

Agent Guide for kelpie-carbon-v1

This document serves as a reference for AI assistants (e.g., Cursor, Claude) to guide the development of the `kelpie-carbon-v1` project. Follow each section in order, using the provided prompts and conventions.

1. Overview

Project name: `kelpie-carbon-v1`

Python package path: `src/kelpie_carbon_v1/`

CLI entry point: `kelpie-carbon-v1`

The project is built incrementally in discrete phases, each comprising:

- **Code** (production files)
- **Tests** (pytest)
- **Docs stubs** (MkDocs)
- **CI / commits**

Failure at any step should halt further work until resolved.

2. Conventions & Tooling

- **Environment:** Poetry-managed; run `poetry install --only main,dev`.
 - **Formatting & static checks:** pre-commit (black, isort, mypy, flake8).
 - **Testing:** pytest; aim for passing tests at each commit.
 - **Docs:** MkDocs Material; docs reside under `docs/`.
 - **Versioning:** Conventional Commits (e.g., `feat(x):`, `fix(x):`, `ci:`).
 - **DVC:** Track large artefacts (models, rasters) under DVC.
 - **CI workflows:** `.github/workflows/ci.yml` (fast-check + docs) and optional cron.
-

3. Development Phases

Phase 0 · Bootstrap Toolchain

Goal: Fresh repo skeleton with Poetry, pre-commit, directories, and initial commit.

Key Files / Commits: `pyproject.toml`, `.pre-commit-config.yaml`, `docs/index.md`, empty `__init__.py` files.

Commit: `chore: bootstrap kelpie-carbon-v1 toolchain & skeleton`.

Phase 1 · CLI Walking Skeleton

Goal: Scaffold Typer-based `hello` command + test.

Files: `cli.py`, `test_cli.py`; add script entry in `pyproject.toml`.

Commit: `feat(cli): add hello command with test`.

Phase 2 · Fast CI Setup

Goal: Add GitHub Actions for lint, type-check, tests on push/pr.

File: `.github/workflows/ci.yml` with `fast-check`.

Commit: `ci: add fast-check workflow`.

Phase 3 · Indices Function (TDD)

Goal: Implement `floating_algae_index` stub + test + docs stub.

Files: `indices.py`, `test_indices.py`, `docs/pipeline/indices.md`.

Commit: `feat(indices): add floating_algae_index with tests and docs stub`.

Phase 4 · Docs Build Job

Goal: Extend CI with `docs-build` job to run MkDocs build.

Commit: `ci: add docs-build job`.

Phase 5 · Pipeline Slices

Goal: Create stubs, tests, and docs for each slice so the app never breaks.

Slices: Fetch, Mask, Model, API.

Commit: `feat(<slice>): add stub, test, and docs`.

Phase 6 · DVC Integration

Goal: Track `models/biomass_rf.pkl` with DVC, configure remote, push artefacts.

Commit: `build: enable DVC for model artefacts`.

4. Next Phases to Full Web App

Once the skeleton and basic slices are in place, continue to:

Phase 7 · Web UI & Extent Selection

- **Implement** a React (or lightweight) web interface under `dashboard/` or `web/` that allows:
 - Drawing or selecting a geographic extent (AOI).
 - Submitting date ranges.
- **Integrate** with the FastAPI backend: `/run?ao...` endpoint to trigger pipeline.
- **Tests:** End-to-end UI tests (e.g. Cypress or Playwright) for extent selection.

Phase 8 · Real Satellite Data Ingestion

- **Replace stubs** with actual Landsat/Sentinel-2 ingestion in `fetch.py`:
- Use `sentinelhub` or `rasterio` to pull real imagery.
- Handle authentication (secrets) and caching.
- **Tests:** Mock HTTP requests or use small public AOI examples.

Phase 9 · Full Math Pipeline (Indices → Biomass)

- **Implement** in `mask.py`, `indices.py`, `model.py`:
- Cloud/water masking, FAI/NDRE formulas, RF inference.
- **Validation:** Compare outputs against known reference for a test AOI.
- **Tests:** Property-based tests (Hypothesis) for indices; regression tests for biomass values.

Phase 10 · Carbon Sequestration Reporting

- **Add** calculations to convert biomass to carbon (tonnes C, CO₂e).
 - **Expose** endpoints or UI charts showing cumulative sequestration for user-selected Kelp farms in BC.
 - **Data store:** Save results per AOI for historical tracking.
 - **Tests:** Validate unit conversions and summarizations.
-

5. Usage & AI Workflow

1. **Open the repo** in Cursor/Claude.
2. **Consult this guide** (`docs/agent-guide.md`) before each phase.
3. **Paste the structured prompts** provided in-guide.
4. **Verify tests pass** and CI is green after each commit.

This guide is the single source of truth for AI-driven development of `kelpie-carbon-v1`. It covers from bootstrapping to a production-ready carbon reporting web app for Kelp farms in BC.