# User Manual

This guide aims to provide the user with a guide for the configuration of the environment for the correct operation of the application. These instructions are provided as a series of consecutive steps that are all necessary requirements for the operation of the application. The user is recommended to follow the steps in the order indicated and to carry out all the instructions in the guide using the material provided in this repository.

It is also recommended to have previously read the tool article to know the scope and structure of the tool.

## Introduction

The tool is composed of numerous Python scripts in version 3.9.7 that perform all the tasks of acquiring, processing and displaying the relevant information. In addition to the Python scripts, the source code of the Web Application developed with the React framework is provided. These elements together with the sources that will be developed in this document are enough to build the tool and proceed with its use. It is necessary for the configuration of the environment to have a MongoDB Atlas service account or a local version of this MongoDB service, if you want to query GDELT data through the BigQuery service, it will be necessary to have an account of this service and a proper configuration of this account. The repository also includes a dump file corresponding to the database status of the tool at the time of the demonstration (10/9/2021). All the necessary steps for the use of this material are developed in this guide.

## Setting up the environment

The use of a virtual environment is recommended to simplify the management of dependencies and versions of the scripts. As it is necessary to execute both the Web Application and the execution of certain python scripts simultaneously, it is also recommended to use different IDEs to separate these executions. To illustrate the configuration of the tool, the IDE Spyder has been chosen for the execution of the python scripts and the IDE Visual Studio for the execution of the Web Application.

In this way, the first step for setting up the environment is creating the python virtual environment.

## Creation of the virtual environment and dependencies installation

Requirements:

* Python 3 installed

As the creation of the virtual environment depends on the system, it is recommended to follow a guide for creating the python virtual environment for the corresponding system.

Windows:

* python3 -m venv c: \path\to\new\virtual\environment

Linux & MacOS:

* python3 -m venv /path/to/new/virtual/environment

Remember that for the correct use of the virtual environment using an IDE it is necessary to configure the virtual environment in the IDE

Spyder:

Captura de pantalla de computadora

Descripción generada automáticamente

Once the virtual environment is created, the installation of all the necessary dependencies is required. A dependencies file is provided with the name "requirements" for the automatic installation of dependencies.

To actívate the virtual environment execute “Scripts/active.bat” file

To install dependencies:

* pip install -r requirements.txt

To check dependecies are intalled:

* pip freeze

## Setting up MongoDB

Depending on whether the MongoDB Atlas or MongoDB service is used locally, the configuration may vary, the steps to be followed instead are the same, so it is recommended to follow all the steps that are developed below. In the particular case, the MongoDB Atlas service will be used.

### Account Creation

The first step in using the MongoDB service is creating an account on the service.

### User Creation and Access Management

Once the service account has been created, to use the service it is necessary to create a user and manage the IP addresses that wish to grant access to the database.

### Database Creation

illeucture that maintains the MongoDB database when it is a NoSQL database is made up of a database made up of collections in turn made up of documents. In our case, the collections ill be created automatically by the ille scripts, in this way, we only need to ille the database that ill contain the collections that are created.

### Linking the database with Python

To enable python to interact with the database it is necessary to configure the access to the database created, for this it is required to have previously installed the dependencies, to grant access to the computers that are going to access the database and to specify the necessary data to access the database.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Click Connect

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Choose Connect your application

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Select the right driver, version and include the driver code

The result is an string with the following format:

* client=pymongo.MongoClient("mongodb+srv://<User>:<password>@<cluster\_url>/<Database\_name>?retryWrites=true&w=majority")

this string is included in all required Python script, so the user just need to set the proper values for the connection.

Remember to avoid special characters such as '@' in the user's password, the presence of these characters can cause connection errors.

## Incidents dataset creation

The first script to be executed for the incident dataset creation is the script "MongoDB\_Group\_incident\_dataset\_generator".

This script is in charge of extracting the advanced incident information from the MITRE and ThaiCert data sources. Once the information is extracted, it is stored in the previously configured database. That is why the previously exposed modifications are a requirement for the execution of this script.

In addition, the script generates a text file called "incidents\_dataset\_generation\_log" which stores all the data queried from the data sources.