VIBESDEV.FR · CLAUDE CODE GUIDE 2026

Claude Code Cheatsheet

Architecture · Skills · Hooks · Agents · Workflow · Commandes

[GRATUIT]

CLAUDE.md | Skills | Hooks | Agents | Java · Spring | Angular | Kafka | Docker

ARCHITECTURE

Les 4 couches Claude Code

L1 CLAUDE.md — Contexte persistant
`./CLAUDE.md · ~/.claude/CLAUDE.md`
Mémoire long-terme : stack, règles, conventions, gotchas. Lu automatiquement à chaque session.

L2 Skills — Packs de connaissances
`.claude/skills/<name>/SKILL.md`
Guides markdown auto-invoqués selon la tâche. La description est la clé de déclenchement.

L3 Hooks — Garde-fous déterministes
`.claude/settings.json → hooks`
Callbacks PreToolUse / PostToolUse. Exit 0 = ok, Exit 2 = bloque l'action.

L4 Agents — Sous-tâches autonomes
`.claude/agents/security-reviewer.md`
Agents spécialisés avec contexte propre pour workflows complexes en parallèle.

TEMPLATE

CLAUDE.md — Spring Boot

```
# Projet: MonApp
stack: Java 21 · Spring Boot 3 · Angular 18
kafka: Confluent Cloud · 3 topics
db: PostgreSQL 16 · Flyway migrations

## Build & Test
./mvnw test -pl core
./mvnw spring-boot:run -Dspring.profiles.active=dev
npm run test --prefix frontend

## Structure
/src/main/java/com/app/
  domain/     ← Entités JPA + services
  kafka/      ← Consumers / Producers
  web/rest/   ← REST Controllers

## Gotchas critiques
- NE JAMAIS toucher flyway_schema_history
- Valider XSD avant envoi TTN
- Toujours @Transactional sur les services
```

DAILY PATTERN

Workflow quotidien

01 SETUP
Init + vérifier CLAUDE.md
`claude /init` · scanner

02 PLANIFICATION
Activer Plan Mode
`Shift+Tab+Tab` · décrire l'intention complète

03 EXÉCUTION
Auto Accept pour aller vi
`Shift+Tab` · `Esc Esc` po annuler

04 CONTEXTE
Compresser en longue sess
`/compact` · évite la d

05 DISCIPLINE
1 session = 1 feature · c souvent
`/clear` entre features

COMMANDES

Commandes slash

COMMANDE	ACTION
/init	Génère CLAUDE.md en scannant le projet
/doccat	Vérifie l'installation & config
/compact	Compresse le contexte de la session
/memory	Édite la mémoire persistante globale
/clear	Reset la conv. (garde la mémoire)

Raccourcis clavier

Shift+Tab+Tab	Plan Mode on/off
Shift+Tab	Changer de mode (auto-accept)
Tab	Toggle Extended Thinking
Esc Esc	Remonter au menu / annuler
Ctrl+C	Interrompre l'exécution

SKILLS

Skills prêts à l'emploi

kafka-consumer
`.claude/skills/kafka/SKILL.md`
Patterns retry, DLQ, idempotency. Auto-déclenché sur "listener", "consumer", "topic".

jpa-entity
`.claude/skills/jpa/SKILL.md`
Conventions @Entity, @Audited, Flyway migration auto. Déclenché sur "entité", "JPA", "repository".

api-design
`.claude/skills/api/SKILL.md`
REST conventions, DTOs, validation Bean. Déclenché sur "endpoint", "controller", "REST".

code-review
`.claude/skills/review/SKILL.md`
Checklist sécurité, perf, couverture tests. Déclenché sur "review", "relire", "audit".

docker-deploy
`.claude/skills/docker/SKILL.md`
Patterns Dockerfile multi-stage, compose, healthchecks. Déclenché sur "docker", "deploy".

HOOKS

Configuration Hooks

```
// .claude/settings.json
{
  "hooks": {
    "PreToolUse": [
      {
        "matcher": "Bash",
        "hooks": [{
          "type": "command",
          "command": "scripts/gua
          "timeout": 5
        }]
      }
    ]
  },
  "permissions": {
    "allow": ["Read:*", "Write:s
    "deny":  ["Read:*.env*", "Bas
  }
}
```

PreToolUse → Exit 2 = BLOQUE
Valide avant d'écrire. Idéal pour DB, check migrations Flyway.

PostToolUse → Notification
Notifie après écriture. Tests auto check, alerte Slack.

MÉMOIRE

Hiérarchie des fichiers

```
~/.claude/CLAUDE.md    ← global (tous projets)
~/CLAUDE.md            ← monorepo root
./CLAUDE.md            ← projet (partagé git)
./src/CLAUDE.md        ← sous-module scoped
```

SÉCURITÉ

Permissions à configurer

✓ Interdire `Bash:sudo:*` et `Bash:rm -rf*`
✓ Bloquer lecture `Read:*.env*` et `Read:*.pem`
✓ Limiter écriture à `Write:src/**` uniquement

JAVA · SPRING · KAFKA

Astuces spécifiques

CLAUDE.md obligatoires
Toujours inclure : Java version, S profiles, conventions packages, rè topics Kafka actifs.

Skill Kafka pattern

Les fichiers enfants ajoutent du contexte, n'écrasent jamais le parent. Max 200 lignes / fichier.

```
# Référencer des docs avec @
@docs/api-spec.md        ← Claude lit le fichier
@schema/teif-1.8.8.xsd   ← validation XSD
@src/kafka/CONSUMER.md   ← patterns consumers
```

- ✓ Autoriser git en lecture : `Bash:git:*`
- ✓ Hook guard.sh pour valider les migrations Flyway
- ✓ Toujours Plan Mode sur les tickets complexes
- ✓ Exit 2 dans hooks = bloque toute la chaîne

Documenter retry policy, DLT topic idempotency key dans SKILL.md pour les respecte sans rappel.

JHipster / JPA

Préciser conventions: @Entity avec nommage snake_case en base, soft-d actif.

Angular in monorepo

Ajouter ./frontend/CLAUDE.md avec Angular, routing, standalone compo signal-based.

AGENTS & MCP

Agents spécialisés + MCP Servers

```
# .claude/agents/security-reviewer.md
name: security-reviewer
description: Analyse sécurité OWASP,
  injection SQL, secrets exposés
tools: [Read, Grep, Glob]
```

- ✓ security-reviewer.md → audit OWASP
- ✓ db-migrator.md → Flyway séquence
- ✓ api-tester.md → tests automatiques

MCP Servers disponibles :

@github/mcp-server	Issues, PRs, repos
@anthropic/mcp-server-filesystem	Fichiers locaux
@modelcontextprotocol/server-postgres	DB PostgreSQL
@modelcontextprotocol/server-slack	Notifications

```
# ~/.claude/settings.json
"mcpServers": {
  "postgres": {
    "command": "npx",
    "args": ["-y", "@mcp/postgres",
             "postgresql://localhost/mydb"]
  }
}
```