DELIVERABLE 4.1: System Architecture

Peter Finch, Energy Action Limited
Kyriakos Ktorides, ENERMAP

09/2017

This project has received funding from the European Union's Horizon 2020 programme under Grant Agreement No 695873

1. Introduction

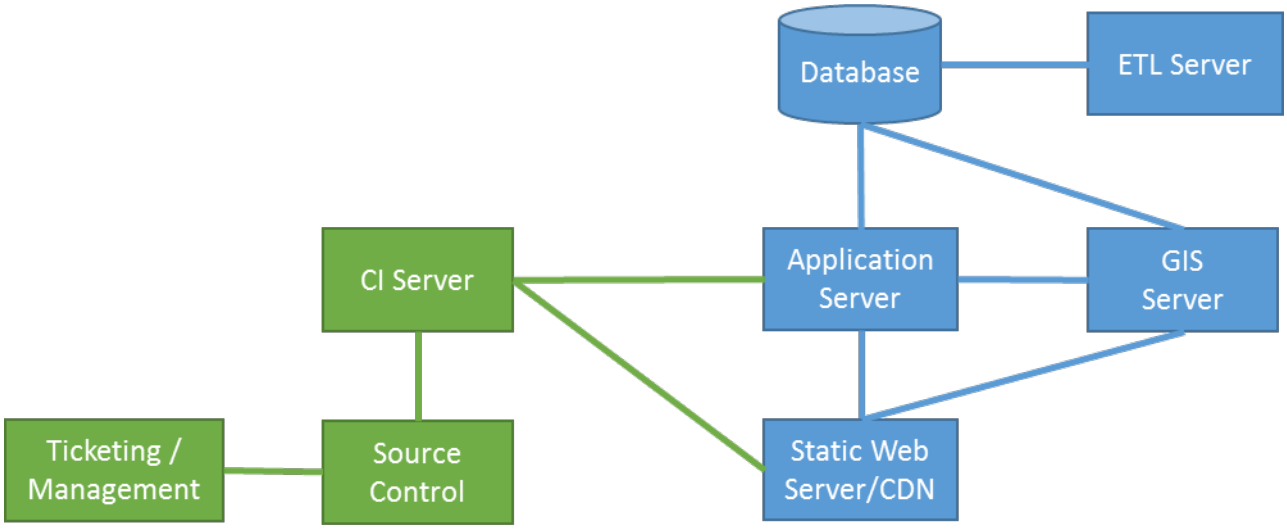
This document describes the architecture of the ENERFUND mapping application, describes the various components and appends some detail on how to set up the environment it is hosted on. It also provides details of the data design.

2. System Architecture

2.1. Overview

This section describes the system and data architecture of the application.

2.2. Database



The ENERFUND system is split into components responsible for different functions of the system. This provides separation for security, allows for the system to scale and allows for redundancy to provide high availability. During development, these components can be deployed on a single machine and can be split out across many machines as needed in production.

2.3. Components

The components in the diagram are split into system (blue) and management (green) components. The system components are responsible for hosting the ENERFUND application. The management components are responsible for hosting the source code, managing tasks and updating the system components from source. Each component will be described in more details below.

2.3.1. Database

The database will hold all variable data in the system, including EPC and geographical data.

2.3.2. ETL Server

The ETL server will be responsible for loading data into the database from various sources.

2.3.3. Application Server

The application server will host the server side / web service layer of the application.

2.3.4. GIS Server

The GIS server is responsible for load spatial data from many sourcing including the ENERFUND database as well as third party GIS Servers and exposing the data as OGC standard web services, in this case WMS and WFS. The ENERFUND application can perform CRUD (create, read, update and delete) operations on this data.

2.3.5. Static Web Server

The static web server will host the client side application. This can be separated from the application server to reduce the complexity and load on the application server.

2.3.6. Source Control

The source control component is responsible for hosting the source code for the project as well as tracking changes and versioning of the source. This can be cloud hosted, we are looking at using bitbucket https://bitbucket.org/ .

2.3.7. Ticketing/Management

The ticketing system allows the team to create and manage tasks related to the development of the system. These tasks can be managed in an agile way, and the ticketing system should allow for the creation of agile boards. We are currently looking at Jira

2.3.8. Continuous Integration (C.I.) Server

The continuous integration server is responsible for taking changes in source control, running unit tests and deploying to the system components. We are looking at Jenkins as our C.I. server at present.

2.4. Software Stack

The table below describes the software stack used in the various components in the system. The H/M column describes if the component is hosted by ENERFUND (H) or is an outsourced managed service (M).

Database	Linux, Postgresql > 9.6, PostGIS > 2.3.0	H/M
Application Server	Linux, Nginx, Kestrel, .NET core	H
Static Web Server / CDN	Amazon Cloudfront, Amazon S3	M
ETL Server	Linux, Python, OGC tools	H
GIS Server	Linux, Tomcat, Nginx, GeoServer	H
CI Server	Linux, Jenkins	H

Source control	Atlassian Bitbucket	M
Ticketing / Management	Atlassian Jira	M

2.5. Database Design

The ENERFUND database will store key contextual data used in the tool spanning many geographical regions. It will normalise this data to give a single standardised view to provide the tool with a consistent dataset. The database will be of a relational design and will make use of this design to provide a clean view of the data to the tool.

2.5.1. Datasets

Examples of key datasets that will be features will include the following information identified in the D3.2 Report – Features of the Tool.

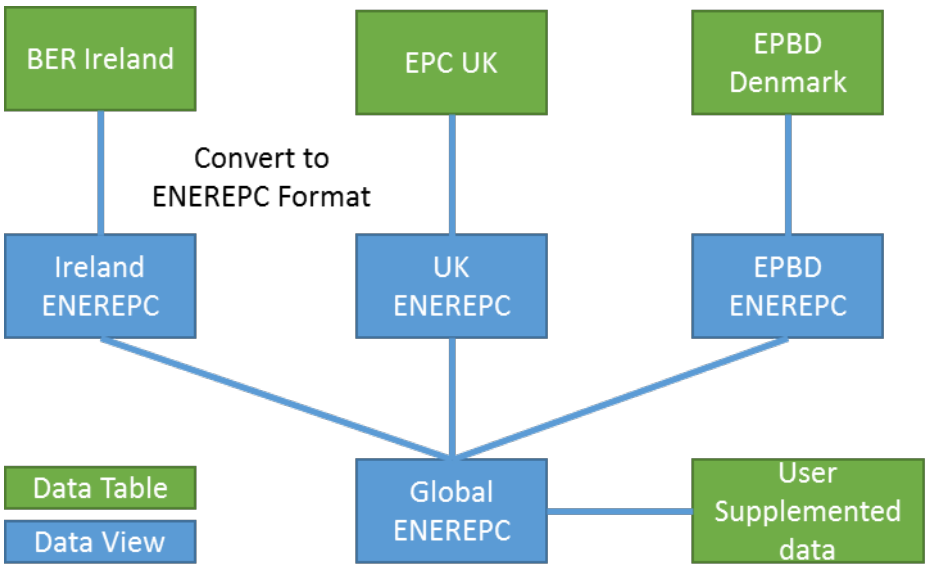
Category	Information Type
EPC/ Energy Use	EPC Rating - potential
	EPC Rating - current
	CO2 - potential
	CO2 - current
	Primary Energy/m2 (existing building)
	Primary Energy/m2 (post renovation)
	Floor area (m2)
	Thermal values (U values) - roof, wall, window
	Fuel types/ sources
	Heat Generator efficiency
	Share of renewable energy sources.
Building Info	Building type/ use
	Building ownership (public/ private)
Property Value/ Rents	Average sale price of buildings/ Property valuations
	Average rental rates

2.5.2. Data Normalising and Standardisation

Contextual data such as EPC will be stored in the database in a structure close to its source. This will reduce the amount of effort in loading the data into the database. Abstraction views on top of this data will transform it into a standardised format which we will call ENEREPC. This will allow the tool to query this data across the various contributing regions in ENERFUND in a consistent manor. The logic to transform this data will reside in the SQL (Structured Query Language) and stored procedures making up the ENEREPC views. A global ENEREPC view will combine all the regional views and user supplemented data into one seamless dataset.

A similar design will be applied to other datasets required by the tool such as energy prices.

Data relevant to a specific region can still be accessed by going directly to the local dataset. Attributes can be sparse in nature, meaning the tool should allow for gaps in the data.



2.6. Third Party Datasets

The tool will be configurable to load non-contextual third party external data in certain formats. These datasets can be overlaid on the map (screening) section of the tool. While the tool will not be able to use these datasets in calculations or analysis, they will provide valuable insight when targeting and screening energy retrofitting opportunities. These datasets will not be ingested by the ENERFUND system but will be loaded directly from the provider to the user's web browser and are therefore out of the scope of this design document.

2.7. Extract, Transform, Load (ETL)

The extract, transform, load process will be implemented by a collection of python scripts bespoke to each dataset. These scripts will validate the data and import them into the ENERFUND database. The data can be

3. Literature and Sources

https://www.postgresql.org/

http://geoserver.org/

https://www.debian.org/

https://www.microsoft.com/net/core#linuxdebian

https://www.atlassian.com/

https://jenkins.io/

https://aws.amazon.com/s3/

https://aws.amazon.com/cloudfront/

https://www.nginx.com/resources/wiki/

https://www.digitalocean.com/

https://asked.io/how-to-install-php-7-x--nginx-1-9-x---laravel-5-x

4. Technical Annex

4.1. Postgresql Configuration

4.1.1. Installing postgresql on the Sever

```
sudo apt-get update
sudo apt-get install postgresql postgresql-contrib
```

4.1.2. Setting Up User and Roles

```
createuser –interactive
```

4.1.3. Securing Postgresql

Stop remote access on postgresql:

```
sudo nano /etc/postgresql/9.1/main/pg_hba.conf
```

```
local   all             postgres                                peer
local   all             all                                     peer
host    all             all             127.0.0.1/32            md5
host    all             all             ::1/128                 md5
```

Create sensible roles and permissions

4.1.4. Install PostGIS

Psql

CREATE EXTENSION postgis;

4.1.5. Install ogr2ogr

```
sudo add-apt-repository ppa:ubuntugis/ppa
sudo apt-get update
```

```
sudo apt-get install gdal-bin
```

4.2. GeoServer Configuration

Geoserver configuration taken from:
docs.geoserver.org/stable/en/user/installation/linux.html

Make sure you have a Java Runtime Environment (JRE) installed on your system. GeoServer requires a Java 8 environment. The Oracle JRE is preferred, but OpenJDK has been known to work adequately. You can download JRE 8 from Oracle.

Note Java 9 is not currently supported.

Select the version of GeoServer that you wish to download. If you're not sure, select Stable.

Select Platform Independent Binary on the download page.

Download the archive and unpack to the directory where you would like the program to be located.

Note A suggested location would be /usr/share/geoserver.

Add an environment variable to save the location of GeoServer by typing the following command:

```
echo "export GEOSERVER_HOME=/usr/share/geoserver" >> ~/.profile
. ~/.profile
```

Make yourself the owner of the geoserver folder. Type the following command in the terminal window, replacing USER_NAME with your own username :

```
sudo chown -R USER_NAME /usr/share/geoserver/
```

Start GeoServer by changing into the directory geoserver/bin and executing the startup.sh script:

```
cd geoserver/bin
sh startup.sh
```

4.2.1. Setup Building Layer

- Setup a new PostGIS Datastore in Geoserver using credentials for the PostgreSQL server setup in 4.1.
- Setup a new layer on Geoserver using the Datastore as source and with an SQL View using the following query:

```
SELECT location, rating, potential_rating FROM buildings WHERE
location IS NOT NULL AND rating_num BETWEEN %minrating% AND
%maxrating% AND (%potential% = 0 OR potential_rating_num &lt;=
%potential%) AND ((%areamin% = 0 AND %areamax% = 0) OR
total_floor_area BETWEEN %areamin% AND %areamax%) AND
((%wallmin% = 0 AND %wallmax% = 0) OR walls_energy_efficiency
BETWEEN %wallmin% AND %wallmax%) AND ((%roofmin% = 0 AND
%roofmax% = 0) OR roof_energy_efficiency BETWEEN %roofmin% AND
```

```
%roofmax%) AND ((%winmin% = 0 AND %winmax% = 0) OR
windows_energy_efficiency BETWEEN %winmin% AND %winmax%) AND
((%heatmin% = 0 AND %heatmax% = 0) OR
main_heat_energy_efficiency BETWEEN %heatmin% AND %heatmax%);
```

- Add visual style to layer

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor version="1.0.0"
                       xsi:schemaLocation="http://www.opengis.net/sld
http://schemas.opengis.net/sld/1.0.0/StyledLayerDescriptor.xsd"
                       xmlns="http://www.opengis.net/sld"
xmlns:ogc="http://www.opengis.net/ogc"
                       xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <NamedLayer>
        <Name></Name>
        <UserStyle>
            <Title>Enerfund point style</Title>
            <FeatureTypeStyle>
                <Rule>
                    <ogc:Filter>
                        <ogc:PropertyIsEqualTo>
                            <ogc:PropertyName>rating</ogc:PropertyName>
                            <ogc:Literal>A</ogc:Literal>
                        </ogc:PropertyIsEqualTo>
                    </ogc:Filter>

                    <PointSymbolizer>
                        <Graphic>
                            <Mark>
                                <WellKnownName>circle</WellKnownName>
                                <Fill>
                                    <CssParameter
name="fill">#005100</CssParameter>
                                    <CssParameter name="fill-
opacity">0.8</CssParameter>
                                </Fill>
                            </Mark>
                            <Size>7</Size>
                        </Graphic>
                    </PointSymbolizer>
                </Rule>

                <Rule>
                    <ogc:Filter>
                        <ogc:PropertyIsEqualTo>
                            <ogc:PropertyName>rating</ogc:PropertyName>
                            <ogc:Literal>B</ogc:Literal>
                        </ogc:PropertyIsEqualTo>
                    </ogc:Filter>

                    <PointSymbolizer>
                        <Graphic>
                            <Mark>
                                <WellKnownName>circle</WellKnownName>
                                <Fill>
                                    <CssParameter
name="fill">#037f03</CssParameter>
                                    <CssParameter name="fill-
opacity">0.8</CssParameter>
                                </Fill>
```

```
                            </Mark>
                            <Size>7</Size>
                        </Graphic>
                    </PointSymbolizer>
                </Rule>
                <Rule>
                    <ogc:Filter>
                        <ogc:PropertyIsEqualTo>
                            <ogc:PropertyName>rating</ogc:PropertyName>
                            <ogc:Literal>C</ogc:Literal>
                        </ogc:PropertyIsEqualTo>
                    </ogc:Filter>

                    <PointSymbolizer>
                        <Graphic>
                            <Mark>
                                <WellKnownName>circle</WellKnownName>
                                <Fill>
                                    <CssParameter
name="fill">#85b602</CssParameter>
                                    <CssParameter name="fill-
opacity">0.8</CssParameter>
                                </Fill>
                            </Mark>
                            <Size>7</Size>
                        </Graphic>
                    </PointSymbolizer>
                </Rule>
                <Rule>
                    <ogc:Filter>
                        <ogc:PropertyIsEqualTo>
                            <ogc:PropertyName>rating</ogc:PropertyName>
                            <ogc:Literal>D</ogc:Literal>
                        </ogc:PropertyIsEqualTo>
                    </ogc:Filter>

                    <PointSymbolizer>
                        <Graphic>
                            <Mark>
                                <WellKnownName>circle</WellKnownName>
                                <Fill>
                                    <CssParameter
name="fill">#fffe00</CssParameter>
                                    <CssParameter name="fill-
opacity">0.8</CssParameter>
                                </Fill>
                            </Mark>
                            <Size>7</Size>
                        </Graphic>
                    </PointSymbolizer>
                </Rule>
                <Rule>
                    <ogc:Filter>
                        <ogc:PropertyIsEqualTo>
                            <ogc:PropertyName>rating</ogc:PropertyName>
                            <ogc:Literal>E</ogc:Literal>
                        </ogc:PropertyIsEqualTo>
                    </ogc:Filter>

                    <PointSymbolizer>
                        <Graphic>
                            <Mark>
                                <WellKnownName>circle</WellKnownName>
                                <Fill>
                                    <CssParameter
name="fill">#ff9600</CssParameter>
                                    <CssParameter name="fill-
opacity">0.8</CssParameter>
```

```
                                </Fill>
                            </Mark>
                            <Size>7</Size>
                        </Graphic>
                    </PointSymbolizer>
                </Rule>
                <Rule>
                    <ogc:Filter>
                        <ogc:PropertyIsEqualTo>
                            <ogc:PropertyName>rating</ogc:PropertyName>
                            <ogc:Literal>F</ogc:Literal>
                        </ogc:PropertyIsEqualTo>
                    </ogc:Filter>

                    <PointSymbolizer>
                        <Graphic>
                            <Mark>
                                <WellKnownName>circle</WellKnownName>
                                <Fill>
                                    <CssParameter
name="fill">#ff5000</CssParameter>
                                    <CssParameter name="fill-
opacity">0.8</CssParameter>
                                </Fill>
                            </Mark>
                            <Size>7</Size>
                        </Graphic>
                    </PointSymbolizer>
                </Rule>
                <Rule>
                    <ogc:Filter>
                        <ogc:PropertyIsEqualTo>
                            <ogc:PropertyName>rating</ogc:PropertyName>
                            <ogc:Literal>G</ogc:Literal>
                        </ogc:PropertyIsEqualTo>
                    </ogc:Filter>

                    <PointSymbolizer>
                        <Graphic>
                            <Mark>
                                <WellKnownName>circle</WellKnownName>
                                <Fill>
                                    <CssParameter
name="fill">#ff0000</CssParameter>
                                    <CssParameter name="fill-
opacity">0.8</CssParameter>
                                </Fill>
                            </Mark>
                            <Size>7</Size>
                        </Graphic>
                    </PointSymbolizer>
                </Rule>
            </FeatureTypeStyle>
        </UserStyle>
    </NamedLayer>
</StyledLayerDescriptor>
```

4.2.2. Configure Geoserver

- Enable Web Map Service
- Enable Tile Caching for new layer

4.3. Application Server Configuration

Following https://asked.io/how-to-install-php-7-x--nginx-1-9-x---laravel-5-x

The server is running PHP and Laravel. The installation instructions are as follows:

4.3.1. Add ondrej repo and Install

```
sudo apt-get install -y python-software-properties
sudo add-apt-repository -y ppa:ondrej/php
sudo apt-get update -y
sudo apt-get install nginx php7.1-cli php7.1-fpm php7.1-
mbstring php7.1-curl php7.1-pgsql
```

4.3.2. Configure NGINX

```
mv /etc/nginx/conf.d/default.conf /etc/nginx/default.conf-bkup
cat <<EOF > /etc/nginx/sites-enabled/app.enefund.eu.conf

server {
     listen 80;
     listen [::]:80;
     root /var/www/app/public/current/public;
        index index.php index.html index.htm;

    server_name localhost;
    charset   utf-8;

    gzip on;
    gzip_vary on;
    gzip_disable "msie6";
    gzip_comp_level 6;
    gzip_min_length 1100;
    gzip_buffers 16 8k;
    gzip_proxied any;
    gzip_types
```

```
        text/plain
        text/css
        text/js
        text/xml
        text/javascript
        application/javascript
        application/x-javascript
        application/json
        application/xml
        application/xml+rss;

    server_name app.enerfund.eu;

  location / {
     try_files $uri $uri/ /index.php$is_args$args;
  }

  location ~ \.php$ {
     try_files $uri =404;
     fastcgi_split_path_info ^(.+\.php)(/.+)$;
     fastcgi_pass unix:/var/run/php/php7.1-fpm.sock;
     fastcgi_index index.php;
     fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
     include fastcgi_params;
    }
}
EOF
```

4.3.3. Configure PHP 7

For FPM we need to fix_pathinfo.

```
echo 'cgi.fix_pathinfo=0' >> /etc/php/7.1/fpm/php.ini
```

- Increase memory limit, max_execution time & enable short tags:

  Edit /etc/php/7.1/fpm/php.ini and change the following:

  ```
  max_execution_time = 60
  memory_limit = 256MB
  short_open_tag = On
  ```

- Install Composer (PHP Package Manager)

  ```
  sudo apt-get install composer
  ```

- Install NodeJS and NPM

  ```
  curl -sL https://deb.nodesource.com/setup_7.x | sudo -E bash -
  sudo apt-get install -y nodejs
  ```

- Install Yarn

  ```
  curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-
  key add -
  echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo
  tee /etc/apt/sources.list.d/yarn.list
  sudo apt-get update && sudo apt-get install yarn
  ```

- Restart Services

  ```
  sudo service nginx restart
  ```

```
/etc/init.d/php7.1-fpm restart
```

4.3.4. Setup Deployment Process

Deployer is used for building and deploying the application to the server.

- Install Deployer on source machine

```
composer install -g deployer/deployer
```

- Add SSH key to server

```
cat ~/.ssh/idserver.pub | ssh enerfund@enerfund.eu 'cat >>
.ssh/authorized_keys'
```

- Setup private key for repository access on application server

```
ssh-keygen -t rsa -b 4096
```

(press enter 4 times)

```
eval $(ssh-agent -s)
```

`ssh-add ~/.ssh/id_rsa`

- Setup public key on Bitbucket

`cat ~/.ssh/id_rsa.pub`

copy the key and paste in Bitbucket at
`https://bitbucket.org/enerfund/enerfundapp/admin/access-keys/`

The following deployer script is used (deploy.php), execute by running: `dep deploy`

```
<?php

namespace Deployer;

require_once __DIR__ . '/vendor/deployer/deployer/recipe/common.php';
set('ssh_type', 'native');
set('ssh_multiplexing', true);
set('repository', 'git@bitbucket.org:enerfund/enerfundapp.git');


host(HOST_NAME)
    ->user('enerfund')
    ->port(22)
    ->identityFile('~/.ssh/idserver')
    ->multiplexing(false)
    ->set('deploy_path', '/var/www/app/public/');

// Laravel shared dirs
set('shared_dirs', [
    'storage',
]);

// Laravel shared file
set('shared_files', [
    '.env',
]);

// Laravel writable dirs
set('writable_dirs', [
    'bootstrap/cache',
    'storage',
    'storage/app',
    'storage/app/public',
    'storage/framework',
    'storage/framework/cache',
    'storage/framework/sessions',
    'storage/framework/views',
    'storage/logs',
]);

/**
 * Helper tasks
 */
desc('Execute artisan migrate:status');
task('artisan:migrate:status', function () {
    $output = run('{{bin/php}} {{release_path}}/artisan migrate:status');
    writeln('<info>' . $output . '</info>');
});

desc('Execute artisan db:seed');
task('artisan:db:seed', function () {
    $output = run('{{bin/php}} {{release_path}}/artisan db:seed --force');
    writeln('<info>' . $output . '</info>');
});
```

```
desc('Execute artisan cache:clear');
task('artisan:cache:clear', function () {
    run('{{bin/php}} {{release_path}}/artisan cache:clear');
});

desc('Execute artisan config:cache');
task('artisan:config:cache', function () {
    run('{{bin/php}} {{release_path}}/artisan config:cache');
});

desc('Execute artisan route:cache');
task('artisan:route:cache', function () {
    run('{{bin/php}} {{release_path}}/artisan route:cache');
});

desc('Execute artisan view:clear');
task('artisan:view:clear', function () {
    run('{{bin/php}} {{release_path}}/artisan view:clear');
});

desc('Execute artisan optimize');
task('artisan:optimize', function () {
    run('{{bin/php}} {{release_path}}/artisan optimize');
});

desc('Execute yarn install');
task('npm:install', function () {
    run('cd {{release_path}} && yarn install');
});

desc('Build front end scripts/css');
task('npm:production', function () {
    run('cd {{release_path}} && npm run production');
});

desc('Execute artisan storage:link');
task('artisan:storage:link', function () {
    run('{{bin/php}} {{release_path}}/artisan storage:link');
});

/**
 * Main task
 */
desc('Deploy Enerfund App');
task('deploy', [
    'deploy:prepare',
    'deploy:lock',
    'deploy:release',
    'deploy:update_code',
    'deploy:shared',
    'deploy:vendors',
    'deploy:writable',
    'artisan:storage:link',
    'artisan:view:clear',
    'artisan:cache:clear',
    'artisan:config:cache',
    'artisan:optimize',
    'npm:install',
    'npm:production',
    'deploy:symlink',
    'deploy:unlock',
    'cleanup',
]);

after('deploy', 'success');
```

4.4. Database Structure

4.4.1. How to Add a New Country

1. Add an entry for the country in the country table (columns explained below). Set active as 0 for now.
2. Add country boundaries in the boundaries table. Make sure admin_level is set correctly. Boundary data in GeoJSON format for all countries can be found here: https://mapzen.com/data/borders/
   Import using ogr2ogr tool, sample command:

   ogr2ogr -skipfailures -f "PostgreSQL" PG:"host=127.0.0.1 dbname=enerfund_db user=enerfund password=thepassword" "file.json" -nln boundaries -select name,admin_level,place

3. Setup entries for each administrative level in table admin_levels. Admin level names for each country can be found at the following URL: http://wiki.openstreetmap.org/wiki/Tag:boundary%3Dadministrative#10_admin_level_values_for_specific_countries
   minzoom/maxzoom are the minimum and maximum zoom levels on the map where the boundaries will be visible, the values are inclusive. They define a range of map zoom values. *Maxzoom* should have a max value of 16 cause at that zoom level the markers become visible on the map. The level value should match the *admin_level* column from table boundaries.
4. Set active in countries table to 1.
5. Country should be visible and functioning in the app.

4.4.2. How to Add a Filter

The site's top bar is generated based on information in the tables filter_categories and filters. Each dropdown is represented by a row in the filter_categories table and each filter is represented by a row in the filters table. You can find the table definitions below. These are the contents of those tables now:

id	title	seq
1	Efficiency	2
2	Construction	3
3	Rating	1

id	title	value_field	field_class	seq	direct	filter_category_id
1	Rent	monthly_rent	Rent	2	false	NULL
6	Windows Energy Efficiency	windows_energy_efficiency	efficiency	6	true	1
5	Roof Energy Efficiency	roof_energy_efficiency	efficiency	5	true	1
4	Walls Energy Efficiency	walls_energy_efficiency	efficiency	4	true	1
2	EnergyRating	rating	EnergyRating	1	false	3
3	Area (mÂ²)	total_floor_area	slider	3	true	2
7	Main Heat Energy Efficiency	main_heat_energy_efficiency	efficiency	7	true	1

Enabled field classes are "efficiency" "EnergyRating" and "slider".

1. **Efficiency** is a slider with values 1-5 mapping on efficiency columns in the buildings table and is presented with the values of Very Poor, Poor, Average, Good, Very Good.
2. **slider** can take any numerical column (it will calculate max and min values automatically) and is presented as a horizontal slider.

3. **EnergyRating** is a special case for Energy Rating and Potential Rating.

4.4.3. Tables

4.4.3.1. Countries

Column	Notes
id	Autogenerated ID
title	The name of the country as shown in select dropdowns
code	The two letter ISO country code
active	Boolean, set to 0 will remove the country. Useful while setting up a new country.
bb_min	Country bounding box south west point (postgis geography point type)
bb_max	Country bounding box north east point (postgis geography point type)
created_at	Time entry created
updated_at	Time last updated

4.4.3.2. Admin_Levels

Column	Notes
id	Autogenerated ID
country_id	Foreign key from countries table.
minzoom	The minimum map zoom level where this administrative level's boundary will be visible.
maxzoom	The maximum map zoom level where this administrative level's boundary will be visible.
level	The level number to map in the boundaries table.
name	Administrative level name, can be found in Open street map wiki.
created_at	Time entry created
updated_at	Time last updated

4.4.3.3. Boundaries

Column	Notes
ogc_fid	Autogenerated ID

name	Name of region
admin_level	Number of administrative level
place	
wkb_geometry	The geometry describing the region boundary.
country_id	The foreign key from countries table.
boundingbox	A pre-calculated bounding box for wkb_geometry. This is sent out with the geometry so the app's front end can filter boundaries faster.

4.4.3.4. Buildings

Column	Type	Notes
id	Integer	Autogenerated ID
local_authority	Varchar	Name of local authority, not currently used so it can be skipped if not available
post_code	Varchar	
address	Varchar	
property_type	Varchar	
rating	Char(1)	Rating as text (A-G)
potential_rating	Char(1)	Potential rating as text (A-G)
hash	Varchar	Unique value, in UK data a hash is provided as a unique key since their reference numbers are not guaranteed to be unique. Maybe if something similar is not available in other countries we can generate one based on some characteristics.

location	Geography	The point on the map
created_at	Timestamp	
updated_at	Timestamp	
total_floor_area	Integer	Area in sq metres.
rooms	smallint	
windows_energy_efficiency	Smallint	From 1 to 5 representing Very Poor, Poor, Average, Good, Very Good
walls_energy_efficiency	Smallint	From 1 to 5 representing Very Poor, Poor, Average, Good, Very Good
roof_energy_efficiency	Smallint	From 1 to 5 representing Very Poor, Poor, Average, Good, Very Good
main_head_energy_efficiency	Smallint	From 1 to 5 representing Very Poor, Poor, Average, Good, Very Good
co2_current	Double	
co2_potential	Double	
rating_num	Smallint	Rating in number form (1 = A, 2 = B etc). Both num fields are needed for geoserver's viewparam based query.

potential_rating_num	Smallint	Potential rating in number form (1 = A, 2 = B etc)
country_id	Integer	Foreign key from countries table

4.4.3.5. *Filter_Categories*

Column	Notes
id	Autogenerated ID
title	Category name
seq	Sequence number defining the order categories show up.

4.4.3.6. *Filters*

Column	Notes
Id	Autogenerated ID
Title	Filter title
Value_field	The column from boundaries table to be filtered
Field_class	Class type (see How to add filter)
Seq	Sequence number defining the order filters be displayed in the category.
Direct	Ignore for now
Filter_category_id	Foreign key from filter_categories (the category the filter will be displayed in)

4.5. Database Scripts

```
create table buildings
(
        id serial not null
                constraint buildings_pkey
```

```
                primary key,
        local_authority varchar(255) not null,
        post_code varchar(32) not null,
        address varchar(255) not null,
        property_type varchar(48) not null,
        rating varchar(1) not null,
        potential_rating varchar(1) not null,
        hash varchar(68) not null
            constraint buildings_hash_unique
                unique,
        location geography(Point,4326),
        created_at timestamp(0),
        updated_at timestamp(0),
        total_floor_area integer,
        rooms smallint,
        windows_energy_efficiency smallint,
        walls_energy_efficiency smallint,
        roof_energy_efficiency smallint,
        main_heat_energy_efficiency smallint,
        co2_current double precision,
        co2_potential double precision,
        rating_num smallint,
        potential_rating_num smallint,
        country_id integer
)
;

create index location_speedup_index
        on buildings (location)
;
```

```
create index buildings_total_floor_area_index
        on buildings (total_floor_area)
;

create index buildings_rooms_index
        on buildings (rooms)
;

create index buildings_main_heat_energy_efficiency_index
        on buildings (main_heat_energy_efficiency)
;

create index buildings_rating_num_index
        on buildings (rating_num)
;

create index buildings_potential_rating_num_index
        on buildings (potential_rating_num)
;

create table geocoder_log
(
        id serial not null
                constraint geocoder_log_pkey
                        primary key,
        building_id integer not null
                constraint geocoder_log_building_id_foreign
                        references buildings,
        geocoded boolean not null,
        geocoder varchar(255),
        created_at timestamp(0),
```

```
        updated_at timestamp(0)
)
;

create table cache
(
        key varchar(255) not null
                constraint cache_key_unique
                        unique,
        value text not null,
        expiration integer not null
)
;

create table geocoder_runs
(
        id serial not null
                constraint geocoder_runs_pkey
                        primary key,
        geocoder varchar(64) not null,
        action varchar(128) not null,
        created_at timestamp(0),
        updated_at timestamp(0)
)
;

create table filters
(
        id serial not null
                constraint filters_pkey
                        primary key,
```

```
        title varchar(64) not null,
        value_field varchar(32) not null,
        field_class varchar(64),
        seq integer default 0,
        direct boolean default false not null,
        filter_category_id integer
)
;

create unique index filters_seq_uindex
        on filters (seq)
;


create table filter_categories
(
        id serial not null
                constraint filter_categories_pkey
                        primary key,
        title varchar(255) not null,
        seq integer not null
)
;

create table countries
(
        id serial not null
                constraint countries_pkey
                        primary key,
        title varchar(255) not null,
        code varchar(2) not null,
```

```
        active boolean default false not null,
        bb_min geography(Point,4326) default NULL::geography,
        created_at timestamp(0),
        updated_at timestamp(0),
        bb_max geography(Point,4326) default NULL::geography
)
;

create table boundaries
(
        ogc_fid serial not null
                constraint boundaries_uk_pkey
                        primary key,
        name varchar,
        admin_level varchar,
        place varchar,
        wkb_geometry geometry,
        country_id integer default 0,
        boundingbox geometry
)
;

create index boundaries_uk_wkb_geometry_geom_idx
        on boundaries (wkb_geometry)
;

create table admin_levels
(
        id serial not null
                constraint admin_levels_pkey
                        primary key,
```

```
    country_id integer not null,
    minzoom integer default 0 not null,
    maxzoom integer default 0 not null,
    level integer default 0 not null,
    name varchar(255) not null,
    created_at timestamp(0),
    updated_at timestamp(0)
)
;
```

PROJECT DETAILS:

Website: www.enerfund.eu

Twitter: @enerfund

Facebook: /enerfund