Jenkins Freestyle – Guide des options (Référence rapide)

> *Cette fiche recense les principales sections et options d'un **job Freestyle** Jenkins, avec l'effet de chaque réglage et quand l'utiliser. Certaines options apparaissent uniquement si les **plugins** correspondants sont installés.*

---

1) Général (General)

- **Description** : texte libre affiché en haut du job (but, contacts, consignes).
- **This project is parameterized / Ce build a des paramètres** : permet de définir des paramètres (texte, booléen, choix, fichiers…) injectés au build. Utile si le job doit être rejouable avec des valeurs différentes.
- **GitHub project** : associe une URL GitHub (affiche un lien, utilisé par certains plugins).
- **Discard old builds / Supprimer les anciens builds** : règles de rétention (nombre max de builds, nombre de jours). Évite la saturation disque.
- **Throttle builds** : limite la concurrence (globalement/par catégorie), utile pour jobs coûteux.
- **Execute concurrent builds if necessary / Exécuter des builds simultanément** : autorise plusieurs builds en parallèle pour **ce même job** (à activer seulement si le job est thread-safe).
- **Advanced** (selon version/plugins) :
    - **Quiet period** : délai (en s) avant de démarrer un build après un déclenchement.
    - **Retry Count** : nombre de tentatives si le démarrage échoue.

---

2) Gestion de code source (Source Code Management)

A) None / Aucune

Aucune récupération de code (build purement scripté, artefacts copiés ailleurs, etc.).

B) Git

- **Repository URL** : URL HTTPS/SSH du dépôt (se termine souvent par `.git`).

- **Credentials** : identifiants (user/pass, token, clé SSH). Inutiles pour un repo public en HTTPS.
- **Branches to build** : branches à builder (ex. `*/main`, `*/release/*`).
- **Repository browser** : lien vers l'interface web (GitHub/GitLab/Bitbucket) pour des liens contextualisés dans les logs.
- **Additional Behaviours** (exemples utiles) :
    - **Clean before checkout** : nettoie le workspace avant `checkout` (évite résidus).
    - **Wipe out repository & force clone** : supprime le repo local et refait un clone complet.
    - **Shallow clone** : récupère un historique restreint (plus rapide, moins d'espace).
    - **Sparse checkout** : ne récupère qu'un sous-ensemble de fichiers.
    - **Checkout to a sub-directory** : place les sources dans un sous-dossier du workspace.
    - **Custom user name/e-mail address** : auteur personnalisé pour les opérations Git.
    - **Polling ignores commits with certain messages** : ignore certains commits lors du polling.

C) Subversion (si plugin SVN installé)

Options similaires (URL, credentials, depth, exclusions…).

---

3) Triggers (Déclencheurs)

- **Build periodically** : CRON **temps-réel** (déclenche à heure fixe même sans changement SCM). Ex. `H 8 * * 1-5` → chaque jour ouvré vers 8h.
- **Poll SCM / Scrutation de l'outil de gestion de version** : CRON qui **vérifie** s'il y a des changements et déclenche seulement s'il y en a. Ex. `H/5 * * * *` → toutes les 5 minutes.
- **GitHub hook trigger for GITScm polling** : démarre le build **sur webhook GitHub** (push, PR…). Plus réactif que le polling.
- **Trigger builds remotely (e.g., from scripts)** : URL avec **token** pour déclencher le job à distance (curl, scripts). À sécuriser.
- **Build after other projects are built** : chaîne de jobs (downstream) après réussite/échec du job amont.
- **Pipeline/Multibranch** : non applicable aux Freestyle (autres types de jobs).

---

4) Build Environment (Environnement du build)

- **Delete workspace before build starts** : supprime le workspace au début du build. Pratique pour des builds reproductibles (plus lent).
- **Use secret text(s) or file(s)** : injecte des secrets/variables depuis Jenkins Credentials.
- **Add timestamps to the Console Output** : horodate chaque ligne des logs (debug).
- **Color ANSI** (si plugin) : colorise la console (utile pour scripts qui sortent des couleurs).
- **Provide Node & npm bin/Installation** (si plugin NodeJS) : configure Node pour le job.

- **With Ant / With Maven** (anciens usages) : utiliser des installations d'outils déclarées dans Jenkins. **Conseil** : préférez les **wrappers** (Maven/Gradle) proches du projet.

---

5) Build Steps (Étapes du build)

- **Execute Windows batch command** : exécute un script `.bat` (Windows).
- **Execute shell** : exécute un shell script (Linux/macOS).
- **Invoke Ant** : lance des targets Ant (si Ant installé/configuré).
- **Invoke Gradle script** : lance Gradle (ou via `gradlew` recommandé).
- **Invoke top-level Maven targets** : lance Maven **global** de Jenkins (moins portable).
  Conseil : dans les projets modernes, utilisez plutôt le **Maven Wrapper** via *Execute shell/batch* :

  ```
  # Linux/macOS
  chmod +x mvnw
  ./mvnw -B clean verify
  ```

  ```
  rem Windows
  call mvnw.cmd -B clean verify
  ```

- **Set build status on GitHub commit** (si plugin GitHub) : remonte le statut (success/failure).

---

6) Post-build Actions (Actions à la suite du build)

- **Archive the artifacts** : conserve des fichiers (ex. `target/*.jar`) en artefacts du build.
- **Publish JUnit test result report** : parse `target/surefire-reports/*.xml`, crée l'onglet **Test Result** (tests, échecs, tendances).
- **Record JaCoCo coverage report** : parse `target/site/jacoco/jacoco.xml`, crée l'onglet **JaCoCo** (pourcentages, tendances). Possible de fixer des **seuils** qui marquent le build **UNSTABLE/FAILED** si couverture insuffisante.
- **Publish HTML reports** : publie un dossier HTML (ex. `target/site/jacoco`, index `index.html`). Ajoute un lien direct sur la page du job.
- **Build other projects** : déclenche d'autres jobs après ce build (downstream).
- **E-mail Notification / Mailer** : envoie un email selon l'état du build.
- **Fingerprinting** : trace les artefacts d'un job à l'autre (traçabilité).
- **Deploy artifacts (Maven)** : déploiement vers un repo Maven (Nexus/Artifactory), si configuré.

---

7) Conseils pratiques

- **Wrappers d'outils** (Maven/Gradle) → plus **portables** et reproductibles que les installations globales.
- **Chemins de rapports** : soyez précis (`target/surefire-reports/*.xml`, `target/site/jacoco/jacoco.xml`).
- **Logs** : activez les **timestamps** et, si possible, l'ANSI color pour une lecture plus claire.
- **SCM** : pour la stabilité, ajoutez **Clean before checkout** si vous alternez les branches.
- **Sécurité** : utilisez les **Credentials** Jenkins pour tokens/SSH et limitez les remote triggers via token.

---

8) Exemples de CRON utiles

- `H/5 * * * *` → toutes les **5 minutes** (réparti).
- `H 8 * * 1-5` → chaque jour ouvré vers **08:00**.
- `H H/2 * * *` → toutes les **2 heures**.
- `H 0 1 * *` → le **1er du mois**.