**Glossaire GitHub – Terminologie & Constituants (Windows 11)**

*Version imprimable • Généré le 24/08/2025 à 00:58*

Ce glossaire regroupe la terminologie la plus utile de GitHub (web, Desktop, CLI, API) et des concepts Git associés. Il inclut aussi une cartographie des principaux constituants du logiciel GitHub (Actions, Projects, Packages, Codespaces, etc.).

**I. Glossaire GitHub (A → Z)**

|  |  |  |  |
| --- | --- | --- | --- |
| Terme | Définition | Catégorie | Voir aussi |
| Actions (GitHub Actions) | Plateforme d’automatisation CI/CD de GitHub pour exécuter des workflows (tests, build, déploiements) déclenchés par des événements (push, PR, schedule…). | CI/CD | Workflow, Runner, Jobs, Steps, Marketplace, Secrets |
| Actions Marketplace | Place de marché d’actions réutilisables (steps) publiées par la communauté et des éditeurs. | CI/CD | Actions, Workflow |
| Artifact | Fichiers produits par un workflow (ex. build, rapports de tests) et stockés pour téléchargement. | CI/CD | Workflow, Jobs |
| Assignee | Personne à qui une Issue ou une Pull Request est assignée. | Collaboration | Issue, Pull Request, Review |
| Audit log | Journal des événements d’une organisation/entreprise (sécurité, conformité, administration). | Sécurité/Administration | Organization, Enterprise |
| Base branch | Branche cible dans une Pull Request (celle dans laquelle on fusionne). | Git/PR | Head branch, Merge |
| Blame | Vue ligne par ligne indiquant l’auteur et le commit ayant modifié chaque ligne d’un fichier. | Code/Traçabilité | Commit, History |
| Branch (branche) | Ligne de développement isolée dans un dépôt. | Git | Main, Feature branch, Merge, Rebase |
| Branch protection rules | Règles imposées sur une branche (ex. reviews obligatoires, checks requis, interdiction de force‑push). | Gouvernance | Protected branch, Rulesets |
| Checks / Status checks | Résultats de validations automatiques (CI, quality gates) associés à un commit/PR. | Qualité/CI | Actions, Required status checks |
| Cherry‑pick | Application d’un commit spécifique sur une autre branche (historiquement côté Git). | Git | Rebase, Merge |
| CI/CD | Intégration continue (CI) et Déploiement continu (CD) automatisés. | CI/CD | Actions, Pipeline/Workflow |
| CLI (GitHub CLI – gh) | Outil en ligne de commande pour agir sur GitHub (PR, issues, repos). | Outils | Git CLI, GitHub Desktop |
| Clone | Copie d’un dépôt distant en local. | Git | Fork, Remote |
| CODEOWNERS | Fichier déclarant les propriétaires de code responsables de la revue selon les chemins. | Gouvernance | Review, Protected branch |
| Codespaces | Environnements de dev hébergés (VS Code dans le cloud) liés à un repo. | Dev/Cloud | Actions, Secrets, Container |
| Commit | Instantané (snapshot) de fichiers avec un message décrivant les changements. | Git | Push, Pull, Branch |
| Community health files | Fichiers de bonnes pratiques (README, CONTRIBUTING, CODE\_OF\_CONDUCT, SECURITY). | Qualité | README, SECURITY.md |
| Conflict (merge conflict) | Conflit de fusion quand deux changements incompatibles touchent la même zone de code. | Git/PR | Merge, Rebase |
| Copilot | Assistant IA de GitHub pour l’auto‑complétion et l’aide contextuelle au code. | IA | Codespaces, VS Code |
| Default branch | Branche par défaut (souvent main) utilisée pour les PR et les clones. | Git | Main, Protected branch |
| Dependabot | Service de veille et de mises à jour des dépendances, avec alertes de sécurité. | Sécurité | Security advisories, Dependency graph |
| Deployment | Publication d’une version d’application vers un environnement (staging, prod). | CI/CD | Environment, Actions |
| Discussion | Espace de conversation communautaire par sujet (Q&A, annonces). | Collaboration | Issue, Pull Request |
| Draft pull request | PR marquée comme brouillon pour signaler un travail en cours non prêt à fusionner. | PR | Pull request, Review |
| Enforce linear history | Règle empêchant les merge commits pour garder un historique linéaire. | Gouvernance | Branch protection rules |
| Enterprise (GitHub Enterprise) | Offre GitHub pour entreprises (Cloud ou Server) avec fonctionnalités avancées. | Administration | Organization, SSO/SAML |
| Environment (Actions) | Cible de déploiement avec secrets/approbations propres (ex. production). | CI/CD | Secrets, Deployment |
| Fork | Copie d’un dépôt sur votre compte pour expérimenter/proposer des changements via PR. | Collaboration | Clone, Pull request |
| Free/Pro/Team/Enterprise | Paliers d’offre GitHub (fonctionnalités/collaboration/administration). | Administration | Organization, Billing |
| Gist | Mini‑dépôt pour partager rapidement des extraits de code ou notes. | Partage | Repository |
| Git LFS | Extension Git pour stocker de gros fichiers (binaires) hors du repo. | Stockage | Repository, Releases |
| GitHub App | Intégration basée sur permissions fines et webhooks pour automatiser sur des ressources. | Intégrations | OAuth App, Webhook |
| GitHub Desktop | Client graphique pour gérer Git localement (commit, branch, pull/push). | Outils | Git CLI, GitHub CLI |
| GitHub Packages / GHCR | Registres de paquets & images (npm, Maven, Docker – GitHub Container Registry). | Distribution | Actions, CI/CD |
| GitHub Pages | Hébergement statique depuis un repo (sites docs/portfolio). | Hébergement | Actions, Branch |
| GraphQL API | API de GitHub orientée graphes avec schéma typé. | API | REST API |
| HEAD | Pointeur sur le commit courant dans Git (sommet de la branche checkoutée). | Git | Branch, Commit |
| Hook / Webhook | Notification HTTP envoyée par GitHub à une URL lors d’événements (push, PR, releases…). | Intégrations | GitHub App, API |
| Insights | Section analytique d’un repo (trafic, clones, dépendances, contributeurs). | Observation | Dependency graph, Security |
| Issue | Ticket pour suivre bugs, demandes, tâches, discussions axées travail. | Collaboration | Label, Assignee, Milestone |
| Job (Actions) | Unité de travail dans un workflow, composée d’étapes (steps), exécutée sur un runner. | CI/CD | Workflow, Runner, Step |
| Key – Deploy key | Clé SSH liée à un repo pour accès en lecture (ou écriture) par un système externe. | Sécurité | SSH, Secrets |
| Key – Personal Access Token (PAT) | Jeton d’accès personnel pour l’API/CLI (remplaçant souvent le mot de passe). | Sécurité | REST API, CLI |
| Label | Étiquette de classification sur Issues/PR (bug, help wanted, priorité…). | Collaboration | Issue, Pull request |
| License (LICENSE) | Fichier indiquant les droits d’usage/distribution du code. | Légal | Open source |
| Linear history | Historique de commits sans merge commits (rebase/squash). | Git/Gouvernance | Enforce linear history |
| Main | Nom habituel de la branche par défaut (ex‑master). | Git | Default branch |
| Markdown | Langage de mise en forme léger utilisé pour README, Issues, PR. | Docs | README, Wiki |
| Marketplace | Catalogue d’applis et actions pour étendre GitHub. | Intégrations | Actions, Apps |
| Merge (fusion) | Intégration des changements d’une branche dans une autre. | Git/PR | Squash merge, Rebase merge |
| Merge methods | Modes de fusion : merge commit, squash merge, rebase merge. | PR | Branch protection rules |
| Merge queue | File d’attente automatisée pour fusionner en série après validation des checks. | PR/Qualité | Checks, Protected branch |
| Milestone | Objectif daté regroupant Issues/PR pour un jalon/version. | Planification | Issue, Project |
| Notifications | Système d’alertes (web/email/mobile) sur activités (issues, PR, reviews). | Collaboration | Watch, Subscribe |
| OAuth App | Intégration utilisant OAuth ; permissions souvent plus globales qu’une GitHub App. | Intégrations | GitHub App, REST API |
| Organization | Espace d’équipe contenant des dépôts, équipes, règles et facturation. | Administration | Team, Roles |
| Pages (GitHub Pages) | Hébergement de sites statiques à partir du repo (docs, blogs). | Hébergement | Actions |
| Project / Projects (v2) | Gestion de projet (tableaux, listes, champs personnalisés) liée aux Issues/PR. | Planification | Milestone, Issue |
| Protected branch | Branche avec règles qui restreignent les opérations (merge, force‑push). | Gouvernance | Branch protection rules |
| Pull | Récupérer des changements d’un remote en local (git pull). | Git | Fetch, Merge |
| Pull request (PR) | Proposition de fusion entre deux branches avec revue, checks et discussions. | Collaboration/Code | Review, Merge |
| Push | Envoyer des commits locaux vers le dépôt distant. | Git | Remote, Branch |
| Queue (merge queue) | Mécanisme qui séquence les fusions pour garantir un état vert de la branche cible. | Qualité/PR | Checks, Required status checks |
| README | Page d’accueil du repo (présentation, installation, usage). | Docs | Community files |
| Rebase | Réappliquer une suite de commits au-dessus d’une autre base (réécrit l’historique). | Git | Merge, Linear history |
| Release | Version paquetée d’un projet avec notes de version et artefacts. | Distribution | Tag |
| Remote | Lien nommé vers un dépôt distant (origin, upstream). | Git | Push, Pull |
| Repository (repo) | Conteneur principal du code, issues, PR, wiki, actions, paramètres. | Fondamental | Branch, Commit |
| Required reviews | Nombre minimal d’avis favorables avant fusion. | Gouvernance | Branch protection rules, Review |
| Required status checks | Checks qui doivent être verts avant fusion. | Qualité/Gouvernance | Actions, Branch protection rules |
| REST API | API HTTP classique pour interagir avec GitHub (JSON). | API | GraphQL API |
| Review (code review) | Évaluation d’une PR (commentaires, approval/changes requested). | Qualité | Pull request, CODEOWNERS |
| Rulesets / Repository rules | Règles centralisées (ex. protéger branches/tags, patterns, conditions). | Gouvernance | Branch protection rules |
| Secret | Valeur sensible stockée chiffrée (pour Actions/Codespaces). | Sécurité | Environment, Actions |
| Secret scanning | Détection automatique de secrets exposés dans l’historique. | Sécurité | Security, Alerts |
| Security advisories | Avis de sécurité sur vulnérabilités et remédiations. | Sécurité | Dependabot, Alerts |
| Self‑hosted runner | Machine gérée par vous pour exécuter les workflows Actions. | CI/CD | Runner, Actions |
| Squash merge | Fusion qui compresse tous les commits d’une PR en un seul commit. | PR | Merge methods |
| SSO / SAML / SCIM | Gestion des identités et provisioning pour organisations/entreprises. | Sécurité/Administration | Enterprise |
| Star (étoile) | Marque‑page/favori public d’un repo par un utilisateur. | Social | Watch, Trending |
| Status | État d’un PR/commit (vert/rouge/pendant) selon checks et reviews. | Qualité | Checks |
| Submodule | Lien vers un autre repo embarqué dans un repo Git. | Git | Monorepo, Dependency |
| Tag | Étiquette référençant un commit (souvent pour versionner). | Git | Release |
| Team | Groupe d’utilisateurs avec permissions et mentions @team. | Administration | Organization |
| Template repository | Repo modèle utilisé pour créer de nouveaux repos pré‑configurés. | Productivité | Community files |
| Triage (role) | Rôle d’organisation pour trier/étiqueter issues/PR sans droits d’écriture. | Administration | Roles, Permissions |
| Two‑factor authentication (2FA) | Double authentification renforçant la sécurité des comptes. | Sécurité | SSO, Organization policies |
| Upstream | Nom commun du dépôt d’origine (par ex. le repo source d’un fork). | Git | Remote, Fork |
| User | Compte individuel GitHub. | Administration | Organization, Team |
| Visibility | Niveau d’exposition d’un repo (public, private, internal). | Administration | Organization policies |
| Watch | S’abonner aux notifications d’un repo. | Social | Notifications |
| Webhook | Appel HTTP sortant de GitHub déclenché par un événement. | Intégrations | GitHub App |
| Wiki | Documentation versionnée intégrée au repo. | Docs | README |
| Workflow (Actions) | Fichier YAML décrivant les jobs/steps exécutés par Actions. | CI/CD | Runner, Jobs, Steps |
| Workflow dispatch | Déclenchement manuel d’un workflow depuis l’UI ou l’API. | CI/CD | Actions |

**II. Constituants principaux de GitHub (cartographie)**

|  |  |
| --- | --- |
| Composant | Rôle / Description |
| GitHub.com (Web) | Interface principale : dépôts, Issues/PR, Actions, Projects, Security, Insights, Settings. |
| GitHub Desktop | Client graphique pour Git sur Windows/macOS (cloner, commit, brancher, synchroniser). |
| GitHub CLI (gh) | Commande 'gh' pour gérer PR, issues, repos, workflows depuis le terminal. |
| GitHub Mobile | Application iOS/Android pour notifications, PR/Issues légères. |
| Codespaces | Environnements de dev hébergés, éditables dans le navigateur ou VS Code. |
| GitHub Actions | Moteur CI/CD : workflows YAML, runners hébergés ou self‑hosted, secrets. |
| GitHub Packages (GHCR) | Registres de paquets/images (npm, Maven, NuGet, Docker). |
| GitHub Pages | Hébergement statique à partir d’un repo. |
| API (REST & GraphQL) | Intégration programmatique pour automatiser et interfacer GitHub. |
| Sécurité avancée | Code scanning, secret scanning, dependency review, Dependabot alerts. |
| Projects (v2) | Gestion de projet flexible, champs personnalisés, vues table/kaban/timeline. |
| Discussions & Wiki | Collaboration communautaire et documentation versionnée. |
| Organizations/Teams | Structuration d’une équipe avec rôles, permissions, audit log. |
| Enterprise | Fonctions de gouvernance/SSO/SAML/SCIM, politiques d’orga, audit, facturation. |

**Onglets d’un dépôt (barre supérieure)**

|  |  |
| --- | --- |
| Onglet | Contenu |
| Code | Fichiers, branches, README, releases, tags. |
| Issues | Tickets/bugs/demandes avec labels, assignees, milestones. |
| Pull requests | Propositions de fusion, reviews, checks, conversations. |
| Actions | Workflows CI/CD, runs, logs, artifacts. |
| Projects | Tableaux/listes pour gérer tâches/roadmap. |
| Wiki | Docs versionnées. |
| Security | Alerts, advisories, dépendances, politiques. |
| Insights | Statistiques, graphes de dépendances, trafic. |
| Settings | Options du repo : branches, secrets, pages, rulesets, permissions. |

**Si**

**III. Rôles & permissions (organisation)**

|  |  |
| --- | --- |
| Rôle | Droits |
| Owner | Contrôle total sur l’organisation (réglages, membres, facturation). |
| Maintainer (d’équipe) | Gère une équipe (membres, permissions, mentions). |
| Member | Membre standard d’une organisation. |
| Triage | Peut trier issues/PR (labels/assignations) sans écrire dans le code. |
| Write | Peut pousser des commits/branches sur le repo. |
| Read | Accès en lecture seule au repo. |

**IV. CI/CD (Actions) – Vocabulaire**

|  |  |
| --- | --- |
| Terme | Définition |
| Workflow | Définition YAML du pipeline. |
| Job | Bloc exécuté sur un runner, composé de steps. |
| Step | Action individuelle (run/script/action du Marketplace). |
| Runner | Machine qui exécute le job (hébergé GitHub ou self-hosted). |
| Secret | Variable sensible chiffrée. |
| Environment | Cible (staging/prod) avec règles d’approbation et secrets propres. |
| Artifact | Fichier généré et archivé par un job. |
| Matrix | Exécutions parallèles sur plusieurs paramètres (ex. OS/Node versions). |
| Cache | Stockage d’artefacts de build pour accélérer les runs. |
| Concurrency | Contrôle des exécutions simultanées (annulation des anciens runs). |

**V. À ne pas confondre**

|  |  |
| --- | --- |
| Couple | Différence |
| Git vs GitHub | Git = système de contrôle de versions (local/distribué). GitHub = plateforme web autour de Git (hébergement, PR, issues, CI/CD…). |
| Fork vs Clone | Fork = copie d’un repo sur GitHub (compte différent). Clone = copie locale d’un repo (depuis GitHub ou ailleurs). |
| Issue vs Discussion | Issue = tâche/bug/actionnable. Discussion = échange libre (Q&A, annonces). |
| Merge vs Rebase vs Squash | Trois stratégies différentes pour intégrer des changements et gérer l’historique. |
| Secrets d’org vs secrets de repo vs secrets d’environnement | Portées différentes pour l’usage par Actions/Codespaces. |

**VI. Note Windows 11 (Famille)**

• GitHub Desktop, GitHub CLI (gh) et VS Code sont pleinement compatibles avec Windows 11.  
• Pour une intégration fluide : installez Git for Windows (inclut Git Bash) et configurez vos clés SSH/PAT.