Biosphere Capital

Right Plant Systems Management Guide

Prepared by

Catalyst.Net Ltd

April 2023

Commercial in confidence

catalyst

open source technologists

Level 6, Catalyst House, 150 Willis Street, Wellington 6011
PO Box 11053, Manners Street, Wellington 6142, New Zealand
+64 4 499 2267 // enquiries@catalyst.net.nz // www.catalyst.net.nz

Contents

Right Plant

Right Plant Right Place Right Time implementation using React and Django.

Initial Setup

Prerequisite Software

Ensure you have the following software tools installed to your local machine before continuing with the setup.

- Docker
- docker-compose
- Make

LINZ API Key

In order to recieve address data you will need to supply a LINZ API key. Such a key can be retrieved by signing up to https://data.linz.govt.nz/. Your key should be placed in the `.env` file as `LINZ_API_KEY`.

Required Layers

Download and extract the following geospatial layers to `backend/right_tree/api/data/resources`:

- Ecological Districts
- Greater Christchurch Area
- Fundamental Soil Layers New Zealand Soil Classification

Your filesystem structure should appear similar to the following:

```
backend/right_tree/api/data/resources/ecological_districts/
- DOC_EcologicalDistricts_2021_08_02.cpg
- DOC_EcologicalDistricts_2021_08_02.dbf
- DOC_EcologicalDistricts_2021_08_02.prj
- DOC_EcologicalDistricts_2021_08_02.sbn
- DOC_EcologicalDistricts_2021_08_02.sbx
- DOC_EcologicalDistricts_2021_08_02.shp
- DOC_EcologicalDistricts_2021_08_02.shp.xml
- DOC_EcologicalDistricts_2021_08_02.shx
```

```
backend/right_tree/api/data/resources/fundamental_soil_layers/
- fundamental-soil-layers-new-zealand-soil-classification.cpg
- fundamental-soil-layers-new-zealand-soil-classification.dbf
- fundamental-soil-layers-new-zealand-soil-classification.prj
- fundamental-soil-layers-new-zealand-soil-classification.shp
- fundamental-soil-layers-new-zealand-soil-classification.shx
- fundamental-soil-layers-new-zealand-soil-classification.xml
```

```
backend/right_tree/api/data/resources/chch_zone/
- Greater_Christchurch_Area.cpg
- Greater_Christchurch_Area.shp
- Greater_Christchurch_Area.dbf
- Greater_Christchurch_Area.shx
- Greater_Christchurch_Area.prj
```

Plant Data Spreadsheet

A valid plant data spreadsheet must be provided and placed at `backend/right_tree/api/data/resources/plant_data.xlsx` within the filesystem.

Data Ingest

WARNING: The supplied `Makefile` does not differentiate development and production environments. If configuring the application for production, it is recommended to rename `docker-compose.production.yaml` to `docker-compose.yaml`, overwriting the existing file.

To initialise the database, first run the Django migrations to create the appropriate tables.

```
$ make migrate
```

It is then safe to apply the Django fixtures and load the geospatial layers (ecological districts, region boundaries, etc.).

```
$ make ingest
```

For address searching, the LINZ 'NZ Street Address' layer must be loaded into a separate schema.

```
$ make ingest_linz
```

Finally, create a superuser from whom the admin interface can be accessed. The tool will prompt you to supply a username, email, and password interactively.

```
$ make superuser
```

Running the Application

Assuming the database has been configured in accordance to the Data Ingest section, you should be able to start the application with

```
$ make start
```

`CTRL + C` will escape the log output, but persist running containers. To stop all running containers, execute

```
$ make stop
```

The User Interface

The development web interface can be accessed at http://localhost. The production server is hosted at https://rightplant.biospherecapital.com/.

The application is contained within a single-page and offers a questionnaire to its users. To access the questionnaire, a user must provide a valid activation key which they can provide or purchase for a fee via a Stripe payment portal.

The user workflow is as follows:

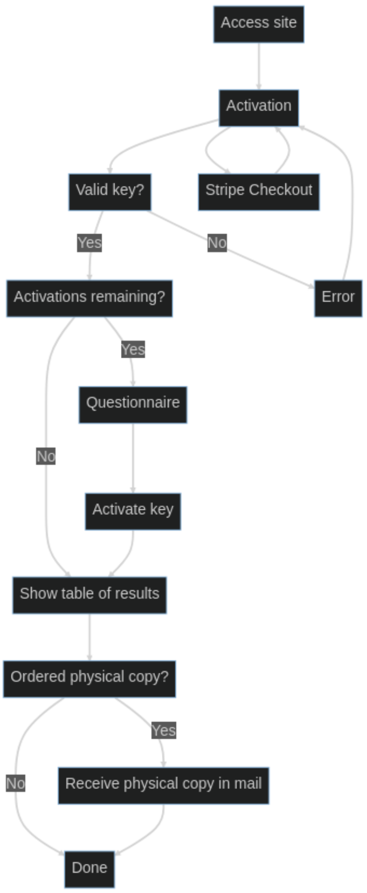

Figure 1: User workflow

The Admin Interface

The admin interface can be accessed from http://localhost/admin or https://rightplant.biospherecapital.com/admin.

There are several important sections within this interface that are especially important for retrieving and exporting completed questionnaires.

Model	Description
Activation key	A consumable activation key that can access the questionnaire a specific number of times
Activation key set	A named group of Activation key objects most useful for bulk exports. ‘Stripe - digital’ and ‘Stripe - physical’ are reserved names for digital and physical copy orders by individuals.
Customer	An identity tied to a physical copy orders
Customer address	A physical address of a Customer used for mailing physical copies. This is the address specified when a user fills out their shipping information from a Stripe checkout.
Questionnaire	A set of location and habitat information for a completed questionnaire, tied to an Activation key. These are filterable by their associated Activation key set in order to bulk export questionnaires and planting guides for only those whom have ordered physical copies.

Model	Description
Export	An export job of one or more questionnaires which can be downloaded as a ZIP archive of PDF documents containing the planting guides for each Questionnaire exported.

Maintenence

Renewing SSL Certificates

To generate new SSL certificates, run

```
$ make cert
```

You may be required to restart the running NGINX container for the change to take effect. You may do this by running

```
$ docker compose restart nginx
```

View Logs

To view and following logs of all running containers, run

```
$ make logs
```

To view individual container logs, you may interact with `docker-compose` directly. e.g.

```
$ docker compose logs [-f] <container-name>
```

Django Shell

WARNING: The following assumes you have an understanding of Python, Django, and its ORM. As these tools are powerful and potentially destructive, it is recommended to to use the Admin Interface, unless you know what you are doing.

To access an interactive Python shell on the Django project, run

```
$ make shell
```

PostgreSQL `psql` CLI Access

WARNING: The `psql` shell provided has superuser privileges and can make permanent changes to actively used production data if not handled properly.

To perform administrative tasks at the database-level, you may access an interactive superuser `psql` shell by running

```
$ make psql
```

Application Reset

WARNING: This is a destructive operation and will permanently remove all data in the running application. It is recommended that you backup any required questionnaires, keys, layers, etc. before continuing.

To reset your working environment, run

```
$ make reset
```

This will restore the repository to its initial state, from which you will need to follow the Initial Setup steps again in order to run the application again.

Running Configuration

Right Plant is currently live and operates with the following configuration.

Cloud Tenant

The Right Plant VM lives in the Hamilton (nz-hlz-1) region of Catalyst Cloud, with an IP address of 103.197.61.141.

It runs a minimal version of 20.04, which will be supported until April 2025. Patching the server is as simple as

```
# apt update
# apt dist-upgrade -y
```

and a system reboot, if required. It has not been configured for automatic upgrades.

SSH

SSH is available on port 22 for Catalyst management purposes, and 43212 for general user access. SSH Keys are required to log into the server, password authentication should not be enabled.

Code

The code and all configuration can be found on the VM in `/opt/rightplant`.

Code is deployed as a git repository - any changes made will be tracked by git. It is not possible to push changes to the Catalyst stored git repository.

Development

Backend

The backend code is stored under `backend/`. It uses

- Python 3.11
- Django web framework (version 3.2)
- Django REST framework for its API
- Pandas for CSV parsing
- pdfkit and PyPDF2 for generating planting guide PDFs
- Celery for async tasks like bulk questionnaire exports
- Redis
  - firstly as a message queue for Celery
  - secondly as a key-value store for mapping activation keys to Stripe checkout sessions
  - not used for database caching at all
- Gunicorn as a WSGI server
- PostGIS
  - `public.righttree` is used as the main Django database
  - `linz.nz_street_address` is a copy of the LINZ layer of the same name used for searching addressese

Frontend

The frontend code is stored under `frontend/`. It uses

- React 17
- React Router (was used in a past feature which has since been integrated into the single-page app)
- React Context for state management
- Material UI for components
- Axios for XHR requests
- Sass for styling
- Leaflet for map viewer
- Bootstrap/Reactstrap for styling conveniences (should double check if we need this)

Creating Habitat Images

1. Create png image from original svg with approprate crop.
2. Create zone polygons/rectangles on the original svg with divider lines anchor points as a guide
3. Copy zone polygons/rectangles to png image and size to fit (this is to ensure the only paths on the image the selectable ones)

4. Ensure all overlays have an almost transparent fill (lowest transparency value - in Inkscape this is 1) and no outline
5. Add a ‘label’ (not an id) to each overlay to match with a column name relating to the zone segment, this may be repeated. In Inkscape this is under ‘Object Properties’.
6. Save the png with overlays as an svg (it may either be inkscape or plain svg)
7. Place svg in relevant directory (`./frontend/src/assets/img/habitatSVG/`) in the frontend
8. Find and replace any instance of colons (:) in property names for the raw svg i.e. inkscape:label -> inkscapelabel. A helper script has been written to do this automatically please run `python process_svg.py`.