0. Subtext: This item hasn't been started.
- In Progress**: 0 / 5 Estimate: 0. Subtext: This is actively being worked on.
- Done**: 0 Estimate: 0. Subtext: This has been completed.

Each column has a "+ Add item" button at the bottom. A red arrow points to the "+ Add item" button in the "Todo" column.

Ajouter une nouvelle issue dans le tableau

Objectifs liés à votre projet

- Utiliser les issues afin de définir l'ensemble des tâches à accomplir
- Création d'un tableau Kanban et mise à jour tout au long du projet
- Utilisation de Git pour versionner le projet et faciliter le travail en équipe

Exemple de gabarit d'issue

```
● ● ●

1 ## Contexte
2 > Mise en contexte de la demande client et de l'impact sur le projet.
3
4 ## Travail à faire
5 - [ ] Détail de la fonctionnalité à faire avec les différentes règles de gestion.
6
7 ## Sources
8
9 > MCD
10 > Maquette
11 > Autre aide
12
13 ## Besoin d'aide ?
14
15 - XXX > Référent fonctionnel
16 - XXX > Référent technique
```