Modularization

Having python modules that can be shared across multiple HOT projects reduces long-term maintenance.

Why Use Modules ?

- Changes only need to be made in one place.
- Sharing code leads to faster software development.
- Easier to isolate portability issues.
- Build on the work of others.
- Much of the code is being pulled from existing projects.

The Projects

- osm-fieldwork
- osm-rawdata
- fmtm-splitter
- conflator
- tm-admin
- osm-login
- TM-Extractor

osm-fieldwork

- Convert files from OpenDataKit XML to OpenStreetMap OSM XML or GeoJson.
- Convert files from ODK Central JSON or CSV to OSM XML.
- Generate imagery basemaps for ODK Collect and Osmand from a variety of sources.
- Create OSM data extracts for ODK Collect.
- Uses a YAML or JSON file to define canned SQL queries for a local postgres database or the Underpass REST API.

osm-rawdata

- Access the raw database maintained for Export Tool for other projects. Export Tool, fAIr, Tasking Manager, and FMTM all need access to a postgres database to make high quality data extracts from the Underpass database which is updated every minute.
- Imports Parquet, OSM, and GeoJson into raw data database schema.
- Works with Underpass remotely or a local postgres.

fmtm-splitter

- Splits an AOI into tasks by squares.
- Support splitting by features in existing OSM map data, using highways, waterways, or railways.
- Returns a GeoJson file.

Conflator

- Conflate OSM data from ODK Collect with existing OSM data.
- Conflate buildings from external datasets with existing OSM data.
- Duplicate buildings.
- Overlapping buildings.
- Conflate highways from external datasets with existing OSM data. (TBD)

osm-login

- Use OAUTH 2.0 to login to front end
- Uses OSM account for token
- Used by TM, FMTM, and fAIr

TM Extractor

- Queries The Tasking Manager database
- Queries the raw-data REST API
- Downloads OSM data

tm-admin Functionality

- Management of user, project, organization, campaigns, and team profiles.
- Supports data exchange between other projects.
- Uses postgres directly instead of sqlalchemy.
- Contains much of the functionality required by any Tasking Manager style project.

tm-admin

Support for Tasking Manager style projects

- projects
- users
- organizations
- tasks
- teams
- campaigns
- messages
- notifications
- chat

Backend Support

- Designed to be part of a backend under FastAPI or Flask.
- Modules have layers of APIs built on top of each other.
- The APIs are all documented: https://docs.hotosm.org.
- Modified database schema for better performance.
- Fully async.

Data Exchange

- Uses gRPC for a communication layer below a REST API.
- gRPC uses Protobuf files to define data packets.
- A single config file generates the protobuf files.
- Python and SQL bindings are also generated.

The Config File

- users:
  - id:
    - int64
    - required: true
    - share: true
    - sequence: true
    - unique: true
  - username:
    - string
    - share: true
  - name:
    - string
    - required: true
    - share: true

SQLAlchemy vs Psycopg2

- SQLAlchemy is an API that works with multiple different databases.
- Psycopg2 only works with Postgres.
- Only Postgres has Postgis.
- SQLAlchemy has many layers of abstraction.
- Psycopg2 is simple, just SQL queries.
- The generated python bindings replace DTOs.