

# CollectorServer

## Abstract

This document describes CollectorServer, the part of Lorem Ipsum project. The aim of server is to collect test results provided by application. Below you can find information about server features, installation and maintenance instruction.

## Features

- receiving result packages

## System requirements

The server requires following components to be available on target platform:

- Python 3.3 or higher (<http://www.python.org/>)
- Tornado Web Server 3.2 or higher (<http://www.tornadoweb.org/>)

Server should work on any platform supporting features. One of such platform is Fedora 20 (Linux), which is the suggested one. Installation instruction below describes some details about installing required components.

## Installation

Following steps describe installation process on target machine. Each step contains both general information and details about how to proceed on suggested operating system (Fedora 20).

1. **General:** Install Python 3.3 or newer (it is also called CPython). On most of operating systems you can install it using bundled package manager. Otherwise, you can always look for instructions and appropriate packages on <http://www.python.org/>.

**Fedora:** Please run following command in the system terminal:

```
sudo yum -y install python3
```

If prompted, provide your root password.

2. **General:** Install Tornado Web Server 3.2 or newer. On most of operating systems you can install it using bundled package manager. Otherwise, you can always look for release on <http://www.tornadoweb.org/>.

*Note:* Usual package name for Tornado is `python-tornado` or `python3-tornado`. If you have both options, please install version for *Python 3* (the latter one).

**Fedora:** Please run following command in the system terminal:

```
sudo yum -y install python3-tornado
```

If prompted, provide your root password.

3. **General:** Copy `CollectorServer` directory to appropriate place in your hard disk.

**Fedora:** Please run following command in the system terminal:

```
cp -R <source> <target>
```

replacing `<source>` by the location of `CollectorServer` directory

4. *Optional:* Do it, if you prefer to use other port number than the default, `40666`.

**General:** Open the configuration file `server.conf` and add following line:

```
port=<port number>
```

replacing `<port number>` with the port number that you'd like to run server on.

**Fedora:** Please run following command in the system terminal:

```
echo "port=<port number>" >> server.conf
```

replacing `<port number>` with the port number that you'd like to run server on.

5. **General:** Run `server.py` in detached mode. This can be done by running with `--start` argument *or* by using system `nohup` command. While being in `CollectorServer` server directory, full command line may look like:

```
python3 server.py --start
```

or

```
nohup python3 server.py&
```

If your system support hashbangs, you can use:

```
./server.py --start
```

or

```
nohup ./server.py&
```

*Note:* Make sure the server is started *using Python 3*, not the older version that may be installed on your system.

**Fedora:** Please navigate in the system terminal to the directory where the server has been copied and run the following command:

```
./server.py --start
```

or

```
nohup ./server.py&
```

If everything went, the server should be available through HTTPS protocol. Please open web browser and enter the following address:

`https://<hostname>:40666/`

or, in case if you set different port number:

`https://<hostname>:<port number>/`

replacing `<hostname>` by IP-address or host name of your server, and `<port number>` by the number that has been set in `server.conf`.

If you see the message informing that *connection is untrusted* or similar, this means that server is working properly. The message comes from the fact that server uses custom, self-signed certificate which is properly recognised by Lorem Ipsum application, but not by the web browsers.

## Maintenance

### Starting and stopping server

To start server, please run following command:

```
./server.py --start
```

To stop server, please run following command:

```
./server.py --stop
```

If your operating system does not support hashbang, you may need to replace `./server.py` parts by `python3 server.py`.

### Received results

The results that are submitted by application are stored in `CollectorServer/results` directory. They are expected to be ZIP files. To avoid conflicts, their names are the names provided by application, prepended by the moment of receiving in format `<year>-<month>-<day>_<hour>-<minute>-<second>`.

## Protocol

Communication between Collector Server and the application is conducted using HTTPS protocol (protocols HTTP and TLS as defined in RFC 2616 and RFC 6520). Server's interface has defined two endpoints, that are used for communication.

## Endpoints

`/submit`

Method: **POST**

Arguments: one (only one) file passed using **multipart/form-data** content type. The name is arbitrary, but server's implementation uses it as part of file name that is stored in results directory.

Result: HTTP status code 200 if submission has succeeded, or other in case of problems.