

Annexes Jira/Bitbucket/GitHub

Sommaire

1. Jira	1
1.1. Création d'un projet	1
1.2. Accès à un projet Kanban	6
1.3. Intégration	10
2. Bitbucket Cloud	12
3. GitHub	19
4. Visual Studio Code	28
5. Les outils graphiques	31
6. Ressources	33

Thierry Vaira - <tvaira@free.fr> - version v0.2 - 23/08/2021 - btssn-lasalle84.github.io

1. Jira

Jira est maintenant un système de gestion de projets développé par **Atlassian**.



1.1. Crédit d'un projet

Se connecter (ou créer un compte) :



Connectez-vous à votre compte

Saisir l'adresse e-mail

Continuer

OU



Continuer avec Google



Continuer avec Microsoft



Continuer avec Apple



Continuer avec Slack

[Vous ne pouvez pas vous connecter ?](#)

[Créer un compte](#)

[Politique de confidentialité](#) • [Avis aux utilisateurs](#)



Un compte pour Jira, Confluence, Trello et [plus](#).

Puis démarrer la création d'un nouveau projet :

The screenshot shows the Jira navigation bar at the top with various links like 'Votre travail', 'Projets', 'Filtres', etc. Below the bar, there's a search bar and some user icons. The main area has a heading 'Projets' and a 'Créer un projet' button.

On choisit le modèle **Kanban** :

Développement logiciel

Planifiez, suivez et livrez des logiciels performants. Soyez opérationnel rapidement grâce aux modèles qui s'adaptent aux méthodes de travail de votre équipe. De plus, profitez des intégrations pour les équipes DevOps qui souhaitent connecter le travail tout au long de leur chaîne d'outils.



Kanban
Affichez et faites avancer votre projet grâce à des tickets sur un tableau performant.



Scrum
Atteignez les objectifs de votre projet grâce à un tableau, un backlog et une feuille de route.



Suivi des bugs
Gérez la liste des tâches de développement et des bugs.

1 Modèle de projet



Kanban
Affichez et faites avancer votre projet grâce à des tickets sur un tableau performant.

Modifier le modèle

Puis le type de projet :

2 Choisir un type de projet

⚠️ Vous devrez créer un projet si vous décidez de changer de type de projet ultérieurement.

Géré par l'équipe

Configuration et gestion par votre équipe.

Pour les équipes qui veulent contrôler leurs propres processus et pratiques de travail dans un espace autonome. Choisissez à la carte des fonctionnalités Agile pour soutenir votre équipe à mesure qu'elle prend de l'ampleur ou gagne en complexité.

Configuration simplifiée



Soyez opérationnel rapidement, grâce à une configuration simplifiée.

Tout membre de votre équipe peut procéder à la configuration et à l'administration

Les paramètres n'impactent pas les autres projets

Configurez facilement les types de tickets et les champs personnalisés

Simplifiez la création de workflows

Définissez des autorisations au niveau de l'accès

Sélectionner un projet géré par l'équipe

Géré par l'entreprise

Configuration et gestion par vos administrateurs Jira

Pour les équipes qui souhaitent travailler avec d'autres équipes sur plusieurs projets de manière standard. Encouragez et favorisez les bonnes pratiques et les processus organisationnels grâce à une configuration partagée.

Configuration pour les experts



Bénéficiez d'un contrôle complet grâce à une configuration, une personnalisation et une flexibilité assurées pour les experts.

Configuration et gestion par vos administrateurs Jira

Configuration standardisée partagée entre les projets

Contrôle total sur les types de tickets et les champs personnalisés

Workflows, états et transitions de tickets personnalisables

Systèmes d'autorisation détaillés

Sélectionner un projet géré par l'entreprise

On précise le nom du projet (et sa clé) :

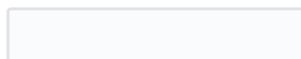
Ajouter les informations du projet

Vous pouvez modifier ces informations à tout moment dans les paramètres de votre projet.

Nom

Essayer un nom d'équipe, un objectif de projet, une étape impo

Clé i



Partager les paramètres avec un projet existant

Connectez des dépôts, des documents et plus encore

Synchronisez le travail de votre équipe à partir d'autres outils avec ce projet pour améliorer la visibilité, l'accès et l'automatisation.

Modèle

Changer de modèle



Kanban

Affichez et faites avancer votre projet grâce à des tickets sur un tableau performant.

Type

Changer de type



Géré par l'entreprise

Travaillez avec d'autres équipes sur plusieurs projets de manière standardisée.

Annuler

Suivant

On connecte le projet à Bitbucket (et éventuellement Confluence) :



Vos outils connectés ?

Plus d'options



Bitbucket Cloud

Créer un dépôt



Confluence Cloud

Créer un espace

Connectez votre projet

Synchronisez le travail à partir de vos outils connectés pour pouvoir accéder à tout ce qui se trouve dans votre projet Jira. Vous pourrez toujours ajouter, supprimer ou modifier ces connexions ultérieurement. [En savoir plus](#)

[+ Ajouter des outils](#)

Ignorer

Connecter (2)



Il est possible d'intégrer GitHub (ou GitLab) à la place de Bitbucket.

Votre travail est maintenant connecté



miniprojet1-team1

Dépôt dans Bitbucket

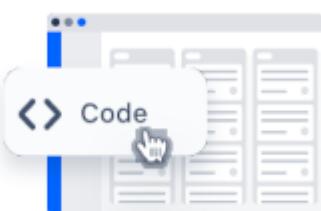
Afficher



miniprojet1-team1

Espace dans Confluence

Afficher



Vous pouvez accéder à votre travail connecté depuis les onglets **Pages**, **Code** ou **Astreinte** de votre projet.

[En savoir plus](#)

[Accéder au projet](#)

1.2. Accès à un projet Kanban

La barre latérale :



miniprojet1-team1

Projet logiciel



Tableau MP1T1



Tableau



Feuille de route



Tableau Kanban



Rapports



Tickets



Composants



Code



Déploiements



Versions



Pages de projet



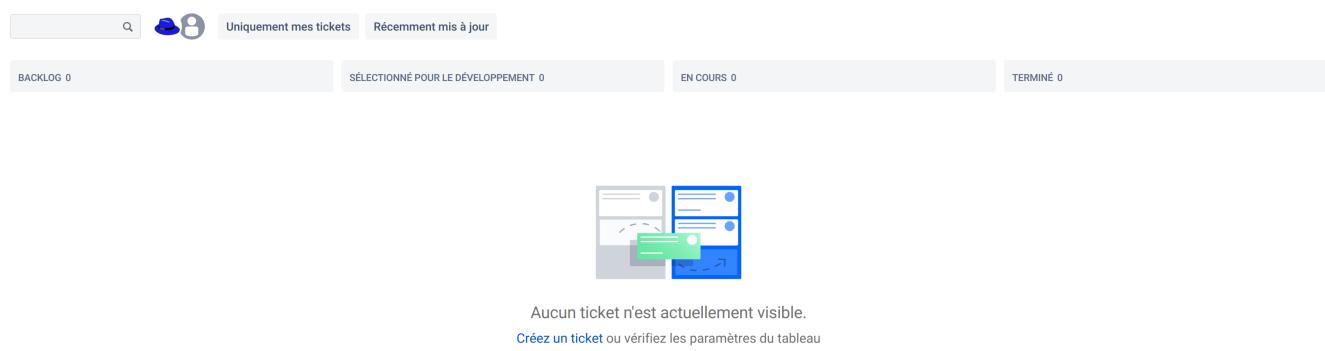
Ajouter un raccourci



Paramètres du projet

La vue du tableau :

Tableau Kanban



Les informations sur le projet :

Informations

Les différents types de tickets :

Types de tickets

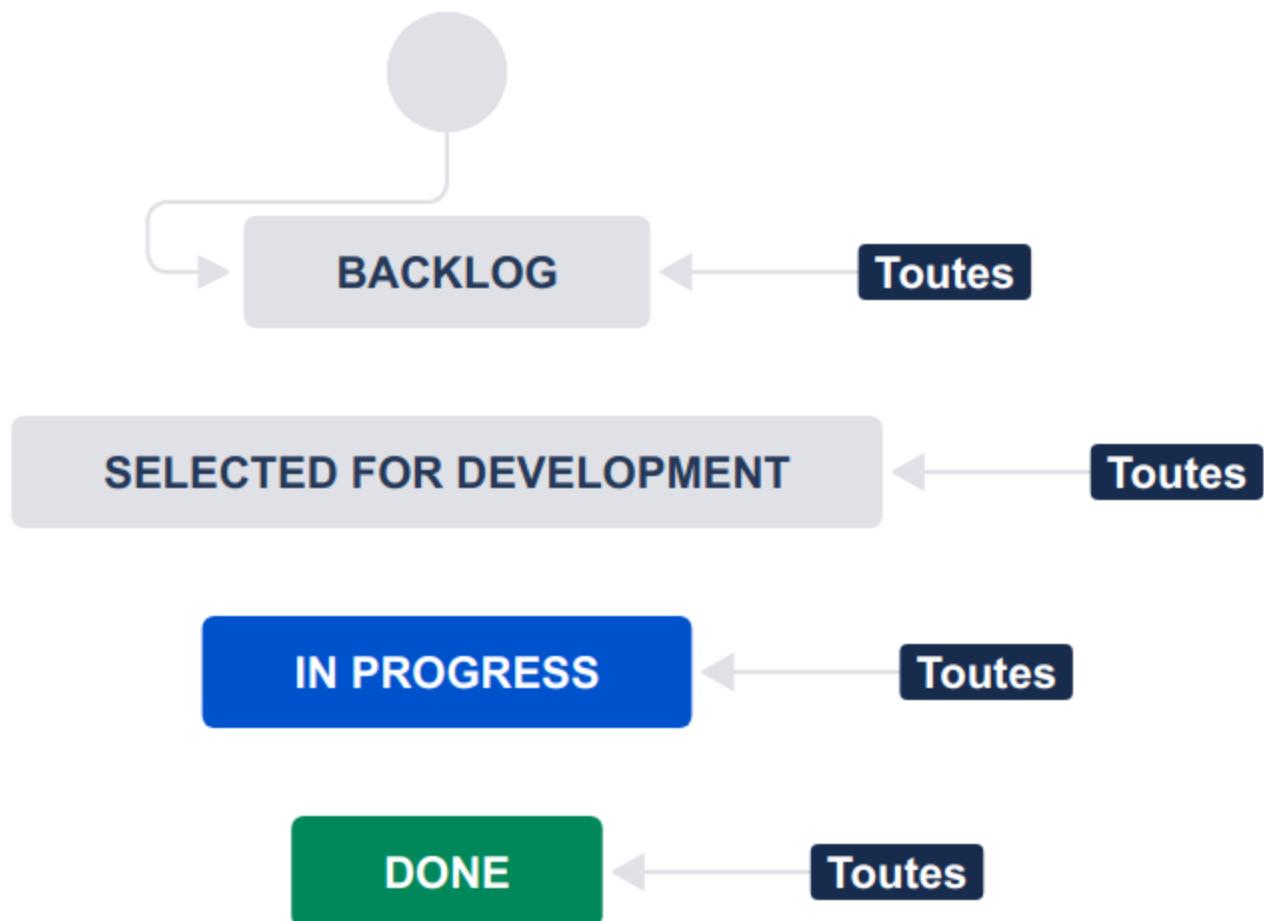
MP1T1: Kanban Issue Type Scheme

Gardez une trace des différents types de tickets, comme les bugs et les tâches.

Le système de types de ticket détermine les types de tickets utilisés par ce projet.

Type de ticket	Description
 Story	Une fonctionnalité exprimée sous la forme d'un objectif utilisateur.
 Bug	Un problème ou une erreur.
 Epic	Une collection de bugs, stories et tâches connexes.
 Tâche	Une tâche distincte.
 Sous-tâche SOUS-TÂCHE	Une petite unité de travail qui fait partie d'une tâche plus importante.

Le *workflow* par défaut :



Il est possible d'automatiser le flux :

The screenshot shows the Jira Automation library interface. On the left, there's a sidebar with project navigation and settings. The main area is divided into sections: Logiciels and DevOps. Each section contains several automation rules represented by icons and descriptions.

Section	Description
Logiciels	Lorsque toutes les stories sont terminées → fermer l'épic
	Lorsqu'une epic est terminée → fermer toutes les stories présentes
	Lorsqu'un ticket est transitionné → l'assigner automatiquement
	Fermer les tickets en double
DevOps	Lorsqu'un parent est terminé → déplacer toutes les sous-tâches vers « Terminé »
	Lorsqu'un ticket est résolu → commenter tous les tickets associés
	Lier les tickets mentionnés dans les commentaires
	Quand un commit est effectué → puis déplacer le ticket vers « En cours »
Quand une branche est créée → puis déplacer le ticket vers « En cours »	
Quand une pull request est mergée → puis déplacer le ticket vers « Terminé »	

Automation

Règles Journal d'audit Bibliothèque

Filtrer les règles

Nom ↗

Quand un commit est effectué → puis déplacer le ticket vers « En cours »

Quand une branche est créée → puis déplacer le ticket vers « En cours »

Quand une pull request est mergée → puis déplacer le ticket vers « Terminé »



Toutes les règles



Règles globales



Règles de projet

1.3. Intégration

Jira est capable de se connecter à des outils de développement compatibles pour offrir des fonctionnalités intéressantes.

On utilisera soit :

- Bitbucket Cloud

Code

Dépôts
Il s'agit de dépôts qui contiennent des commits, des branches ou des pull requests associés à ce projet.


Bitbucket connecté
 Bitbucket est sur votre site. Vous pouvez utiliser des clés de ticket pour lier vos dépôts à Jira. En


Bitbucket
miniprojet1-team1
 Dernière mise à jour le il y a 2 minutes

Utilisez les clés de ticket dans vos noms de branches, messages de commit et titres de pull request pour commencer à voir les dépôts liés ici.



Utilisez des clés de ticket pour associer les outils de développement aux tickets.

Incluez une clé de ticket dans les noms de branches, les messages de commit ou les pull requests que vous créez.

Découvrez les informations de développement dans les tickets Jira et passez rapidement à l'action.

• GitHub

 **MP1-Team0**
Projet logiciel

-  **Tableau MP1T0**
Tableau
-  Feuille de route
-  Tableau Kanban
-  Rapports
-  Tickets
-  Composants
-  **Code**

Code

Dépôts
Il s'agit de dépôts qui contiennent des commits, des branches ou des pull requests associés à ce projet.


GitHub
btssn-lasalle84/mp1-team0
 Dernière mise à jour le il y a 22 minutes

Utilisez les clés de ticket dans vos noms de branches, messages de commit et titres de pull request pour commencer à voir les dépôts liés ici.

Smart Commits

Il est possible d'utiliser des messages de « commit intelligent » (*Smart Commits*) pour faire le lien avec Jira à l'aide de commandes spéciales dans vos messages de validation :

- commenter (#comment <commentaires>)
- enregistrer les informations de suivi du temps (#time 1w 2d 4h 30m)
- faire passer les tickets à n'importe quel statut (par exemple « Résolu ») défini dans le flux de travail du projet Jira (#close)

Au minimum, il faut inclure la clé du ticket au début du message de *commit* pour faire lien avec Jira (Exemple : `git commit -m "JRA-123 <commit message>"`).

Liens :

- <https://support.atlassian.com/jira-cloud-administration/docs/enable-smart-commits/>
- <https://support.atlassian.com/jira-software-cloud/docs/process-issues-with-smart-commits/>

2. Bitbucket Cloud

Bitbucket est un service web d'hébergement et de gestion de développement logiciel utilisant le logiciel de gestion de versions Git. Bitbucket est gratuit pour les particuliers et les petites équipes comptant jusqu'à 5 utilisateurs, avec des référentiels publics et privés illimités.



L'accès initial à Bitbucket :



Placez quelques bits dans votre paquet

HTTPS ▾

git clone https://btssn-avignon-admin@bitbucket.org/btssn-avignon/minip



Lancez-vous

Creating a README or a .gitignore is a quick and easy way to get something into your repository.

[Créer un fichier LISEZMOI](#)

[Create a .gitignore](#)

Obtenez votre dépôt Git local sur Bitbucket

Étape 1 : Basculer vers le répertoire de votre dépôt

```
1 cd /path/to/your/repo
```

Étape 2 : Connecter votre dépôt existant à Bitbucket

```
1 git remote add origin https://btssn-avignon-admin@bitbucket.org/btssn-avignon/miniprojet1-team1.g  
2 git push -u origin master
```

Need more information? [Learn more](#)

On clone le dépôt :

```
$ git clone https://btssn-avignon-admin@bitbucket.org/btssn-avignon/miniprojet1-team1.git
Clonage dans 'miniprojet1-team1'...
Password for 'https://btssn-avignon-admin@bitbucket.org':
warning: Vous semblez avoir cloné un dépôt vide.

$ cd miniprojet1-team1/

$ git status
Sur la branche master

Aucun commit

rien à valider (créez/copiez des fichiers et utilisez "git add" pour les suivre)

$ git remote -v
origin  https://btssn-avignon-admin@bitbucket.org/btssn-avignon/miniprojet1-team1.git
(fetch)
origin  https://btssn-avignon-admin@bitbucket.org/btssn-avignon/miniprojet1-team1.git
(push)

$ git push -u origin master
```

Si le dépôt local existe déjà, on peut le relier au dépôt distant :

```
$ git remote add origin https://btssn-avignon-admin@bitbucket.org/btssn-avignon/miniprojet1-team1.git
```



Si besoin, on renomme la branche `master` en `main` (l'option `-M` est un raccourci pour les options `--move` et `--force`) :

```
$ git branch -M main
```

On peut configurer Git localement :

```
$ git config --local user.name "vaira"
$ git config --local user.email "vaira@lasalle84.org"

$ cat .git/config
...
```

On peut installer les extensions `git-flow` :

```
$ sudo apt install git-flow
```

```
$ git flow help  
usage: git flow <subcommand>
```

Available subcommands are:

init	Initialize a new git repo with support for the branching model.
feature	Manage your feature branches.
bugfix	Manage your bugfix branches.
release	Manage your release branches.
hotfix	Manage your hotfix branches.
support	Manage your support branches.
version	Shows version information.
config	Manage your git-flow configuration.
log	Show log deviating from base branch.

Try 'git flow <subcommand> help' for details.

On initialise le dépôt avec la prise en charge du modèle de branchement **Gitflow** :

```
$ git flow init  
No branches exist yet. Base branches must be created now.  
Branch name for production releases: [master]  
Branch name for "next release" development: [develop]  
  
How to name your supporting branch prefixes?  
Feature branches? [feature/]  
Bugfix branches? [bugfix/]  
Release branches? [release/]  
Hotfix branches? [hotfix/]  
Support branches? [support/]  
Version tag prefix? []  
Hooks and filters directory? [$HOME/jira/miniprojet1-team1/.git/hooks]  
  
$ git push -u origin develop  
Décompte des objets: 2, fait.  
Écriture des objets: 100% (2/2), 160 bytes | 160.00 KiB/s, fait.  
Total 2 (delta 0), reused 0 (delta 0)  
To https://bitbucket.org/btssn-avignon/miniprojet1-team1.git  
 * [new branch]      develop -> develop  
  
$ git ls-remote  
From https://btssn-avignon-admin@bitbucket.org/btssn-avignon/miniprojet1-team1.git  
ddc49adcf578dea85148ff5aa146033b6162e8a6      HEAD  
ddc49adcf578dea85148ff5aa146033b6162e8a6      refs/heads/develop  
ddc49adcf578dea85148ff5aa146033b6162e8a6      refs/heads/master
```

Dans les détails du projet, on définit la branche principale (ici **master**) :



miniprojet1-team1



Back

GÉNÉRAL

Détails du dépôt

Accès utilisateurs et gr...

Clés d'accès

Alias de nom d'utilisat...

▼ ADVANCED

Forking

Allow forks



Language

C++



Main branch

master



Default landing page

Source



Website

Google Analytics key

Dans les modèles de branches, on définit la branche de développement (ici **develop**) :

Modèle de branches

Define how you want your branches to be named, and specify your development and production branches, to help us suggest source and target branches. [En savoir plus](#)

Development branch [i](#)

Use specific branch [▼](#) develop [▼](#)

Production branch [i](#)

No production branch [▼](#)

Branch prefixes

Define the default prefixes for new branches, to allow automated workflows and make branch types clearer.

- Bugfix bugfix/ [i](#)
 - Feature feature/ [i](#)
 - Hotfix hotfix/ [i](#)
 - Release release/ [i](#)
- [Enregistrer](#) [Annuler](#)

On obtient :

btssn-avignon / miniprojet1-team1 / miniprojet1-team1

Branches

Search branches [Q](#) All branches [▼](#) Branch type [▼](#)

Branch [▼](#)

master MAIN
develop DEVELOPMENT

3. GitHub



Liens :

- <https://support.atlassian.com/jira-cloud-administration/docs/integrate-with-github/>
- <https://github.com/atlassian/github-for-jira>

The screenshot shows a sidebar on the left with navigation links: 'Code' (selected), 'Versions', 'Pages de projet', 'Ajouter un raccourci', and 'Paramètres du projet'. Below this, a message says 'Vous faites partie d'un projet géré par l'entreprise' and a link 'En savoir plus'. The main area is titled 'ATLASSIAN Marketplace' and displays three integration cards:

- GitHub for Jira**: Your code, where you need it. Deployments, Integrations, Sour... 74.2k installs
- Git Integration for Jira**: Integration for Git Commits, Branches, Tags & Pull Requests in Jira. Connect git Custom fields, Integrations, Rep... 9.9k installs
- GitLab.com for Jira Cloud**: View GitLab.com activity from Jira Cloud. Repository connectors, Reposito... 16.7k installs

At the bottom right, there is a link 'Parcourez l'Atlassian Marketplace pour plus d'intégrations de code'.



Add to Jira



GitHub for Jira

by Atlassian

★ ★ ★ ★ 263



CLOUD SECURITY PARTICIPANT

Get app

FREE

FREE

Overview

GitHub for Jira will perform the following actions:

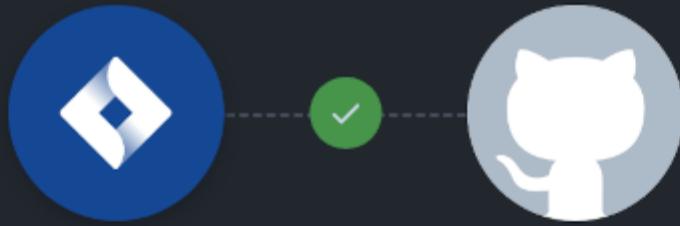
- Delete data from the host application
- Write data to the host application
- Read data from the host application

[View app details](#)

[Get it now](#)

Cancel

[Commit changes](#) Cancel



Jira by **Atlassian** would like permission to:



Verify your GitHub identity (tvaira)



Know which resources you can access



Act on your behalf

Jira has not been installed on any accounts you have access to.

[Learn more about Jira](#)

[Cancel](#)

[Authorize Jira](#)

Authorizing will redirect to

<https://github.atlassian.com>

Not owned or
operated by GitHub

Created 4 years ago

More than 1K
GitHub users



Connect an Organization to Jira

Which organization do you want to connect to?

Don't see your account? [Install Jira](#) on your user account or an Organization you own.

Please note: We currently sync data from all repositories you've selected in the installation's settings. If there a large number of repositories in the installation, initial data sync will take some time to complete. To review or update the repositories you've selected for an installation, [click here](#).

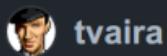


<https://btssn-lasalle84.atlassian.net>



Install Jira

Where do you want to install Jira?



tvaira



btssn-lasalle84



Campus-Advisors





Install Jira

Install on your organization btssn-lasalle84



All repositories

This applies to all current *and* future repositories.

Only select repositories

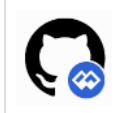
with these permissions:

- Read** access to code, members, and metadata
- Read and write** access to actions, deployments, issues, and pull requests

[Install](#)

[Cancel](#)

Next: you'll be directed to the GitHub App's site to complete setup.



GitHub integration for Whiteboards

Work with GitHub and Jira issues together on the same virtual board. See the full picture and synchronize your teams' efforts

Continuous integration, Integrations, Source code
1.1k installs



GitHub integration for Jira

GitHub integration for Jira. Create branches from the issue view. Manage Pull Requests & Tags & Code & More

Integrations, Repository connectors, Source code, ...
★★★★★ 5
373 installs



GitHub for Jira

Your code, where you need it

Deployments, Integrations, Source code, Workflow
★★★★★ 263
ADDED

CLOUD SECURITY PARTICIPANT

 Jira / Marketplace apps / GitHub integration for Jira

Feedback



GitHub integration for Jira

by Paweł Sieniarski

★★★★★ 5

Get app FREE

[Overview](#) [Support](#)

GitHub integration for Jira. Create branches from the issue view. Manage Pull Requests & Tags & Code & More

On obtient :

Apps

ATLASSIAN MARKETPLACE

Trouver de nouvelles apps

Gérer les apps

Demandes d'apps

Promotions

Informations d'identification O...

Apps

Gérer les apps

Vous pouvez installer, mettre à jour, activer et désactiver des apps ici. [Trouvez de nouvelles apps.](#)

Filtrer les apps visibles | User-installed

+ Développer une nouvelle ap

User-installed apps

- >  GitHub
- >  GitHub integration for Jira
- >  JIRA Toolkit Plugin

Configuration :

GitHub configuration

Organization	Added	Repos Synced	Sync Status	Last Sync Update	Retry	Manage Organizations
 btssn-lasalle84	2 minutes ago	2 (All)	COMPLETE	7 minutes ago	<input type="button" value="Normal"/>	<input type="button" value="Submit"/>

Metadata for commits, branches, and pull requests that use the [Smart Commit syntax](#) will be synced to Jira and appear in the [Development Information panel](#) of the relevant issue.

To view your Organization's updated sync status, [please refresh](#) this page.

GitHub integration for Jira



Projects Branches

Visibility	Project	Key	Repositories
<input checked="" type="checkbox"/>	Projet de services de démo	DESK	Select related repositories <input type="button" value="▼"/>
<input checked="" type="checkbox"/>	MP1-Team0	MP1T0	mp1-team0 <input type="button" value="x"/> <input type="button" value="▼"/>

GitHub integration for Jira



Projects Branches

Templates

Branch Name Template

`${prefix}${issuekey}-${summary}`

For example: feature/KAN-12-your-issue-summary

Useful variables with examples

`${prefix} - bugfix
 ${issuekey} - KAN-12
 ${summary} - your-issue-summary
 ${projectkey} - KAN
 ${displayname} - john-doe`

Prefixes

Issue Type	Template
Sous-tâche	<input type="text"/>
Story	<input type="text"/> feature/
Tâche	<input type="text"/>
Bug	<input type="text"/> bugfix/
IT Help	<input type="text"/>
Service Request	<input type="text"/>
Service Request with Approvals	<input type="text"/>
Epic	<input type="text"/>

On clone le dépôt :

```
$ git clone git@github.com:btssn-lasalle84/mp1-team0.git  
$ cd mp1-team0/  
  
$ echo "# mp1-team0" >> README.md  
$ git add README.md  
$ git commit -m "Ajout README.md"  
  
$ git branch -M main  
  
$ git push -u origin main
```

On installe des extensions [git-flow](#) :

```
$ sudo apt install git-flow
```

On initialise le dépôt Git avec la prise en charge du modèle de branchemet Gitflow :

```
$ git flow init  
  
Which branch should be used for bringing forth production releases?  
- main  
Branch name for production releases: [main]  
Branch name for "next release" development: [develop]  
  
How to name your supporting branch prefixes?  
Feature branches? [feature/]  
Bugfix branches? [bugfix/]  
Release branches? [release/]  
Hotfix branches? [hotfix/]  
Support branches? [support/]  
Version tag prefix? []  
Hooks and filters directory? [/home/tv/Documents/jira/mp1-team0/.git/hooks]  
  
$ git push -u origin develop  
Total 0 (delta 0), reused 0 (delta 0)  
remote:  
remote: Create a pull request for 'develop' on GitHub by visiting:  
remote:     https://github.com/btssn-lasalle84/mp1-team0/pull/new/develop  
remote:  
To github.com:btssn-lasalle84/mp1-team0.git  
 * [new branch]      develop -> develop  
La branche 'develop' est paramétrée pour suivre la branche distante 'develop' depuis  
'origin'.
```

4. Visual Studio Code

Visual Studio Code (un des IDE les plus utilisés actuellement) intègre la gestion du contrôle de source (SCM) et inclut par défaut la prise en charge de Git.

Installation de Visual Studio Code :

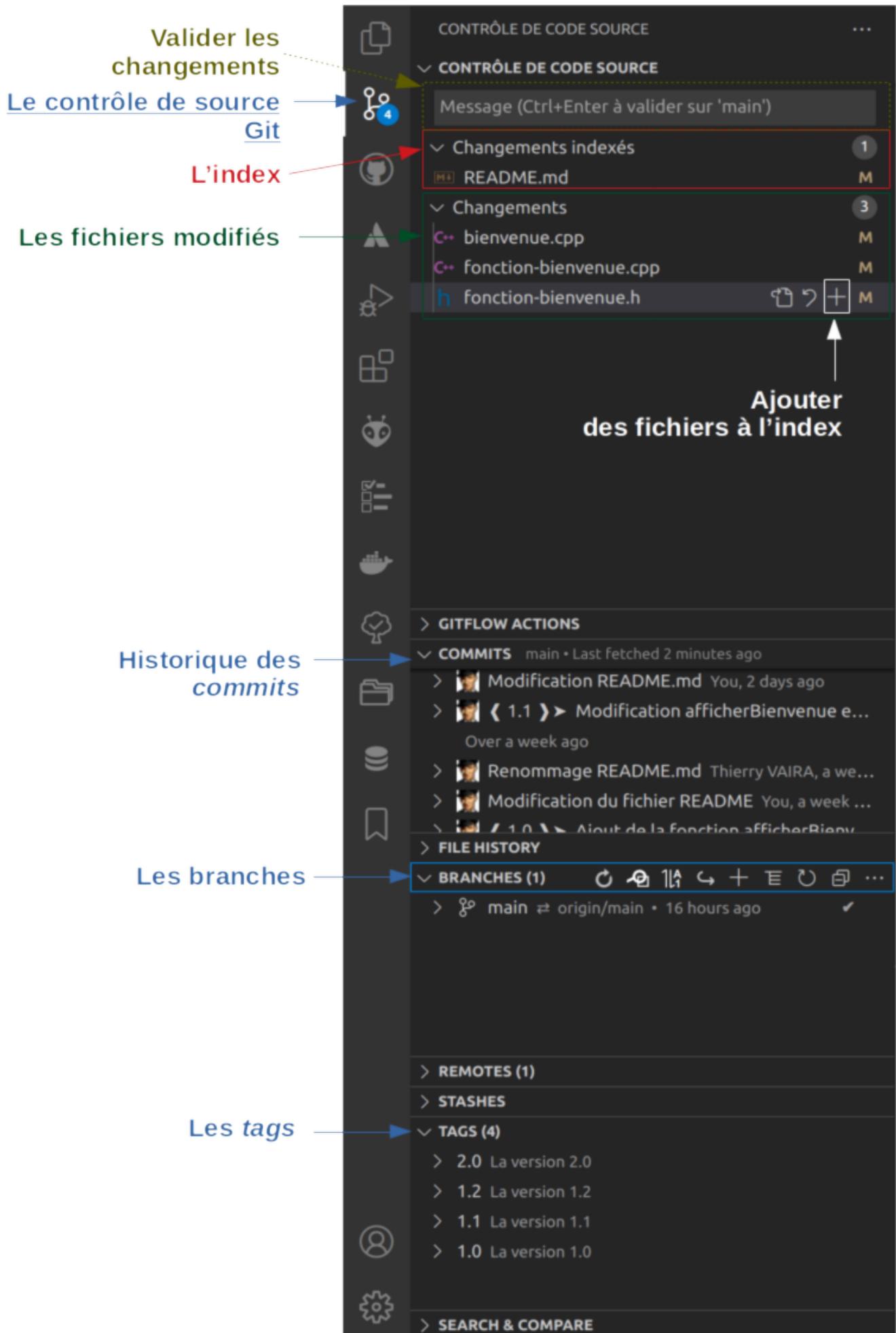
- [Download](#)
- [Setup](#)
- [Getting Started](#)

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a file tree with a folder named 'TP-GIT-SEQUENCE-1' containing files like '.gitignore', 'bienvenue.cpp', 'bienvenue.h', 'fonction-bienvenue.cpp', 'fonction-bienvenue.h', 'hello.cpp', and 'Makefile'. The 'STRUCTURE' section shows a '# Bienvenue' folder with sub-items like '# La fonction afficherBienvenue' and '# Exemple d'utilisation'. The 'CHRONIQUE' section shows a log of recent changes, including a merge from 'modification-fonction' into 'main' and a modification of 'fonction-bienvenue.h'. The main editor area shows a C++ code snippet for a 'bienvenue' program. Below the editor is a terminal window displaying command-line history, including a 'make rebuild' command and several './bienvenue' executions. The bottom status bar shows various tool icons and the current terminal session information.

Liens :

- <https://code.visualstudio.com/docs/editor/versioncontrol>
- <https://code.visualstudio.com/docs/editor/github>

Le contrôle de source (SCM) :



Il existe de nombreuses extensions pour faciliter l'utilisation de Git dont [Git Extension Pack](#) qui comprend :

- [Git History](#)
- [Project Manager](#)
- [GitLens](#)
- [gitignore](#)
- [Open in GitHub / Bitbucket / VisualStudio.com](#)

Et quelques autres :

- [Git Graph](#)
- [Git Project Manager](#)
- [Git Blame](#)

Et pour gitflow :

- [Gitflow Actions Sidebar](#)
- [gitflow](#)

The screenshot shows the Visual Studio Marketplace interface. At the top, there's a search bar with the text "gitflow". Below it, a list of extensions is displayed:

- gitflow** 1.2.1 by vector-of-bool. It has 75K downloads and a 4.5 rating. The description reads: "Gitflow integration and support in Visual ...". There's a gear icon for settings.
- gitflow** 2.1.0 by Bui Anh Thang. It has 1K downloads and a 4.5 rating. The description reads: "Gitflow integration and support in Visual ...". There's an "Installer" button.
- Gitflow Actions Sidebar** 0.5.2 by Nugroho Ardi Sutrisno. It has 4K downloads and a 4 rating. The description reads: "Gitflow integration sidebar in Visual Studi...". There's a gear icon for settings.

Et pour Jira/Bitbucket :

- Jira and Bitbucket (Atlassian Labs)

The screenshot shows the Atlassian Settings interface for the "Jira and Bitbucket (Atlassian Labs)" extension. The top navigation bar includes tabs for "JIRA" and "BITBUCKET". On the left, there's a sidebar with sections for "Authentication", "Jira Issues Explorer", "Jira Issue Hovers", "Create Jira Issue Triggers", "Start Work", and "Status Bar". The main content area displays configuration options for these features. On the right, there are buttons for "SEND FEEDBACK", "Source Code", "Got Issues?", and "User Guide".

5. Les outils graphiques

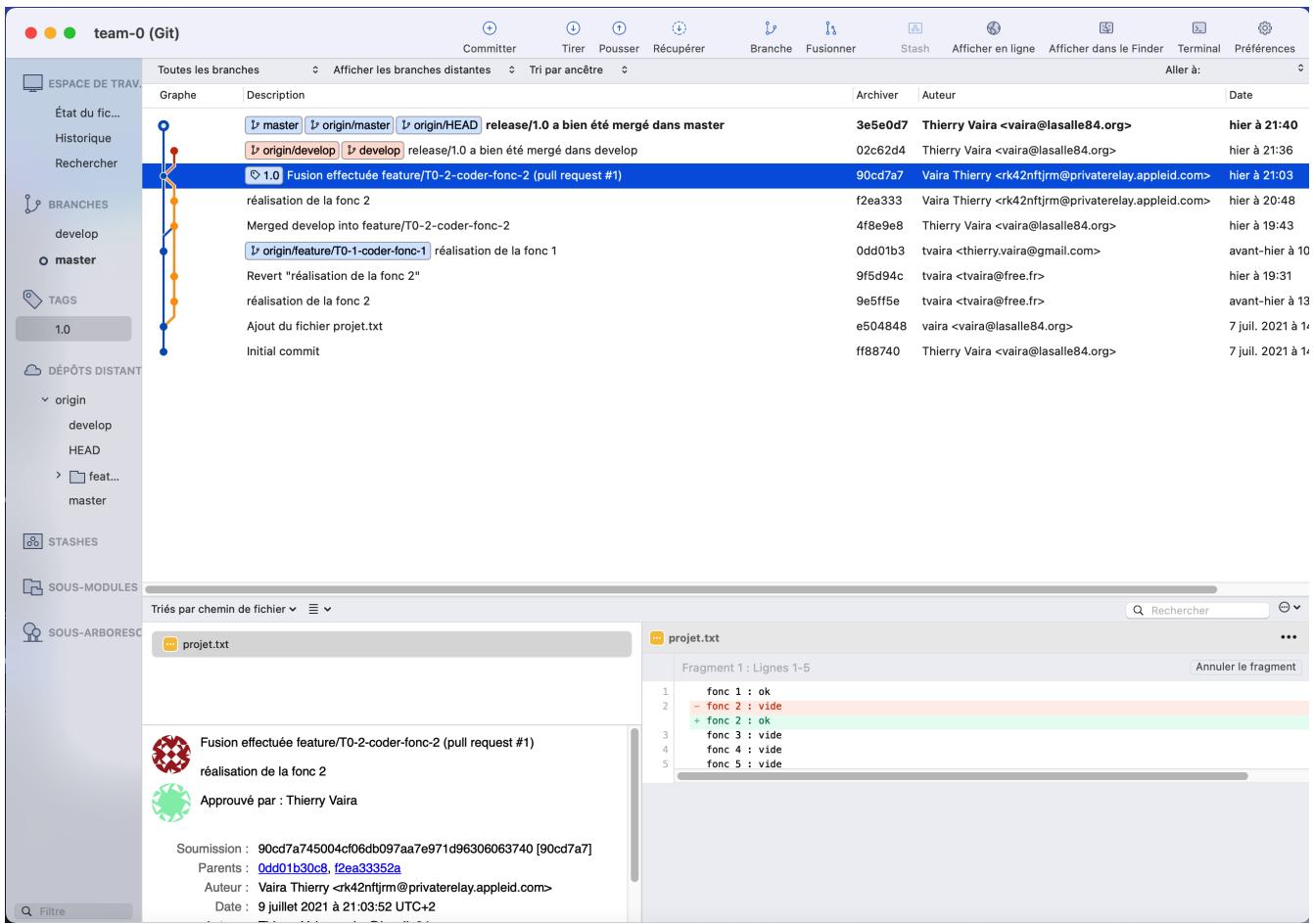
Il existe de nombreuses interfaces graphiques permettant de gérer des projets Git.

En standard :

- une interface web avec [GitWeb](#)
- une interface de visualisation détaillée et graphique avec [gitk](#)

Il existe également de nombreuses autres applications :

- [qgit](#) propose des outils supplémentaires par rapport à [gitk](#) ;
- [Giggle](#) : une interface en GTK+ ;
- [GitExtensions](#) : un client Git graphique pour Windows © ;
- [TortoiseGit](#) : logiciel libre pour Windows reprenant les éléments d'interface de [TortoiseSVN](#) (un classique) ;
- ...
- [SourceTree](#) : un logiciel propriétaire gratuit pour Windows © et macOS © édité par Atlassian ;



- **GitEye** : un client graphique pour Windows ©, macOS © et Linux

```
$ cd ~/Téléchargements/
$ wget -c https://www.collab.net/sites/default/files/downloads/GitEye-2.2.0-
linux.x86_64.zip

$ mkdir /tmp/GitEye
$ unzip -d /tmp/GitEye ~/Téléchargements/GitEye-2.2.0-linux.x86_64.zip

$ sudo chown -R root:root /tmp/GitEye
$ sudo mv /tmp/GitEye /opt/GitEye
$ sudo ln -s /opt/GitEye/GitEye /usr/local/bin/GitEye
$ GitEye
```

The screenshot shows the CollabNet GitEye interface. On the left, there's a sidebar with navigation links like 'File', 'Edit', 'Search', 'Git', 'Window', 'Help', 'Git Repositories', 'CollabNet Sites', 'Branches', 'Tags', 'References', 'Remotes', 'Working Tree', and 'git'. The main area has tabs for 'Dashboard', 'Files', 'History', 'Task List', and 'Builds'. Under 'History', a table lists commits with columns for 'Id', 'Message', 'Author', 'Authored Date', 'Committer', and 'Committed Date'. The commits are:

ID	Message	Author	Authored Date	Committer	Committed Date
90cd7a7	1.0 develop master release/1.0 origin/HEAD origin/develop origin/master origin/release/1.0 HEAD	Vaira Thierry	12 hours ago	Thierry Vaira	12 hours ago
f2e0333	réalisation de la fnc 2	Vaira Thierry	12 hours ago	Vaira Thierry	12 hours ago
4f69e88	Merged develop into feature/TO-2-coder-func-2	Thierry Vaira	13 hours ago	Thierry Vaira	13 hours ago
9f5d94c	Revert "réalisation de la fnc 2"	tvaira	13 hours ago	tvaira	13 hours ago
9ef5fe5	réalisation de la fnc 2	tvaira	2 days ago	tvaira	2 days ago
0dd01b3	origin/feature/... réalisation de la fnc 1	tvaira	2 days ago	tvaira	2 days ago
e504848	origin/feature/... Ajout du fichier projet.txt	vaira	3 days ago	vaira	3 days ago
ff88740	Initial commit	Thierry Vaira	3 days ago	Thierry Vaira	3 days ago

Below the commits, a detailed commit message for '90cd7a7' is shown:

```

commit 90cd7a7a5094cfe00b07aa7e371d9e36e095740
Author: Thierry Vaira <tvaira@lasalle84.org> 2021-07-09 21:03:52
Committer: Thierry Vaira <tvaira@lasalle84.org> 2021-07-09 21:03:52
Parent: 8dd91b30c3b0e8e19150103c496edcd7fbfa232 (réalisation de la fnc 1)
Parent: f2e03332ac3786cb05f982116a01bd432ef09 (réalisation de la fnc 2)
Branches: develop, master, release/1.0, origin/develop, origin/master, origin/release/1.0
Tags: 1.0

Fusion effectuée feature/TO-2-coder-func-2 (pull request #1)

réalisation de la fnc 2

Approuvé par : Thierry Vaira
  
```

6. Ressources

Git :

- [Git](#)
- [Manuel de référence](#)
- [Livre Pro Git en français](#)
- [Livre Git Community Book en français](#)

GitHub :

- [Git Handbook sur Github](#)
- [Collaborating with pull requests](#)

Atlassian :

- [Les guides Jira](#)
- [À quoi sert Jira ?](#)
- [Découvrez Agile grâce à Jira](#)
- [Tutoriels](#)
- [Comparaison des workflow git](#)
- [Smart Commits](#)
- [Integrate with GitHub](#)
- [GitHub for Jira](#)