
BEL Commons Documentation

Release 0.3.1

Charles Tapley Hoyt

Feb 24, 2023

Contents

1	Installation	1
1.1	Database	1
1.2	Message Broker	1
1.3	Server	1
2	Running with Docker	3
2.1	Dockerfile	3
3	Pages	5
4	Configuration	7
5	Indices and tables	9

This application runs on Python 3.7+.

1.1 Database

For production, it is preferred to use a multi-threading relational database management system. PyBEL has been best tested on PostgreSQL, so this is preferred for now.

1.2 Message Broker

This application uses [Celery](#) as a task management system to support asynchronous parsing of BEL documents, running of analyses, and other slow operations.

RabbitMQ, or any other message queue supported by Celery are appropriate.

1.3 Server

Because this application is built with Flask, it can be run with the WSGI protocol. Running on a single machine is possible with either the built-in `werkzeug` test server or something easy to install like `gunicorn`.

For production, `uwsgi` seems to work pretty well.

Running with Docker

Docker is very powerful as a general way to specify how things should be installed, but has a steep learning curve. After installing and running `docker-machine` and `docker-compose`, BEL Commons can be run with a few simple commands.

2.1 Dockerfile

A simple Dockerfile is included at the root-level of the repository. This Dockerfile is inspired by the tutorials from [Container Tutorials](#) and [Digital Ocean](#).

Warning:

- The virtual machine needs at least 2GB memory for the worker container
- The database needs a packet size big enough to accommodate large BEL files (>10 mb)

CHAPTER 3

Pages

CHAPTER 4

Configuration

Default configuration can be found in the module `bel_commons.config`.

By default, PyBEL searches for a configuration file called `config.json` in `~/.config/pybel/`. This directory can be modified with the environment variable `PYBEL_CONFIG_DIRECTORY`. Additionally, the location of another custom configuration can be specified by the environment variable `BEL_COMMONS_CONFIG_JSON`.

In `config.json` add an entry `PYBEL_MERGE_SERVER_PREFIX` for the address of the server. Example: `http://lisa:5000` with no trailing backslash. This is necessary since celery has a problem with flask's url builder function `flask.url_for`.

Add an entry `PYBEL_CONNECTION` with the database connection string to either a local SQLite database or a proper relational database management system. It's suggested to `pip install psycopg2-binary` in combination with MySQL since it enables multi-threading.

For a deployment with a local instance of RabbitMQ, the default configuration already contains a setting for `amqp://localhost`. Otherwise, an entry `CELERY_BROKER_URL` can be set.

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`