# D5.2 - MANIFESTS DSS - Installation guides

COPtool

COP Viewer

Web application – Model interface

Silvia Allen-Perkins, Garbiñe Ayensa, Alberto Gómez, Pedro Montero, Samuël Orsi

## ACKNOWLEDGEMENT

The work described in this report was supported by the Directorate-General for European Civil Protection and Humanitarian Aid Operations (DG-ECHO) of the European Union through the Grant Agreement number 101004912 - MANIFESTS – UCPM-2020-PP-AG, corresponding to the Call objective "Enhancing prevention and protection from the effects of maritime disasters" under priority 1: "Developing response capacity for marine pollution".

# COPtool

COPtool INSTALLATION GUIDE

Pedro Montero, Garbiñe Ayensa, Silvia Allen-Perkins, Alberto Gómez

Co-funded by the European Union Civil Protection

## ACKNOWLEDGEMENT

The work described in this report was supported by the Directorate-General for European Civil Protection and Humanitarian Aid Operations (DG-ECHO) of the European Union through the Grant Agreement number 101004912 - MANIFESTS – UCPM-2020-PP-AG, corresponding to the Call objective “Enhancing prevention and protection from the effects of maritime disasters” under priority 1: “Developing response capacity for marine pollution”.

Co-funded by the European Union Civil
Protection

| | |
|---|---|
| Project Acronym | **MANIFESTS** |
| Project Full Title | **MAN**aging risks and **I**mpacts **F**rom **E**vaporating and gaseous **S**ubstances **T**o population **S**afety |
| Gant Agreement Nr. | **101004912** |
| Project Website | [https://www.manifests-project.eu/](https://www.manifests-project.eu/) |

| | |
|---|---|
| Deliverable Nr. | **D5.2 MANIFESTS DSS Installation guide** |
| Status (Final/Draft/Revised) | **Final** |
| Work Package | **5** |
| Task Number | **5.4** |
| Responsible Institute | **INTECMAR** |
| Author/s | **P- Montero, G. Ayensa, S. Allen-Perkins, A. Gómez** |
| Recommended Citation | |
| Dissemination Level | |

| Document History | | | |
|---|---|---|---|
| Version | Date | Modification Introduced | |
| | | Modification Reason | Modified by |
| 1.0 | 31/01/2023 | | |
| | | | |
| | | | |

Co-funded by the European Union Civil Protection

## Content

# 1. Background

In the event of a maritime accident involving HNS, maritime authorities must take numerous decisions to organize the best response strategy, i.e., one that minimizes risks to human health (including incident response teams, crew members and coastal communities), to the marine environment, for maritime safety and for socio-economic activities and facilities. While many key decisions and considerations are prescribed in national or regional contingency plans, operational response activities will generally need ongoing adjustment or review to reflect the most recent information available as the contamination event evolves. In such a rapidly changing situation, an efficient exchange of information between competent decision-making authorities and response teams on the ground can greatly facilitate both decision-making processes and organizational processes.

The purpose of work package 5 of the Manifests project (Manifests decision support system) is to develop an efficient information system that helps (1) decision makers understand the situation at stake and its likely evolution in the coming hours and days; (2) identify the population, ecosystems and socio-economic assets at risk and (3) share useful information with response teams deployed at sea, in the air or on the coast.

Building on the experience gained and development carried out during the previous HNS-MS and MARINER projects, the MANIFESTS decision support system (DSS) will integrate several services, including the DSS Common Operational Picture (COPtool).

This COPtool refers to a system designed so that during a contingency, the exchange of information that occurs between the maritime authorities and the different response teams (sea, coast, air) is carried out in the most efficient way possible, ensuring that all actors involved in the crisis committee and response teams can access the same data. These can be standard reports (such as the Standard Pollution Observation Report of the Bonn Agreement), images, videos and any other georeferenced data collected by response teams, as well as satellite observations, model simulation results, exclusion areas, location of response media, requests for new response actions shared by the crisis.

The COPtool documentation is composed by:

Co-funded by the European Union Civil Protection

- Installation guides (this document).
- User Manual.
- DSS Implementation report.

# 2. Services

The platform consists of the following services:

- API: Consisting of a REST API that will allow the consultation, creation, update, and deletion of data. This service will be in charge of user authentication.
- WEB: This service will be in charge of serving the resources and managing the redirection of requests to the data service.

## 2.1. Software Requirements

- Operating System: Windows/Linux
- PostgreSQL https://www.postgresql.org/ (>9.5)
- PostGIS: https://postgis.net/
- Java/OpenJDK 1.8
- Apache with mod_proxy_ajp, mod_ssl, mod_rewrite https://httpd.apache.org/
- JDBC authentication driver

## 2.2. Files

- API service resources:
  *https://github.com/MANIFESTS-DSS/COPTOOL/tree/main/API_Service*
- WEB service resources:
  *https://github.com/MANIFESTS-DSS/COPTOOL/tree/main/WEB_Service*
- Database backup (PostgreSQL):
  *https://github.com/MANIFESTS-DSS/COPTOOL/tree/main/Database*
- *JDBC driver:*
  *https://github.com/MANIFESTS-DSS/COPTOOL/tree/main/JDBC_driver*
- *WinSW software:*
  *https://github.com/MANIFESTS-DSS/COPTOOL/tree/main/WinSW*

# 3. Database

The database requires postgreSQL 9.5 or higher with the PostGIS extension (created with PostGis v3.1).

Create the user and the database

```
CREATE USER coptool WITH SUPERUSER PASSWORD [yourpassword];
CREATE DATABASE "coptool" WITH owner=coptool;
```

Edit pg_hba.conf to trust local addresses.

Import into the database the data structure and default data from the backup **coptool.sql**. This backup was created with PostGis v3.1.

```
pg_restore -h localhost -p 5432 -U coptool -d coptool -v coptool.sql
```

The connection to the database will be configured in the properties file application.yml of the API service.

# 4. Setting API Service

1- Get API service code from GitHub.

2- Edit application.yml from \BOOT-INF\classes to suit your application environment.

```
server:
  port: ${port:9000}
spring:
  http:
    multipart:
      enabled: true
      location: /path/to/temp
      max-file-size: 200MB
      max-request-size: 200MB
  application:
    name: coptool-api
  datasource:
    platform: postgres
    url: jdbc:postgresql://localhost:5432/${db:coptool}
    username: coptool
    password: [yourpassword]
arcopol:
  visor-url: http://url/to/viewer/
```

3- Compress META-INF, BOOT-INF and org folders into a zip file, with no compression.

4- Rename the compressed file to coptool-api.jar

## 5. Setting WEB Service

1- Get WEB service code from GitHub.

2- Edit application.yml from \BOOT-INF\classes to suit your application environment.

```
server:
  port: 8080
tomcat:
  ajpEnabled: true
  ajpPort: 8010
  ajpSecretRequired: false
zuul:
  routes:
    api:
      service-id: api
      path: /api/**
      custom-sensitive-headers: true
      sensitive-headers: Cookie,Set-Cookie
      url: http://localhost:9000
    pub:
      service-id: pub
      path: /pub/**
      custom-sensitive-headers: true
      sensitive-headers: Cookie,Set-Cookie
      url: http://localhost:9000/pub
    auth:
      service-id: auth
      path: /auth/**
      custom-sensitive-headers: true
      sensitive-headers: Cookie,Set-Cookie
      url: http://localhost:9000/auth
```

3. In the root folder, compress META-INF, BOOT-INF and org  folders into a zip file, with no compression

4. Rename the compressed file to coptool-web.jar

## 6. Services deployment (LINUX)

A folder must be created for each service. Copy the *.jar files inside them.

```
/path/to/coptool/api (API)
    -   coptool-api.jar
/path/to/coptool/web (WEB)
    -   coptool-web.jar
```

Create a repository and temp folder

```
/path/to/coptool/repository
/path/to/coptool/temp
```

Set permissions in jar files

```
chmod +x /path/to/coptool/api/coptool-api.jar
chmod +x /path/to/coptool/web/coptool-web.jar
```

Set system services

```
# /etc/init.d
ln -s /path/to/coptool/api/coptool-api.jar /etc/init.d/coptool-api
ln -s /path/to/coptool/web/coptool-web.jar /etc/init.d/coptool-web
```

Start services

```
systemctl start coptool-api
systemctl start coptool-web
```

## 7. Service deployment (Windows)

A folder must be created for each service. Copy the *.jar files inside them.

```
C:\path\to\coptool\api (API)
    -   coptool-api.jar
C:\path\to\coptool\web (WEB)
    -   coptool-web.jar
```

Create a repository and temp folder

```
C:\path\to coptool\repository
C:\path\to coptool\temp
```

Open two consoles (one for each of the folders) and test the execution with the following commands

```
C:\path\to coptool\api\java –jar coptool-api.jar
       (..)
C:\path\to coptool\api\java –jar coptool-web.jar
```

Opening a browser the service should be accessible locally at the URL: [localhost:8080](localhost:8080)

### *Install as a service*

In windows environment, it is recommended that the services run as a system service. For this task, WinSW tool can be used. Some template files are provided in GitHub folder.

https://github.com/winsw/winsw

coptool-api.xml sample

```
<service>
    <id>CopTool-API</id>
    <name>CopTool-API</name>
    <description>CopTool API REST</description>
    <env name="MYAPP_HOME" value="%BASE%"/>
    <executable>C:\Program Files\Java\jdk1.8.0_45\bin\java</executable>
    <arguments>-Xmx512m -jar "C:\path\to\coptool\api\coptool-api.jar"</arguments>
    <logmode>rotate</logmode>
</service>
```

# 8. Apache web server

It is mandatory for the service to be published through a secure protocol in order to geolocation functionalities work properly.

Verify SSL context: \Apache\conf\extra\httpd-ssl.conf

```
<VirtualHost _default_:443>

ProxyRequests On
ProxyPass / ajp://localhost:8010/ retry=0
ProxyPassReverse / ajp://localhost:8010/
Timeout 600
ProxyTimeout 600
Header set Access-Control-Allow-Origin "*"
Header always set Access-Control-Allow-Headers "*"
SSLCertificateFile "${SRVROOT}/conf/server.crt"
SSLCertificateKeyFile "${SRVROOT}/conf/server.key"
SSLCACertificateFile "${SRVROOT}/conf/ssl/ca_bundle.crt"
```

Verify settings in \Apache\conf\httpd.conf

```
Listen 80

LoadModule proxy_http_module modules/mod_proxy_http.so
```

Co-funded by the European Union Civil Protection

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule ssl_module modules/mod_ssl.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
LoadModule headers_module modules/mod_headers.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so

<IfModule ssl_module>
Include conf/extra/httpd-ssl.conf
```

# 9. Publishing service

External access should be through a subdomain e.g. coptool.plancamgal.gal under HTTPS. Proxy, firewall, etc must be configured so that port 443 is accessible from the outside.

Co-funded by the European Union Civil Protection

# COP Viewer

COP Viewer INSTALLATION GUIDE

Pedro Montero, Garbiñe Ayensa, Silvia Allen-Perkins, Alberto Gómez

Co-funded by the European Union Civil Protection

## ACKNOWLEDGEMENT

The work described in this report was supported by the Directorate-General for European Civil Protection and Humanitarian Aid Operations (DG-ECHO) of the European Union through the Grant Agreement number 101004912 - MANIFESTS – UCPM-2020-PP-AG, corresponding to the Call objective “Enhancing prevention and protection from the effects of maritime disasters” under priority 1: “Developing response capacity for marine pollution”.

Co-funded by the European Union Civil
Protection

| | |
|---|---|
| Project Acronym | **MANIFESTS** |
| Project Full Title | **MAN**aging risks and **I**mpacts **F**rom **E**vaporating and gaseous **S**ubstances **T**o population **S**afety |
| Gant Agreement Nr. | **101004912** |
| Project Website | [https://www.manifests-project.eu/](https://www.manifests-project.eu/) |

| | |
|---|---|
| Deliverable Nr. | **D5.2 MANIFESTS DSS Installation guide** |
| Status (Final/Draft/Revised) | **Final** |
| Work Package | **5** |
| Task Number | **5.4** |
| Responsible Institute | **INTECMAR** |
| Author/s | **P. Montero, G. Ayensa, S. Allen-Perkins, A. Gómez** |
| Recommended Citation | |
| Dissemination Level | |

| Document History | | | |
|---|---|---|---|
| Version | Date | Modification Introduced | |
| | | Modification Reason | Modified by |
| 1.0 | 31/01/2023 | | |
| | | | |
| | | | |

Co-funded by the European Union Civil Protection

## Content

Co-funded by the European Union Civil Protection

# 1. Background

In the event of a maritime accident involving HNS, maritime authorities must take numerous decisions to organize the best response strategy, i.e., one that minimizes risks to human health (including incident response teams, crew members and coastal communities), to the marine environment, for maritime safety and for socio-economic activities and facilities. While many key decisions and considerations are prescribed in national or regional contingency plans, operational response activities will generally need ongoing adjustment or review to reflect the most recent information available as the contamination event evolves. In such a rapidly changing situation, an efficient exchange of information between competent decision-making authorities and response teams on the ground can greatly facilitate both decision-making processes and organizational processes.

The purpose of work package 5 of the Manifests project (Manifests decision support system) is to develop an efficient information system that helps (1) decision makers understand the situation at stake and its likely evolution in the coming hours and days; (2) identify the population, ecosystems and socio-economic assets at risk and (3) share useful information with response teams deployed at sea, in the air or on the coast.

Building on the experience gained and development carried out during the previous HNS-MS and MARINER projects, the MANIFESTS decision support system (DSS) will integrate several services, including the DSS Common Operational Picture (COPtool).

This COPtool refers to a system designed so that during a contingency, the exchange of information that occurs between the maritime authorities and the different response teams (sea, coast, air) is carried out in the most efficient way possible, ensuring that all actors involved in the crisis committee and response teams can access the same data. These can be standard reports (such as the Standard Pollution Observation Report of the Bonn Agreement), images, videos and any other georeferenced data collected by response teams, as well as satellite observations, model simulation results, exclusion areas, location of response media, requests for new response actions shared by the crisis.

The COPtool documentation is composed by:

Co-funded by the European Union Civil Protection

- Installation guides (this document).
- User Manual
- DSS Implementation report.

# 2. Administration guide

The viewer has several configuration files adjustable to the needs of the users who manage them:

## 2.1. Map configuration (basemap.js)

**File .js in** which you define, in JSON format, the different layers or services that will be used as basemap in the viewer. The properties available in each of them are the following:

- **name**. The name by which the layer will be recognized and displayed in the layer selector of the viewer. Example: 'IGN-BASE'.
- **URL**. Address of the service to load the map.
- **layers**. Identifiers of the layers to be recovered.
- **CRS**. Coordinate Reference System for this map.
- **version**. Version of the layer.
- **attribution**. Attribution of the map.
- **default**. Possible values are *True* or *False* and indicates whether the layer is enabled by default when you start the map. Only one of the basemaps can be set *to True*.

## 2.2. Default application settings (cfg.js)

**File .js in** which the default configuration of the application is defined, in JSON format. The available properties are as follows:

- **profile**. By default, the profile will be 'default'.
- **proxy**. This field indicates the file where the *proxy* used by the application is located.
- **mapProxy**. Possible values are *True* or *False*.
- **layerProxy**. Possible values are *True* or *False*.
- **encoder**. This field indicates the file where the *encoder* used in the application is located.
- **defaultLocale**. Default language of the application.

Co-funded by the European Union Civil Protection

- **availabelPremises**. *Array* with the different languages available in the application.
- **CRS**. Coordinate Reference System used by Leaflet by default in the application.
- **initialZoom**. The zoom or distance with which the viewer is loaded by default.
- **minZoom**. Maximum distance to which we can move away in the viewer view.
- **maxZoom**. Maximum distance to which we can approach in the viewer view.
- **initialCenter**. The coordinates of the point where the view will focus when you first load your view.
- **ignoreData**. Possible values are *True* or *False*. Indicates, if *True*, that there is no initial data to load.
- **layersXML**. List of different XML from which the cartography layers will be retrieved.
- **toponymURL**. URL of the toponymy service.
- **toponymField**. Name of the toponymy field.
- **toponymName**. Name of toponymy.
- **toponymFeatName**. The name of the toponymy feature.
- **bitacoraUrl**. Bitacora URL.
- **episodes.** URL of the episodes.

## 2.3. Layer configuration (eg test.xml)

**Files .xml** in which you can define the different layers that can be loaded into the viewer, as well as add new ones. To do this, these files must be referenced in the **cfg.js** configuration file, specifically in **layersXML**.

The elements that make up these. XML are as follows:

- **EpisodeId**. The ID of the episode to which the layers belong.
- **BeginTime**. Start date.
- **EndTime**. End date.
- **LatLonBoundingBox**. Coordinates that delimit the maximum surface area to be covered by the layers to be loaded.
- **Panel**. A panel under which a series of layer groups or layers are grouped. It has a **label** attribute, which is the name (or key to translate using the translation files) that it is in the panel.

Co-funded by the European Union Civil Protection

- **LayerGroup**. A group of layers under which different layers are grouped. It has a group attribute responsible for naming that group, in the same way as the **label** attribute in a **Panel** element.
  - **Layer**. Each of the layers that can be loaded into the application viewer.
    - **Id**. The ID of the layer.
    - **Name**. The name of the layer.
    - **Title**. The title of the layer displayed in the layer collector in the viewer.
    - **LatLonBoundingBox**. Coordinates that delimit the maximum surface area covered by the layer.
    - **Style**. The style of the layer.
    - **Filter**. The filter of the layer.
    - **LayerType**. The type of the layer.
    - **Url**. URL from which the layer is retrieved.

## 2.4. Translation files into different languages

**Files .js, one for each of the** languages available in the application, both for the "main" application and for the Log, which serve as a kind of dictionary, in which a certain key is assigned a value, which corresponds to the text we want to use in the application in the language that interests us. Example:

```
...

'campanha': 'Campaign',

'bathymetry': 'Bathymetry (m)',

...
```

To change the translation of a certain text is as simple as modifying the second field. If you want to add a new translation, simply add a new key-value pair in each of the languages.

# 3. Installation guide

To deploy the project on the server and ensure its proper functioning, it is only necessary to have **php** (version 7.3 or higher) installed on the server.  In addition, it is necessary to have the **php cURL** library (usually already is installed by default;  if this is not the case, it will be necessary to install it).

If the above requirements are met, it is sufficient to decompress or clone the source code of the project (if obtained through a ZIP or from a Git repository, respectively), and paste the source code into the server.

**IMPORTANT**: The source code must be copied to the server maintaining the exact structure of the application to avoid problems and communication errors between the components of the application (such as Log).

# D5.2 - MANIFESTS DSS

# Installation guide

Web application - Model interface

Samuël Orsi

## ACKNOWLEDGEMENT

The work described in this report was supported by the Directorate-General for European Civil Protection and Humanitarian Aid Operations (DG-ECHO) of the European Union through the Grant Agreement number 101004912 - MANIFESTS – UCPM-2020-PP-AG, corresponding to the Call objective “Enhancing prevention and protection from the effects of maritime disasters” under priority 1: “Developing response capacity for marine pollution”.

Co-funded by the European Union

| | |
|---|---|
| **Project Acronym** | **MANIFESTS** |
| **Project Full Title** | **MAN**aging risks and **I**mpacts **F**rom **E**vaporating and gaseous **S**ubstances **T**o population **S**afety |
| **Gant Agreement Nr.** | **101004912** |
| **Project Website** | **https://www.manifests-project.eu/** |

| | |
|---|---|
| **Deliverable Nr.** | **D5.2** |
| **Status (Final/Draft/Revised)** | **Final** |
| **Work Package** | **WP5** |
| **Task Number** | **5.3** |
| **Responsible Institute** | **RBINS** |
| **Author/s** | **Samuël Orsi** |
| **Recommended Citation** | **Samuël Orsi (2023)**<br>**D5.2 - MANIFESTS DSS - Installation guide**<br>**Deliverable of MANIFESTS project** |
| **Dissemination Level** | **MANIFESTS Consortium, DG ECHO** |

| **Document History** | | | |
|---|---|---|---|
| **Version** | **Date** | **Modification Introduced** | |
| | | **Modification Reason** | **Modified by** |
| 1.0.0 | 24/04/2023 | First version | Samuël Orsi |
| | | | |
| | | | |

Co-funded by the European Union

# 1. Introduction

This document is focused on the installation and setting up of a model user interface application as developed in the scope of the MANIFESTS project.

It describes a typical installation procedure on Debian and RedHat (RHEL) based systems.

A good knowledge of Linux server administration and of the python language is necessary to understand and perform this task.

This document is dedicated to experimented users.

As the technology evolve and operating systems gets updated, the following installation procedure must be used as a base of work and adapted accordingly.

# 2. Prerequisites

- A Linux system with a root console access and the Apache web server installed.
- Valid reCAPTCHA v2 keys

# 3. Installation guide

## 3.1. Python 3.8 Installation

This procedure makes sure that the minimal version of python is available. Further python modules will be installed automatically during deployment of the application code.

### 3.1.1. Ubuntu 22.04

```
sudo apt update && sudo apt upgrade

sudo add-apt-repository ppa:deadsnakes/ppa

sudo apt update

sudo    apt    install    libpython3.8-dev    libpython3.8-minimal    ↲
 libpython3.8-stdlib   libpython3.8   python3.8-dev   python3.8-minimal   ↲
 python3.8-venv python3.8
```

### 3.1.2. Debian 11

```
sudo apt update && sudo apt upgrade

sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev ↲
 libnss3-dev  libssl-dev  libsqlite3-dev  libreadline-dev  libffi-dev  ↲
 curl  libbz2-dev  make  wget  llvm  libncursesw5-dev  xz-utils  tk-dev  ↲
 liblzma-dev checkinstall

cd /opt

wget https://www.python.org/ftp/python/3.8.12/Python-3.8.12.tar.xz

tar -xf Python-3.8.12.tar.xz

cd /opt/Python3.8.12/

./configure         --enable-optimizations         -enable-shared         ↲
 -with-ensurepip=install                   --without-gcc                 ↲
 LDFLAGS="-Wl,--rpath=\$$LIB:/usr/local/lib"

make -j 8
```

**! Keep an eye on the output !**

The message "**Python build finished successfully!**" does not mean the compilation succeeded. You must take care of all error messages depending on your system.

```
checkinstall   -D   --install=no   --fstrans=no   --pkgname=python3.8   ↲
 make altinstall
```

```
dpkg -i python3.8_3.8.12-1_amd64.deb

mv python3.8_3.8.12-1_amd64.deb /home/www-data/

cd /opt

rm -r Python-3.8.12/

rm Python-3.6.8.tgz
```

#### 3.1.3. Fedora 37

```
sudo dnf update

sudo dnf install python3.8
```

## 3.2. Setup Python virtual environments

Virtual environments ensure all the packages and versions you install only apply to this specific application and will not affect your entire system. You can see it as a private island.

The configuration path and virtual environments have been put in slightly different locations for different OSes.

- On Debian/Ubuntu type systems: /var/opt/<project_name>/
- On RHEL/Fedora systems: /opt/<project_name>/

From now on, both paths will be referred to as "<pathconfdir>"

#### 3.2.1. On all systems

```
cd <pathconfdir>

mkdir -p Django

mkdir -p Django_env_vars

mkdir -p apache_configs

sudo python3.8 -m venv ./venv
```

## 3.3. Install PostgreSQL with PostGIS 3

#### 3.3.1. Ubuntu 22.04

```
sudo apt update && sudo apt upgrade

sudo apt install postgresql postgresql-contrib postgresql-14-postgis-3 ↲
 postgresql-14-postgis-3-scripts
```

Co-funded by the European Union

### 3.3.2. Debian 11

```
sudo apt update && sudo apt upgrade
```

```
sudo apt install postgresql postgresql-contrib postgresql-13-postgis-3 ↲
 postgresql-13-postgis-3-scripts
```

### 3.3.3. Fedora 37

sudo dnf update

sudo dnf install postgresql postgresql-contrib postgis

## 3.4. Create Database and role

A “role” is a database user.

The following procedure is the same for all systems.
You must change the followings according to your needs:

- “<db_username>” refers to the username used to connect to the database
- “<database>” refers to the name of the database

### 3.4.1. Create role

```
sudo -u postgres createuser –interactive
```

Output :

```
Enter name of role to add: <db_username>

Shall the new role be a superuser? (y/n) n
```

### 3.4.2. Create database

```
sudo -u postgres createdb -O <db_username> <database>
```

### 3.4.3. Enable PostGIS extensions

```
sudo -u postgres

psql -U <db_username> -d <database>

CREATE EXTENSION postgis;

CREATE EXTENSION postgis_raster;

CREATE EXTENSION postgis_topology;

\q

exit
```

## 3.5. Set your environment variables

Environment variables are sensitive information you don't want to publish in any code repository. They are in the ".env" file into <pathconfdir>/Django_env_vars/

You can add as many parameters as you need for your project.

Here is the list of mandatory parameters:

- DEBUG=False
- SECRET_KEY="<Django_secret_key>"
- API_SECRET_SALT_KEY="<API_PreShardKey>"
- JWT_RSA_PRIVATE_KEY_FILE=JWT_RSA_PRIVATE_KEY.pem
- JWT_RSA_PUBLIC_KEY_FILE=JWT_RSA_PUBLIC_KEY.pem
- ALLOWED_HOSTS="*"
- DB_NAME=<database>
- DB_USER=<db_username>
- DB_PASSWORD=<db_username_password>
- DB_HOST=localhost
- DB_PORT=5432
- ADMINS="<admin_name>, <admin_email>"
- MANAGERS="<manager_name>, <manager_email>"
- RECAPTCHA_PRIVATE_KEY=<private_key>
- RECAPTCHA_PUBLIC_KEY=<public_key>

## 3.6. Deploy application code

### 3.6.1. Clone the code from a git repository

The following example shows how to clone from a GitHub repository.

Alternatively, you can manually copy the application code to : <pathconfdir>/Django/<project_name>

```
cd <pathconfdir>

git    clone    git@github.com:<github_user>/<project_name>.git    ↲
 Django/<project_name>
```

### 3.6.2. install python dependencies

```
cd <pathconfdir>

source venv/bin/activate

cd Django/<project_name>

pip   install   --no-cache-dir   --upgrade   --force-reinstall   -r   ↲
 requirements.txt
```

```
python manage.py migrate

python manage.py collectstatic

python manage.py createsuperuser

deactivate
```

#### 3.6.3. Set user rights

- On Debian/Ubuntu:
  ```
  cd <pathconfdir>
  chown -R www-data.www-data Django
  ```
- On RHEL/Fedora:
  ```
  cd <pathconfdir>
  chown -R apache.apache Django
  ```

## 3.7. Generate mod_wsgi configuration

#### 3.7.1. Debian/Ubuntu

```
cd <pathconfdir>

source venv/bin/activate

cd <pathconfdir>/Django/<project_name>

mod_wsgi-express setup-server <project_name>/wsgi.py --port=8002 ↲
 --user www-data --group www-data --url-alias /static ./static ↲
 --url-alias /media ./media --url-alias /templates ./templates ↲
 --server-root=../../apache_configs

deactivate
```

#### 3.7.2. RHEL/Fedora

```
cd <pathconfdir>

source venv/bin/activate

cd <pathconfdir>/Django/<project_name>

mod_wsgi-express setup-server <project_name>/wsgi.py --port=8002 ↲
 --user apache --group apache --url-alias /static ./static ↲
 --url-alias /media ./media --url-alias /templates ./templates ↲
 --server-root=../../apache_configs

deactivate
```

## 3.8. Configure application as a system service

### 3.8.1. Creating systemd unit (all systems)

```
sudo nano /etc/systemd/system/<project_name>_mod_wsgi.service
```

Paste the following, using the correct path to your project :

```
[Unit]
Description=<project_name> via apache2 with mod_wsgi
After=network.target remote-fs.target nss-lookup.target
Documentation=https://httpd.apache.org/docs/2.4/

[Service]
Type=forking
Environment=APACHE_STARTED_BY_SYSTEMD=true
ExecStart=<pathconfdir>/apache_configs/apachectl start
ExecStop=<pathconfdir>/apache_configs/apachectl graceful-stop
ExecReload=<pathconfdir>/apache_configs/apachectl graceful
KillMode=mixed
PrivateTmp=true
Restart=on-abort
RestartSec=10s
TimeoutSec=10s
OOMPolicy=continue

[Install]
WantedBy=multi-user.target
```

Enable the service and start it.

### 3.8.2. Available commands

- Enable service

  sudo systemctl enable <project_name>_mod_wsgi.service

- Start the server

  sudo systemctl start <project_name>_mod_wsgi.service

- Get server status

  sudo systemctl status <project_name>_mod_wsgi.service

- Reload the server

  sudo systemctl reload <project_name>_mod_wsgi.service

Co-funded by the European Union

- Stop the server
  sudo systemctl stop <project_name>_mod_wsgi.service

## 3.9. Setup global apache proxy

Now that the application is started as a system service. It is not yet accessible on the network. For this we need a gateway via the native system apache.

### 3.9.1. Activate proxy module

sudo a2enmod proxy_http

### 3.9.2. Create an apache config file

We strongly advise the file name be prefixed by an unique ID, then the name of the website.

```
cd /etc/apache2/sites-available/

sudo cp 000-default.conf 001-<website_URL>.conf
```

Paste the following in your configuration file:

```
# <project_name> (Start)
Alias /oserit/static    <pathconfdir>/Django/<project_name>/static
Alias /oserit/media     <pathconfdir>/Django/<project_name>/media
Alias /oserit/templates <pathconfdir>/Django/<project_name>/templates

<Directory "<pathconfdir>/Django/<project_name>/static">
Options -Indexes +FollowSymLinks
AllowOverride None
Require all granted
</Directory>
<Directory "<pathconfdir>/Django/<project_name>/media">
Options -Indexes +FollowSymLinks
AllowOverride None
Require all granted
</Directory>
<Directory "<pathconfdir>/Django/<project_name>/templates">
Options -Indexes
AllowOverride None
Require all granted
</Directory>


ProxyPass /oserit http://127.0.0.1:8002 flushpackets=On keepalive=On
ProxyPassReverse /oserit  http://127.0.0.1:8002
```

```
# <project_name> (End)
```

Modify the config file according to your project and network infrastructure needs. You can test your configuration with the following command:

```
sudo apachectl configtest
```

And finally start the website proxy to make it accessible.

```
sudo a2ensite 001-www.my-example-website.com
```

## 4. Administration

You can log into the administration interface using the superadmin user at:

```
<website_URL>/django-admin/
```

From there you need to create a user group per Django app you developed and allocate them to app users.

## 5. Glossary

This glossary contains all the term that need to be changed by the end user.

| | |
|---|---|
| <project_name> | The project name |
| <pathconfdir> | Path to the directory the project is located at. |
| <db_username> | Username used to connect to the database |
| <database> | Name of the PostgreSQL database |
| <Django_secret_key> | The Djando secret key |
| <API_PreShardKey> | The API pre shared key (must be at least 64 chars) |
| <db_username_password> | Password to connect to the database |
| <admin_name> | Administrator name |
| <admin_email> | Administrator e-mail |
| <manager_name> | Manager name |
| <manager_email> | Manager e-mail |

**MANIFESTS**
**MAN**aging risks and **I**mpacts **F**rom **E**vaporating and gaseous **S**ubstances **T**o population **S**afety

Co-funded by the European Union

| | |
|---|---|
| <private_key> | reCAPTCHA v2 private key |
| <public_key> | reCAPTCHA v2 public key |
| <github_user> | GitHub username |
| <website_URL> | Project website URL where the service will be accessible |

Co-funded by the European Union