Building Entrance Finder – One-Command App

This project serves a minimal web UI and API to:

- Geocode an address to its bounding box and center, and
- Estimate the most likely building entrance by projecting the nearest road point to the bbox edge,
- Show an interactive satellite map with the result.

It runs with a single command and auto-installs a local Node.js runtime and dependencies if needed. By default, it exposes a public URL so anyone on the internet can access it while it runs on your Mac.

One Command (macOS, Linux, Windows)

Run from the repo root:

- macOS/Linux: `./run`
- Windows (PowerShell or CMD): `.\run`

What the runner does:

- Ensures a local Node.js 20 runtime under `.runtime/` (downloaded if missing or system Node < 18).
- Installs Node dependencies into `node_modules/` (only if missing).
- Creates `.env` with defaults on first run and starts the server.
- Opens a public tunnel URL by default (Cloudflare Quick Tunnel). Set `ENABLE_TUNNEL=0` to disable.

Open the printed Public URL (or `http://localhost:8080/`) and use the web UI to enter an address and view the entrance map.

Security notes

- Defaults are conservative: only GET/HEAD are allowed cross-origin by default; body size limited to 10kb; rate limit of 60 requests/minute per IP.
- To allow broader access (e.g., POST), set `CORS_ORIGIN=*` in your environment.
- This is still an internet-exposed endpoint. Avoid adding stateful or file-system operations unless you fully validate and authenticate requests.
- Prefer running behind a random tunnel URL. Disable `ENABLE_TUNNEL` when not in use.

Configuration

Environment variables (can be placed in a `.env` file). See `env.sample` for a template.

- `PORT` (default `8080`) – port to listen on
- `HOST` (default `0.0.0.0`) – listen interface
- `CORS_ORIGIN` (default `GET_ONLY`) – `GET_ONLY`, `*`, or comma-separated allowed origins
- `RATE_LIMIT_WINDOW_MS` (default `60000`)
- `RATE_LIMIT_MAX` (default `60`)
- `BODY_LIMIT` (default `10kb`)
- `AUTH_TOKEN` – if set, require `Authorization: Bearer <token>` for all requests

- `TRUST_PROXY` (default `1`) – trust proxy hops for correct client IPs behind tunnels
- `ENABLE_TUNNEL` (default `1` via `./run`) – set to `0` to disable the public tunnel
- `TUNNEL_SUBDOMAIN` – optional preferred subdomain
- `GEOCODER_BASE_URL` – base URL for a Nominatim-compatible geocoder used by `/geocode/bbox` (default `https://nominatim.openstreetmap.org`)

API Overview

- `GET /entrance?q=<address>`: returns the geocoded center, bbox, a derived entrance candidate, and a link to an HTML map saved under `/sessions/<id>/map.html`.
- `GET /geocode/bbox?q=<address>`: returns center and bbox only.
- `GET /`: static web UI to query `/entrance` and preview the map.
- `GET /ping` or `/health`: health checks.

Example `/ping` response:

```
{
  "message": "ping",
  "method": "GET",
  "path": "/anything",
  "query": { "foo": "bar" },
  "headers": { "user-agent": "..." },
  "timestamp": "2025-01-01T00:00:00.000Z"
}
```

Scripts (optional)

If you want to manage Node yourself instead of using `./run`:

- `npm install` – install deps
- `npm start` – start server locally
- `npm run expose` – start and open a public tunnel (cross-platform)

Repo Layout

- `server.js`: Express API with `/ping`, `/geocode/bbox`, and `/entrance`.
- `web/index.html`: Minimal web UI for querying `/entrance` and previewing the session map.
- `run`: One-command bootstrapper and launcher.
- `src/satdist`: Python package for bbox distance, imagery fetch, and map helpers.
- Generated artifacts (e.g., `sessions/`, `config/cache/`, `*.png`) are ignored.

Geocoding Endpoint

- `GET /geocode/bbox?q=<address>`: Returns the latitude/longitude for the best match and the bounding box for the address.

Example:

```
# URL-encoded address (works in any shell)
curl "http://localhost:8080/geocode/bbox?q=1600%20Pennsylvania%20Ave%20NW%2C%20Washington%2C%20DC"

# or let curl encode it for you
curl -G --data-urlencode "q=1600 Pennsylvania Ave NW, Washington, DC" http://localhost:8080/geocode/bbox
```

Response:

```
{
  "query": "1600 Pennsylvania Ave NW, Washington, DC",
  "provider": "nominatim",
  "center": { "lat": 38.897675, "lon": -77.036547 },
  "bbox": { "south": 38.897, "west": -77.037, "north": 38.898, "east": -77.036 }
}
```

Notes:

- The default provider is OpenStreetMap Nominatim. You can point to your own Nominatim instance by setting `GEOCODER_BASE_URL`.
- If `AUTH_TOKEN` is set, include `Authorization: Bearer <token>` in requests.
- Default CORS allows only GET/HEAD from anywhere. This endpoint is GET-only and works with the default.
- Errors: `400` for missing/too-long query, `404` if not found, `502` if provider is unavailable, `504` for timeouts.

---

satdist: Satellite BBox Utilities

Production-grade utilities for working with satellite imagery of a geographic bounding box:

- Fetch static satellite images for a WGS84 bbox via ArcGIS endpoints.
- Compute distances between points using haversine or mapped image pixels.
- Generate an interactive Folium map with an on-map measurement tool.

Contents:

- src/satdist: Python package with distance, fetch, and map modules, plus a CLI.
- plot_bbox.py: Example that generates a map and saves a satellite image.
- compute_distance_example.py: Example CLI for quick distance calculations.
- config/cache: Default output directory for fetched images.

Quick start:

1. Install (dev):

   - pip install -e .[dev]
2. Build a map with measurement:
   - python -m satdist.cli map --south 47.6006395 --west -122.139536 --north 47.6008162 --east -122.1392861
   - Outputs building_bbox.html with an interactive measure tool.
3. Fetch a static image for the bbox:
   - python -m satdist.cli fetch --south 47.6006395 --west -122.139536 --north 47.6008162 --east -122.1392861 --width 1024 --height 1024
4. Compute distance:
   - Lat/Lon: python -m satdist.cli distance --lat1 47.6007247 --lon1 -122.139411 --lat2 47.6007247 --lon2 -122.139300
   - Pixels: python -m satdist.cli distance --south 47.6006395 --west -122.139536 --north 47.6008162 --east -122.1392861 --width 1024 --height 1024 --x1 512 --y1 512 --x2 522 --y2 512

Dev tools:

- Tests: pytest -q
- Lint: ruff check src tests
- Format: black src tests

Notes:

- External imagery Terms of Use apply to ArcGIS services. Intended for debug/development usage.
- The pixel-to-lat/lon mapping linearly interpolates across the bbox and is accurate for small areas.

Examples

Moved to `examples/`:

- `examples/plot_bbox.py` – builds an interactive map and saves HTML.
- `examples/fetch_satellite_bbox.py` – fetches and saves a satellite image for a bbox.
- `examples/compute_distance_example.py` – computes distances via pixels or lat/lon.

Run an example (from repo root):

```
python examples/plot_bbox.py
```