

# 1 DevOps Exercise: Python

## 1.1 Introduction

For the second interview we would like you to complete the following exercise. The goal of this small assessment is to get a good impression of your Python skills.

During the interview we will execute the application to test whether it fulfills the requirements (on one of our Laptops) and we may also review the source code and briefly discuss it with you.

### 1.1.1 Submitting the result to us

Please respond *directly* to the E-Mail through which you have been provided with this exercise. This way it will end up on the corresponding HR Generalist's inbox.

The Python package (and its source-code) that is to be created shall be provided via some cloud storage provider (Dropbox, Google Drive, [sendspace.com](https://sendspace.com), etc).

## 1.2 Create a small Python 2 and Python 3 compatible application

### 1.2.1 Requirements

- save following content in **config.ini** file as input:

#### config.ini

```
[Data]
; --- data to be processed ---
username = testuser
urlpath  = /rest/1.0/request

[Urls]
; --- urls you can work with ---
url1 = https://jenkins1.tttech.com
url2 = https://jenkins2.tttech.com
url3 = https://jenkins3.tttech.com
url4 = https://jenkins4.tttech.com
```

- parse the file using the **configparser** module
- make it so the application can be started from the command-line using the **docopt** module (including `--help` )
- make it **Python2 and Python3 compatible** (any recent version)

## 1.2.2 Result

The result shall just be visible as text in the terminal (Windows or Linux).

Example output:

### Results

```
$ python2 rest_helper.py -n 2 -c config.ini

https://testuser@jenkins1.tttech.com/rest/1.0/request
https://testuser@jenkins2.tttech.com/rest/1.0/request

$ python3 rest_helper.py --num 4 --config config.ini

https://testuser@jenkins1.tttech.com/rest/1.0/request
https://testuser@jenkins2.tttech.com/rest/1.0/request
https://testuser@jenkins3.tttech.com/rest/1.0/request
https://testuser@jenkins4.tttech.com/rest/1.0/request
```

## 1.3 Create a Python package (optional)

Create a Python package (wheel) that can be installed via **pip** locally.

If you already (and only) know how to create egg packages that are installable via `easy_install`, this is an alternative.

## 1.4 Open Questions?

If you have any questions regarding this exercise please don't hesitate to get in touch with either Simon Steinbeiß ([simon.steinbeiss@tttech.com](mailto:simon.steinbeiss@tttech.com)) or Florian Schüller ([florian.schueller@tttech.com](mailto:florian.schueller@tttech.com))!