

SafePlaces API Architecture

- Overview
- Ingestion Service
 - Overview
 - Endpoints
 - Schema
- WebUI Service
 - Overview
 - Endpoints
 - Schema
- Syncing of Data Between Ingestion and WebUI databases
- Scope of Example Application

WORK IN PROGRESS

Overview

The goal of this document is to outline the services that comprise the SafePlaces backend.

Ingestion Service

Overview

The ingestion service will be a public facing API that the SafePaths mobile application interfaces with. It will be responsible for validating access codes, registering user's consent to HA's terms of service, and persisting location data users upload from the SafePaths mobile app.

Endpoints

- GET /access-code/valid
- POST /consent
- POST /upload

Schema

We need to support the private application having access to the following data:

- Read/write to `access_codes` table
- Read from `cases` table
- Write to `trails` table

WebUI Service

Overview

The WebUI service will be a private API that is utilized by the SafePlaces contact tracer workflow tool. It will be responsible for the following:

- User authentication
- Organization configuration
- Creation & deletion of cases
- Creation, editing, and deletion of points of concern data associated with a case
- Publishing an aggregated and anonymized case data

Endpoints

- POST /login
- GET /organization
- GET /organization/configuration
- POST /organization/configuration
- GET /organization/cases
- POST /organization/case
- DELETE /case
- GET /case/points
- POST /case/point

- POST /case/publish
- PUT /point
- DELETE /point

Schema

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 12.2
-- Dumped by pg_dump version 12.2

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

--
-- Name: postgis; Type: EXTENSION; Schema: -; Owner: -
--

CREATE EXTENSION IF NOT EXISTS postgis WITH SCHEMA public;

--
-- Name: EXTENSION postgis; Type: COMMENT; Schema: -; Owner:
--

COMMENT ON EXTENSION postgis IS 'PostGIS geometry, geography, and raster
spatial types and functions';

--
-- Name: state_type; Type: TYPE; Schema: public; Owner: safeplaces_dev_user
--

CREATE TYPE public.state_type AS ENUM (
    'unpublished',
    'staging',
    'published'
);

ALTER TYPE public.state_type OWNER TO safeplaces_dev_user;
```

```

--
-- Name: on_update_timestamp(); Type: FUNCTION; Schema: public; Owner:
safeplaces_dev_user
--

CREATE FUNCTION public.on_update_timestamp() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    NEW.updated_at = now();
    RETURN NEW;
END;
$$;

ALTER FUNCTION public.on_update_timestamp() OWNER TO safeplaces_dev_user;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
-- Name: cases; Type: TABLE; Schema: public; Owner: safeplaces_dev_user
--

CREATE TABLE public.cases (
    id uuid NOT NULL,
    organization_id uuid NOT NULL,
    publication_id integer,
    state public.state_type NOT NULL,
    created_at timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
    updated_at timestamp without time zone DEFAULT CURRENT_TIMESTAMP
);

ALTER TABLE public.cases OWNER TO safeplaces_dev_user;

--
-- Name: knex_migrations; Type: TABLE; Schema: public; Owner:
safeplaces_dev_user
--

CREATE TABLE public.knex_migrations (
    id integer NOT NULL,
    name character varying(255),
    batch integer,
    migration_time timestamp with time zone
);

```

```

ALTER TABLE public.knex_migrations OWNER TO safeplaces_dev_user;

--
-- Name: knex_migrations_id_seq; Type: SEQUENCE; Schema: public; Owner:
safeplaces_dev_user
--

CREATE SEQUENCE public.knex_migrations_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE public.knex_migrations_id_seq OWNER TO safeplaces_dev_user;

--
-- Name: knex_migrations_id_seq; Type: SEQUENCE OWNED BY; Schema: public;
Owner: safeplaces_dev_user
--

ALTER SEQUENCE public.knex_migrations_id_seq OWNED BY public.knex_migrations.
id;

--
-- Name: knex_migrations_lock; Type: TABLE; Schema: public; Owner:
safeplaces_dev_user
--

CREATE TABLE public.knex_migrations_lock (
    index integer NOT NULL,
    is_locked integer
);

ALTER TABLE public.knex_migrations_lock OWNER TO safeplaces_dev_user;

--
-- Name: knex_migrations_lock_index_seq; Type: SEQUENCE; Schema: public;
Owner: safeplaces_dev_user
--

CREATE SEQUENCE public.knex_migrations_lock_index_seq
    AS integer
    START WITH 1
    INCREMENT BY 1

```

```
NO MINVALUE
NO MAXVALUE
CACHE 1;
```

```
ALTER TABLE public.knex_migrations_lock_index_seq OWNER TO safeplaces_dev_user;
```

```
--
-- Name: knex_migrations_lock_index_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: safeplaces_dev_user
--
```

```
ALTER SEQUENCE public.knex_migrations_lock_index_seq OWNED BY public.
knex_migrations_lock.index;
```

```
--
-- Name: organizations; Type: TABLE; Schema: public; Owner: safeplaces_dev_user
--
```

```
CREATE TABLE public.organizations (
    id uuid NOT NULL,
    name character varying(255),
    updated_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP,
    created_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP
);
```

```
ALTER TABLE public.organizations OWNER TO safeplaces_dev_user;
```

```
--
-- Name: publications; Type: TABLE; Schema: public; Owner: safeplaces_dev_user
--
```

```
CREATE TABLE public.publications (
    id integer NOT NULL,
    organization_id uuid NOT NULL,
    start_date timestamp with time zone NOT NULL,
    end_date timestamp with time zone NOT NULL,
    publish_date timestamp with time zone NOT NULL,
    updated_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP,
    created_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP
);
```

```
ALTER TABLE public.publications OWNER TO safeplaces_dev_user;
```

```
--
-- Name: publications_id_seq; Type: SEQUENCE; Schema: public; Owner:
safeplaces_dev_user
```

```

--

CREATE SEQUENCE public.publications_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE public.publications_id_seq OWNER TO safeplaces_dev_user;

--
-- Name: publications_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner:
safeplaces_dev_user
--

ALTER SEQUENCE public.publications_id_seq OWNED BY public.publications.id;

--
-- Name: settings; Type: TABLE; Schema: public; Owner: safeplaces_dev_user
--

CREATE TABLE public.settings (
  id uuid NOT NULL,
  organization_id uuid NOT NULL,
  info_website_url character varying(255),
  reference_website_url character varying(255),
  api_endpoint_url character varying(255),
  region_coordinates json,
  notification_threshold_percent integer DEFAULT 66,
  notification_threshold_count integer DEFAULT 6,
  chunking_in_seconds integer DEFAULT 43200,
  days_to_retain_records integer DEFAULT 30,
  updated_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP,
  created_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP
);

ALTER TABLE public.settings OWNER TO safeplaces_dev_user;

--
-- Name: trails; Type: TABLE; Schema: public; Owner: safeplaces_dev_user
--

CREATE TABLE public.trails (
  id integer NOT NULL,
  case_id uuid NOT NULL,

```

```
coordinates public.geometry(Point,4326),
"time" timestamp with time zone,
hash character varying(255),
created_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP NOT NULL,
updated_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP NOT NULL
);
```

```
ALTER TABLE public.trails OWNER TO safeplaces_dev_user;
```

```
--
-- Name: trails_id_seq; Type: SEQUENCE; Schema: public; Owner:
safeplaces_dev_user
--
```

```
CREATE SEQUENCE public.trails_id_seq
AS integer
START WITH 1
INCREMENT BY 1
NO MINVALUE
NO MAXVALUE
CACHE 1;
```

```
ALTER TABLE public.trails_id_seq OWNER TO safeplaces_dev_user;
```

```
--
-- Name: trails_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner:
safeplaces_dev_user
--
```

```
ALTER SEQUENCE public.trails_id_seq OWNED BY public.trails.id;
```

```
--
-- Name: users; Type: TABLE; Schema: public; Owner: safeplaces_dev_user
--
```

```
CREATE TABLE public.users (
id uuid NOT NULL,
organization_id uuid NOT NULL,
username character varying(64),
email character varying(128),
password character varying(60),
maps_api_key character varying(255),
is_admin boolean,
updated_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP,
created_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP
);
```

```

ALTER TABLE public.users OWNER TO safeplaces_dev_user;

--
-- Name: knex_migrations id; Type: DEFAULT; Schema: public; Owner:
safeplaces_dev_user
--

ALTER TABLE ONLY public.knex_migrations ALTER COLUMN id SET DEFAULT nextval
('public.knex_migrations_id_seq'::regclass);

--
-- Name: knex_migrations_lock index; Type: DEFAULT; Schema: public; Owner:
safeplaces_dev_user
--

ALTER TABLE ONLY public.knex_migrations_lock ALTER COLUMN index SET DEFAULT
nextval('public.knex_migrations_lock_index_seq'::regclass);

--
-- Name: publications id; Type: DEFAULT; Schema: public; Owner:
safeplaces_dev_user
--

ALTER TABLE ONLY public.publications ALTER COLUMN id SET DEFAULT nextval
('public.publications_id_seq'::regclass);

--
-- Name: trails id; Type: DEFAULT; Schema: public; Owner: safeplaces_dev_user
--

ALTER TABLE ONLY public.trails ALTER COLUMN id SET DEFAULT nextval('public.
trails_id_seq'::regclass);

--
-- Name: cases cases_pkey; Type: CONSTRAINT; Schema: public; Owner:
safeplaces_dev_user
--

ALTER TABLE ONLY public.cases
    ADD CONSTRAINT cases_pkey PRIMARY KEY (id);

--
-- Name: knex_migrations_lock knex_migrations_lock_pkey; Type: CONSTRAINT;
Schema: public; Owner: safeplaces_dev_user

```



```

--

ALTER TABLE ONLY public.knex_migrations_lock
    ADD CONSTRAINT knex_migrations_lock_pkey PRIMARY KEY (index);

--
-- Name: knex_migrations knex_migrations_pkey; Type: CONSTRAINT; Schema:
public; Owner: safeplaces_dev_user
--

ALTER TABLE ONLY public.knex_migrations
    ADD CONSTRAINT knex_migrations_pkey PRIMARY KEY (id);

--
-- Name: organizations organizations_pkey; Type: CONSTRAINT; Schema: public;
Owner: safeplaces_dev_user
--

ALTER TABLE ONLY public.organizations
    ADD CONSTRAINT organizations_pkey PRIMARY KEY (id);

--
-- Name: publications publications_pkey; Type: CONSTRAINT; Schema: public;
Owner: safeplaces_dev_user
--

ALTER TABLE ONLY public.publications
    ADD CONSTRAINT publications_pkey PRIMARY KEY (id);

--
-- Name: settings settings_pkey; Type: CONSTRAINT; Schema: public; Owner:
safeplaces_dev_user
--

ALTER TABLE ONLY public.settings
    ADD CONSTRAINT settings_pkey PRIMARY KEY (id);

--
-- Name: trails trails_pkey; Type: CONSTRAINT; Schema: public; Owner:
safeplaces_dev_user
--

ALTER TABLE ONLY public.trails
    ADD CONSTRAINT trails_pkey PRIMARY KEY (id);

```

```

--
-- Name: users users_pkey; Type: CONSTRAINT; Schema: public; Owner:
safeplaces_dev_user
--

ALTER TABLE ONLY public.users
    ADD CONSTRAINT users_pkey PRIMARY KEY (id);

--
-- Name: cases cases_updated_at; Type: TRIGGER; Schema: public; Owner:
safeplaces_dev_user
--

CREATE TRIGGER cases_updated_at BEFORE UPDATE ON public.cases FOR EACH ROW
EXECUTE FUNCTION public.on_update_timestamp();

--
-- Name: organizations organizations_updated_at; Type: TRIGGER; Schema:
public; Owner: safeplaces_dev_user
--

CREATE TRIGGER organizations_updated_at BEFORE UPDATE ON public.organizations
FOR EACH ROW EXECUTE FUNCTION public.on_update_timestamp();

--
-- Name: publications publications_updated_at; Type: TRIGGER; Schema: public;
Owner: safeplaces_dev_user
--

CREATE TRIGGER publications_updated_at BEFORE UPDATE ON public.publications
FOR EACH ROW EXECUTE FUNCTION public.on_update_timestamp();

--
-- Name: settings settings_updated_at; Type: TRIGGER; Schema: public; Owner:
safeplaces_dev_user
--

CREATE TRIGGER settings_updated_at BEFORE UPDATE ON public.settings FOR EACH
ROW EXECUTE FUNCTION public.on_update_timestamp();

--
-- Name: trails trails_updated_at; Type: TRIGGER; Schema: public; Owner:
safeplaces_dev_user
--

```

```
CREATE TRIGGER trails_updated_at BEFORE UPDATE ON public.trails FOR EACH ROW
EXECUTE FUNCTION public.on_update_timestamp();
```

```
--
-- Name: users users_updated_at; Type: TRIGGER; Schema: public; Owner:
safeplaces_dev_user
--
```

```
CREATE TRIGGER users_updated_at BEFORE UPDATE ON public.users FOR EACH ROW
EXECUTE FUNCTION public.on_update_timestamp();
```

```
--
-- Name: cases cases_organization_id_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: safeplaces_dev_user
--
```

```
ALTER TABLE ONLY public.cases
    ADD CONSTRAINT cases_organization_id_fkey FOREIGN KEY (organization_id)
REFERENCES public.organizations(id) ON DELETE CASCADE;
```

```
--
-- Name: cases cases_publication_id_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: safeplaces_dev_user
--
```

```
ALTER TABLE ONLY public.cases
    ADD CONSTRAINT cases_publication_id_fkey FOREIGN KEY (publication_id)
REFERENCES public.publications(id) ON DELETE CASCADE;
```

```
--
-- Name: publications publications_organization_id_foreign; Type: FK
CONSTRAINT; Schema: public; Owner: safeplaces_dev_user
--
```

```
ALTER TABLE ONLY public.publications
    ADD CONSTRAINT publications_organization_id_foreign FOREIGN KEY
(organization_id) REFERENCES public.organizations(id) ON DELETE CASCADE;
```

```
--
-- Name: settings settings_organization_id_foreign; Type: FK CONSTRAINT;
Schema: public; Owner: safeplaces_dev_user
--
```

```
ALTER TABLE ONLY public.settings
```

```

    ADD CONSTRAINT settings_organization_id_foreign FOREIGN KEY
(organization_id) REFERENCES public.organizations(id) ON DELETE CASCADE;

--
-- Name: trails trails_case_id_foreign; Type: FK CONSTRAINT; Schema: public;
Owner: safeplaces_dev_user
--

ALTER TABLE ONLY public.trails
    ADD CONSTRAINT trails_case_id_foreign FOREIGN KEY (case_id) REFERENCES
public.cases(id) ON DELETE CASCADE;

--
-- Name: users users_organization_id_foreign; Type: FK CONSTRAINT; Schema:
public; Owner: safeplaces_dev_user
--

ALTER TABLE ONLY public.users
    ADD CONSTRAINT users_organization_id_foreign FOREIGN KEY (organization_id)
REFERENCES public.organizations(id) ON DELETE CASCADE;

--
-- PostgreSQL database dump complete
--

```

Syncing of Data Between Ingestion and WebUI databases

HA's will be responsible for syncing data between the two databases. This could be accomplished in a variety of ways but will be left up to the individual HA's to implement.

Scope of Example Application

The example applications will not concern itself with syncing of data between the individual Ingestion and WebUI service databases. For ease of testing and local development the example applications will be setup to point to the same local database.