

# Guide Pratique des Commandes Slash

---

## Introduction

Ce guide présente toutes les commandes slash disponibles dans le projet Quelyos ERP Suite. Ces commandes automatisent les tâches courantes de développement, déploiement et maintenance.

**Usage :** Tapez `/nom-commande` dans Claude Code pour exécuter une commande.

---

## 1. DevOps - Git & Déploiement

Commandes pour gérer le code source et les déploiements.

Commande	Description	Usage
<code>/commit</code>	Créer un commit local avec message conventionnel	<code>/commit</code>
<code>/ship</code>	Commit + push vers GitHub	<code>/ship</code>
<code>/deploy</code>	Checklist déploiement production	<code>/deploy</code>
<code>/test</code>	Lancer la suite de tests complète	<code>/test</code>

### `/commit` - Commit Rapide

Crée un commit local **sans pusher**. Idéal pour préparer plusieurs commits avant un push groupé.

**Workflow :**

1. Analyse les fichiers modifiés
2. Demande le type (feat, fix, chore, refactor)
3. Suggère un message basé sur les changements
4. Crée le commit avec co-author Claude

**Types de commit :**

- `feat:` - Nouvelle fonctionnalité
- `fix:` - Correction de bug
- `chore:` - Maintenance
- `refactor:` - Refactoring sans changement de comportement
- `docs:` - Documentation
- `test:` - Tests
- `perf:` - Performance

### `/ship` - Commit & Push

Comme `/commit` mais avec push automatique vers la branche `main`.

**Différence avec /commit :**

- `/commit` = local uniquement
  - `/ship` = local + push remote
-

## 2. Odoo - Serveurs & Modules

Commandes spécifiques à la gestion du backend Odoo.

Commande	Description	Port
/upgrade-odoo	Upgrader module après modifications	8069
/restart-odoo	Redémarrer serveur Odoo	8069
/restart-backoffice	Redémarrer dashboard React	5175
/restart-vitrine	Redémarrer site vitrine	3000
/restart-eCommerce	Redémarrer e-commerce	3001
/restart-all	Redémarrer tous les services	-

### /upgrade-odoo - Upgrade Module

**OBLIGATOIRE** après modification de :

- Modèles ( `models/*.py` )
- Vues ( `views/*.xml` )
- Données ( `data/*.xml` )
- Sécurité ( `security/*.csv` )
- Manifest ( `__manifest__.py` )

**Workflow :**

1. Vérifie Docker actif
2. Propose backup DB (optionnel)
3. Exécute upgrade via Docker
4. Redémarre Odoo
5. Vérifie le health check

**Commande exécutée :**

```
docker-compose run --rm odoo odoo -d quelyos -u quelyos_api --stop-after-init
```

### /restart-\* - Redémarrage Services

Redémarre un service spécifique sans toucher aux autres.

**Ports fixes (NE JAMAIS MODIFIER) :**

- Vitrine : 3000
- E-commerce : 3001
- Dashboard : 5175
- Odoo : 8069

## 3. Qualité - Audits & Vérifications

Commandes pour assurer la qualité du code et de l'interface.

Commande	Description	Score

/uiux	Audit UI/UX complet d'une page	/140
/coherence	Audit cohérence tri-couche	/100
/parity	Parité Odoo ↔ Quelyos	-
/polish	Refactoring complet	-
/clean	Nettoyage fichiers/imports	-
/docs	Sync documentation	-
/analyze-page	Analyse page + plan admin	-

### /uiux - Audit UI/UX

Audit complet d'une page dashboard selon la charte UI/UX à 140 points.

**Usage :**

```
/uiux src/pages/finance/budgets/page.tsx
/uiux --fix src/pages/crm/Leads.tsx
/uiux --module finance
```

**Sections évaluées :**

1. Structure de Base (25 pts) - Layout, Breadcrumbs, Header, PageNotice
2. Menus et Navigation (20 pts) - Tabs, Dropdowns, états
3. Composants Standard (25 pts) - SkeletonTable, Button, icônes
4. États et Erreurs (20 pts) - Loading, Error, Empty
5. Dark Mode (15 pts) - Toutes variantes adaptatives
6. Documentation (10 pts) - JSDoc
7. Responsive (5 pts) - Breakpoints
8. Composants Enfants (20 pts bonus)

**Grades :**

- S+ : 140/140
- S : 130-139
- A : 110-129
- B : 90-109
- C : < 90

### /coherence - Audit Tri-Couche

Vérifie la cohérence entre les 3 couches :

- Backend Odoo (modèles, API)
- API Controllers (endpoints)
- Frontend React (hooks, pages)

**Vérifie :**

- Champs présents dans les 3 couches
- Types cohérents (string ↔ Char, number ↔ Float)
- Endpoints correspondant aux hooks
- Noms de champs anonymisés

### /parity - Parité Fonctionnelle

Compare les fonctionnalités Odoo natives avec l'implémentation Quelyos pour identifier les gaps.

## 4. Architecture - Réflexion & Conception

Commandes pour guider les décisions architecturales.

Commande	Description
/leverage	Capitaliser sur existant Odoo
/no-odoo	Vérifier anonymisation

### /leverage - Capitalisation Odoo

#### À UTILISER AVANT TOUTE IMPLÉMENTATION

Vérifie systématiquement ce qu'Odoo offre nativement pour éviter de réinventer la roue.

**Usage :**

```
/leverage point de vente  
/leverage gestion employés  
/leverage facturation
```

**Workflow :**

1. Identifie les modules Odoo pertinents
2. Liste les modèles natifs exploitables
3. Génère matrice de décision (natif vs héritage vs custom)
4. Recommande l'approche optimale

#### Règle d'Or - TOUJOURS utiliser natif pour :

- Comptabilité → `account.move`
- Stock → `stock.picking`, `stock.move`
- Produits → `product.product`
- Contacts → `res.partner`
- Utilisateurs → `res.users`

### /no-odoo - Anonymisation

Vérifie qu'aucune référence "Odoo" n'est exposée dans les frontends.

**Recherche :**

- Variables : `OdooClient`, `ODOO_URL`
- Messages d'erreur : "Odoo returned..."
- Champs API : `list_price`, `default_code`
- Jargon : `OCA`, `ir.model`, `res.partner`

## 5. Sécurité & Performance

Commandes pour auditer sécurité et performance.

Commande	Description
----------	-------------

/security	Audit sécurité OWASP
/perf	Analyse performance
/db-sync	Vérification DB

### /security - Audit Sécurité

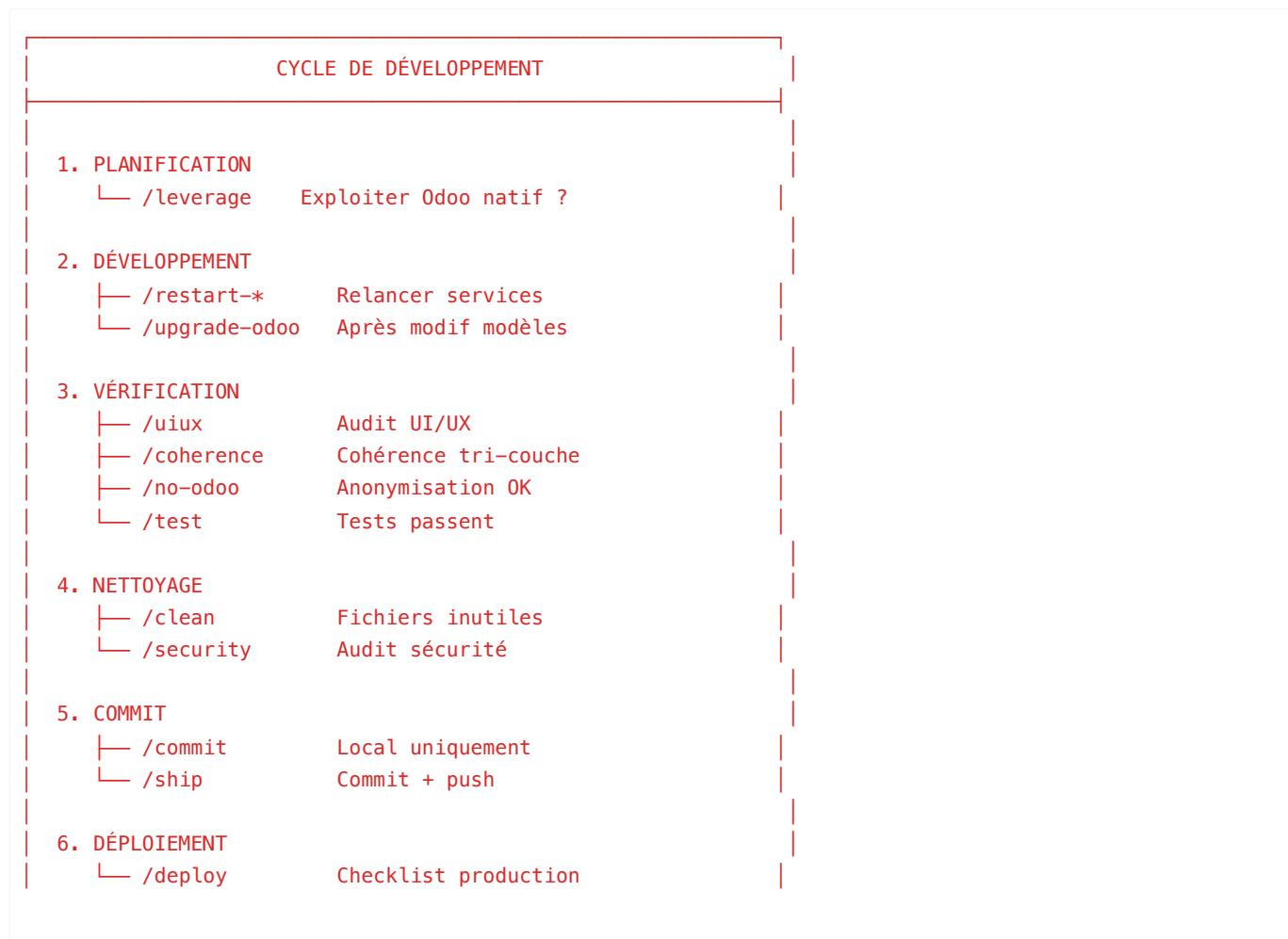
Audit multi-niveaux basé sur OWASP :

- Injection SQL/XSS
- Authentification/Sessions
- Permissions/RBAC
- Secrets exposés
- Headers sécurité

### /perf - Analyse Performance

- Requêtes N+1
- Bundle size
- Temps de réponse API
- Optimisation images
- Lazy loading

## 6. Workflow Recommandé



## 7. Référence Rapide

### Par Fréquence d'Usage

Fréquence	Commandes
Quotidien	<code>/commit</code> , <code>/ship</code> , <code>/restart-*</code> , <code>/upgrade-odoo</code>
Régulier	<code>/uiux</code> , <code>/leverage</code> , <code>/no-odoo</code> , <code>/test</code>
Ponctuel	<code>/coherence</code> , <code>/parity</code> , <code>/clean</code> , <code>/polish</code>
Rare	<code>/deploy</code> , <code>/security</code> , <code>/perf</code> , <code>/db-sync</code>

### Par Catégorie

Catégorie	Commandes
<b>Git</b>	<code>/commit</code> , <code>/ship</code>
<b>Serveurs</b>	<code>/restart-odoo</code> , <code>/restart-backoffice</code> , <code>/restart-vitrine</code> , <code>/restart-e-commerce</code> , <code>/restart-all</code>
<b>Odoo</b>	<code>/upgrade-odoo</code> , <code>/leverage</code> , <code>/no-odoo</code> , <code>/parity</code>
<b>Qualité</b>	<code>/uiux</code> , <code>/coherence</code> , <code>/polish</code> , <code>/clean</code> , <code>/docs</code>
<b>Sécurité</b>	<code>/security</code> , <code>/test</code>
<b>DevOps</b>	<code>/deploy</code> , <code>/perf</code> , <code>/db-sync</code>

## 8. Bonnes Pratiques

### Avant de coder

- Toujours lancer `/leverage` pour vérifier l'existant Odoo
- Lire le code existant avant de modifier

### Pendant le développement

- `/upgrade-odoo` après chaque modification de modèle
- Incrémenter la version dans `__manifest__.py`

### Avant de commiter

- `/uiux` sur les pages modifiées
- `/no-odoo` pour vérifier l'anonymisation
- `/test` pour valider les tests

### Conventions de commit

```
feat: nouvelle fonctionnalité  
fix: correction de bug  
chore: maintenance  
refactor: refactoring  
docs: documentation  
test: tests  
perf: performance
```

---

## 9. Aide & Support

**Documentation complète :** [.claude/commands/\\*.md](.claude/commands/*.md)

**Commandes built-in Claude Code :**

- `/help` - Aide générale
- `/clear` - Effacer conversation
- `/compact` - Résumer conversation

**Contact :** <https://quelyos.com>

---

*Guide généré automatiquement - Janvier 2026 Quelyos ERP Suite v19.0*