

DHTI CLI Cheat sheet

Installation & Setup

```
npx dhti-cli help # Show available commands
```

Command Reference

compose - Manage Docker Compose Services

Generate and manage docker-compose.yml files from modules.

Operations: add, delete, read, reset, env

Common Usage:

```
npx dhti-cli compose add -m langserve -m openmrs
```

```
npx dhti-cli compose delete -m redis
```

```
npx dhti-cli compose read
```

```
npx dhti-cli compose env -e FHIR_BASE_URL -v
http://example.com -s langserve
```

Modules: langserve, openmrs, ollama, langfuse, cqlFhir, redis, neo4j, mcpFhir, mcpX, docktor, medplum, fhir, gateway, webui

Key Flags:

- f, --file — Docker compose file path (default: ~/dhti/docker-compose.yml)
- m, --module — Module to add/delete (multiple allowed)
- e, --env — Environment variable name
- v, --value — Environment variable value
- s, --service — Service name to update (default: langserve)
- host — Use host env pattern (\${VAR:-default})
- dry-run — Preview changes without executing

conch - OpenMRS Frontend Development

Initialize, install, or start OpenMRS ESM frontend development.

Operations: init, install, start

Common Usage:

```
npx dhti-cli conch init -n my-app -w ~/projects
```

```
npx dhti-cli conch install -n my-app -w ~/projects -b
develop
```

```
npx dhti-cli conch start -n my-app -w ~/projects
```

```
npx dhti-cli conch start -n my-app -w ~/projects -s
packages/esm-chatbot-agent
```

Key Flags:

- n, --name — Name of the conch (required for init/install/start)
- w, --workdir — Working directory (default: ~/dhti)
- b, --branch — Git branch to install from (default: develop)
- g, --git — GitHub repo (default: dermatologist/openmrs-esm-dhti-template)
- s, --sources — Additional sources to include when starting
- dry-run — Preview changes without executing

docker - Build & Manage Docker Containers

Build Docker images and manage container operations.

Common Usage:

```
npx dhti-cli docker -n my-image -t elixir
```

```
npx dhti-cli docker -u # Start containers
```

```
npx dhti-cli docker -d # Stop containers
```

```
npx dhti-cli docker -g # Restart gateway container
```

```
npx dhti-cli docker -r dhti-langserve-1 # Restart
```

```
npx dhti-cli docker bootstrap -f
~/dhti/elixir/app/bootstrap.py
```

Key Flags:

- n, --name — Container name to build
- t, --type — Service type: elixir or conch (default: elixir)
- f, --file — Docker compose file path (default: ~/dhti/docker-compose.yml)
- c, --container — Target container name (default: dhti-langserve-1)
- u, --up — Run docker-compose up -d
- d, --down — Run docker-compose down
- g, --gateway — Restart gateway container
- r, --restart — Restart specific container by name
- dry-run — Preview changes without executing

elixir - Manage Backend Elixirs (Python Services)

Install, uninstall, or manage elixir dependencies.

Operations: init, install, uninstall, dev, start

Common Usage:

```
npx dhti-cli elixir init -n my-elixir -w ~/dhti
```

```
npx dhti-cli elixir install -n my-elixir -g
dermatologist/my-repo
```

```
npx dhti-cli elixir uninstall -n my-elixir
```

```
npx dhti-cli elixir dev -n my-elixir -d ~/dev/my-elixir
```

```
npx dhti-cli elixir start -n my-elixir --elixir
http://localhost:8001/langserve/my_elixir/cds-services
```

Key Flags:

- n, --name — Name of the elixir
- w, --workdir — Working directory (default: ~/dhti)
- g, --git — GitHub repo URL to install from
- b, --branch — Git branch (default: develop)
- p, --pypi — PyPi package spec (e.g., "dhti-elixir-base>=0.1.0")
- l, --local — Local directory path for installation
- e, --whl — .whl file to install
- s, --subdirectory — Subdirectory for monorepos
- v, --repoVersion — Elixir version (default: 0.1.0)
- f, --fhir — FHIR endpoint URL (default: http://hapi.fhir.org/baseR4)
- c, --container — Container name (default: dhti-langserve-1)
- d, --dev — Dev folder for live reload
- dry-run — Preview changes without executing

synthea - Synthetic FHIR Data Generation

Manage Synthea for generating synthetic patient data.

Subcommands: install, generate, upload, delete, download

Common Usage:

```
npx dhti-cli synthea install
```

```
npx dhti-cli synthea generate -p 50
```

```
npx dhti-cli synthea generate -p 10 -s MA -c Boston -g M -a 18-65
```

```
npx dhti-cli synthea upload -e http://fhir:8005/baseR4 -t bearer-token
```

```
npx dhti-cli synthea download --covid19
```

```
npx dhti-cli synthea delete
```

****Key Flags:****

- -w, --workdir — Working directory (default: ~/dhti)
- -p, --population — Number of patients to generate (default: 1)
- -s, --state — State for patient generation
- -c, --city — City for patient generation
- -g, --gender — Patient gender (M or F)
- -a, --age — Age range (e.g., "0-18")
- --seed — Random seed for reproducibility
- -e, --endpoint — FHIR server URL (default: http://fhir:8005/baseR4)
- -t, --token — Bearer token for authentication
- Download Flags: --covid19, --covid19_10k, --covid19_csv, --covid19_csv_10k, --synthea_sample_data_csv_latest, --synthea_sample_data_fhir_latest, --synthea_sample_data_fhir_stu3_latest
- --dry-run — Preview changes without executing

docktor - Manage MCPX Inference Pipelines

Install, remove, restart, or list inference pipelines.

****Operations:**** install, remove, restart, list

****Common Usage:****

```
npx dhti-cli docktor install my-pipeline -i my-image:latest -m ./models
```

```
npx dhti-cli docktor remove my-pipeline
```

```
npx dhti-cli docktor list
```

```
npx dhti-cli docktor restart
```

****Key Flags:****

- -c, --container — MCPX container name (default: dhti-mcp-x-1)
- -i, --image — Docker image for inference pipeline (required for install)
- -m, --model-path — Local model directory path
- -e, --environment — Environment variables (format: VAR=value, multiple allowed)
- -w, --workdir — Working directory (default: ~/dhti)

mimic - FHIR Data Import from MIMIC-IV

Submit FHIR import requests to a FHIR server.

****Common Usage:****

```
npx dhti-cli mimic http://localhost/fhir/$import
```

```
npx dhti-cli mimic http://fhir:8005/baseR4/$import -t bearer-token
```

****Arguments:****

- server — FHIR server URL (default: http://localhost/fhir/\$import)

****Key Flags:****

- -t, --token — Bearer token for authentication
- --dry-run — Preview request without sending

synthetic - Generate Synthetic Data with LLM

Process data using LLM to generate synthetic outputs.

****Arguments:****

- input — Input file path
- output — Output file path (required)
- prompt — Prompt file path

****Common Usage:****

```
npx dhti-cli synthetic input.json output.json prompt.txt
```

```
npx dhti-cli synthetic output.json --maxCycles 5 -r 10 -m 0
```

```
npx dhti-cli synthetic input.json output.json -i input -o output
```

****Key Flags:****

- -i, --inputField — Input field name (default: input)
- -o, --outputField — Output field name (default: output)
- -m, --maxCycles — Max cycles for generation (0 = process file records)
- -r, --maxRecords — Max records to process (default: 10)
- --dry-run — Preview changes without executing

Common Patterns

****Full Stack Setup:****

```
npx dhti-cli compose add -m openmrs -m langserve
```

```
npx dhti-cli docker -u # Start everything
```

```
npx dhti-cli elixir install -n my-service -g my/repo
```

****Development Workflow:****

```
npx dhti-cli elixir dev -n my-service -d ~/dev/my-service
```

```
npx dhti-cli docker -g # Restart gateway if needed
```

****Data Generation & Upload:****

```
npx dhti-cli synthea generate -p 100
```

```
npx dhti-cli synthea upload -e http://fhir:8005/baseR4
```

Global Options

All commands support:

- --dry-run — Preview changes without executing
- --help — Show command help
- --version — Show CLI version