AI TOOLS WIKI · MÉMO

Paradev *mémo*

Démarrage, commandes et garde-fous pour **Paradev** (anciennement Superset). Voir aussi **les avis sur Paradev**. La CLI s'appelle toujours `superset` ; `paradev` en est un alias.

Famille Postes de pilotage multi-agent · **Éditeur** Paradev (anciennement Superset) · **Licence** MIT
Note du wiki 11/15 · **Commandes vérifiées le** 15 novembre 2026
Fiche, notes et avis aitoolswiki.viellard.fr/outils/postes-de-pilotage/paradev

Installation

`curl -fsSL https://paradev.com/cli/install.sh | sh`	CLI autonome : macOS ou Linux (x64, arm64)
`brew install paradevai/tap/superset`	Alternative Homebrew
`superset auth login` `superset auth whoami`	Se connecter, puis vérifier la session

L'application de bureau inclut aussi la CLI dans `~/.paradev/bin`. Ajouter ce dossier au `PATH` pour l'utiliser depuis un terminal externe : `export PATH="$HOME/.paradev/bin:$PATH"`. En CI, utiliser une clé d'API : `superset auth login --api-key …` ou la variable `SUPERSET_API_KEY`.

Cycle minimal

`superset projects create --name "mon-app" --local --import /chemin/mon-app`	Enregistrer un dépôt local
`superset projects list` `superset workspaces list --local`	Lister projets et workspaces

`superset workspaces create \`
`  --project PROJECT_ID \`
`  --name "fix-login" \`
`  --branch fix/login \`
`  --local`
	Créer un worktree isolé

Cycle minimal (suite)

`superset workspaces create \`
`  --project PROJECT_ID \`
`  --name "fix-login" \`
`  --branch fix/login \`
`  --local \`
`  --agent codex \`
`  --prompt "Corrige le bug et lance les tests ciblés"`
	Créer et démarrer directement un agent

`superset workspaces create --session --local --agent claude --prompt "Explore cette idée"`	Session temporaire sans projet existant

Configuration d'un projet

Créer `.paradev/config.json` (un dépôt configuré avant le renommage garde `.superset/config.json`, que Paradev continue de lire) :

```
{
  "setup": [
    "bun install",
    "cp \"$SUPERSET_ROOT_PATH/.env\" .env"
  ],
  "teardown": ["docker compose down"],
  "run": ["./.paradev/run.sh"]
}
```

Étape	Conseil
`setup`	Installer ou lier les dépendances et préparer les fichiers ignorés.
`run`	Attribuer un port distinct à chaque workspace.
`teardown`	Arrêter conteneurs, watchers et ressources temporaires.
Secrets	Ne jamais les committer ; les copier depuis le checkout racine ou un gestionnaire.

Variables disponibles dans les scripts : `SUPERSET_ROOT_PATH` (checkout racine), `SUPERSET_WORKSPACE_NAME`, `SUPERSET_WORKSPACE_PATH`.

Commandes pratiques

`superset workspaces list --json`
`superset workspaces list --quiet`
`superset tasks list --assignee-me`
`superset automations list`
`superset automations run AUTOMATION_ID`
`superset automations logs AUTOMATION_ID`
`superset config check`

`--json` convient aux scripts et `--quiet` renvoie seulement les identifiants lorsque cela est possible. `superset` seul ouvre un navigateur interactif des commandes ; `superset --help` en donne la liste.

Choisir le bon mode

Situation	Mode conseillé
Deux fonctionnalités indépendantes	Deux worktrees et deux branches.
Implémentation puis revue du même diff	Même workspace, sessions distinctes.
Exploration jetable	Session sans projet (`--session`).
Petite correction dans le checkout courant	`--checkout local`, avec prudence.
Agents modifiant les mêmes fichiers	Séquentiel, ou branches puis fusion explicite.

Routine de livraison

1. Lire le diff et les fichiers non suivis.
2. Exécuter tests, lint et build dans le workspace.
3. Envoyer les commentaires de revue à un agent distinct.
4. Commit et push seulement après validation.
5. Créer la PR, vérifier la CI, puis fusionner.
6. Archiver le workspace et exécuter le teardown.

Sources (documentation officielle)

- Premiers pas avec la CLI
- Référence CLI
- Scripts de cycle de vie