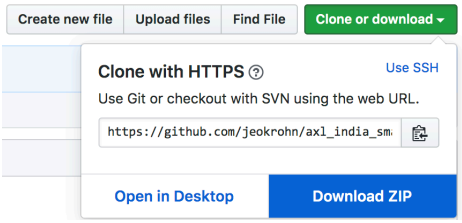
## Table of Contents

# 1 Download the GitHub repository

Browse to https://github.com/jeokrohn/axl_india_smart_cities

At the right hand side click on „Clone or download“ and then select „Download ZIP“.




Save the ZIP file to your computer.

Extract the ZIP file to a working directory of your choice; take note of the directory you extracted the ZIP file to.

# 2 Python/PyCharm vs. Docker

To be able to follow the AXL demos and execute own Python code during the lab you either have to install Python and PyCharm as IDE on your machine or at least install Docker.

If you install Python and PyCharm you will be able to use breakpoints in your code, inspect variables, and you will benefit from all the other features of the IDE.

To avoid the complexities of installing Python and the IDE locally you can use prepared Jupyter notebooks running in a Docker container instead. For this you only need to install Docker and then run the prepared Docker container. You will then use your web browser to access the Jupyter notebooks and execute Python code.

To install Docker follow the description in section 3, Install Docker.

To install Python and Pycharm follow the steps in sections 4, Installing Python to 7, Open the Project in PyCharm.

# 3 Install Docker

Download the Community Edition (Docker CE) of Docker Desktop for Mac or Windows from https://hub.docker.com/search/?type=edition&offering=community

The download is for free but requires creation of a free account (Docker ID).

Install the downloaded software.

Verify the operaton by executing this command from the CLI:

```
docker run hello-world
```

The output should be similar to:

```
JKROHN-M-106P:~ jkrohn$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest:
sha256:92695bc579f31df7a63da6922075d0666e565ceccad16b59c3374d2cf4e8e50e
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs
the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent
it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

# 4 Installing Python

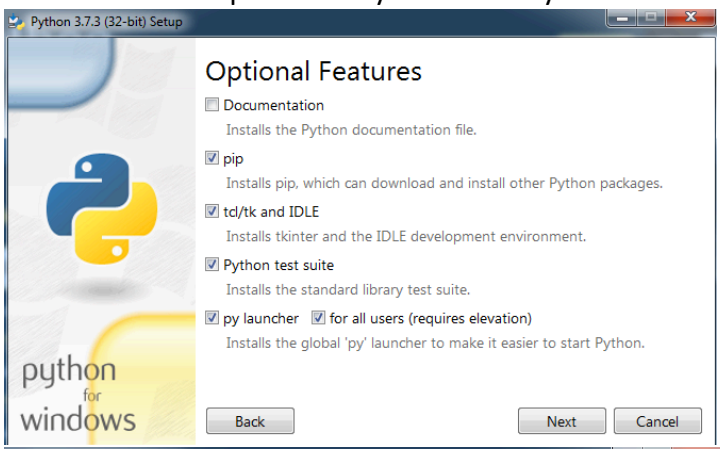
Installing Python (and PyCharm) is recommended (but not necessarily required). If you don't want to install Python and PyCharm you can also simply install Docker (see section 3) and use the pre-built Docker image to run a Jupyter environment with the demos.

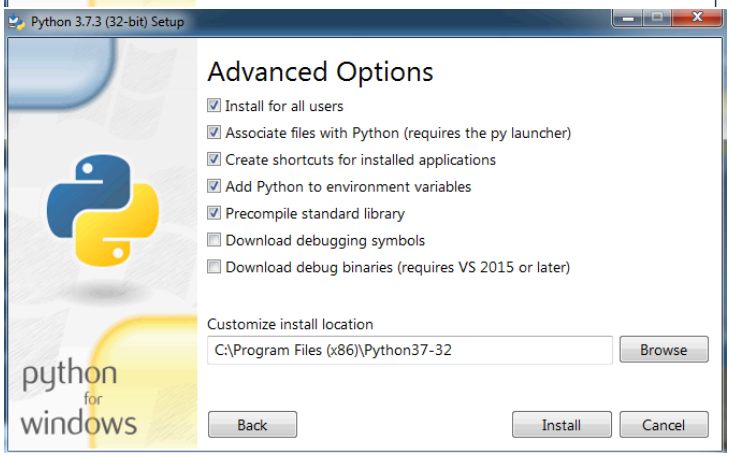
Download the latest stable Python release from https://www.python.org/
(currently release 3.7.3)


Run the installer

# !! Important: on Windows you have to perform an all-user installation !!

This installation option is only available if you select the “Customize installation” option.




Verify the release on the CLI (Terminal on OS-X, Command Prompt on Windows) by typing „python3“ on the CLI. The output should show the installed version:

```
JKROHN-M-106P:~ jkrohn$ python3
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
C:\Users\jkrohn>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

## 5 Installing PyCharm

Installing Python (and PyCharm) is recommended (but not necessarily required). If you don’t want to use PyCharm you an also simply install Docker (see section 3) and use the pre-built Docker image to run a Jupyter environment with the demos.

Download the installer for the “Community” edition from
https://www.jetbrains.com/pycharm/download

Run the Installer

## 6 Installing the project requirements

This step is only required if you installed Python and plan to use Python and not the prepared Docker image with the Juypter environment.

You might want to create a virtual environment 1st to keep all dependencies for this project in one place. Creating an managing Python virtual environments is outside the scope of this document. See https://docs.python.org/3/library/venv.html for more details.

To install the requirements open a CLI (Terminal on OS-X or command prompt on Windows).

Navigate to the project directory created by unzipping the GitHiub repository on you machine.
Then execute the following command:

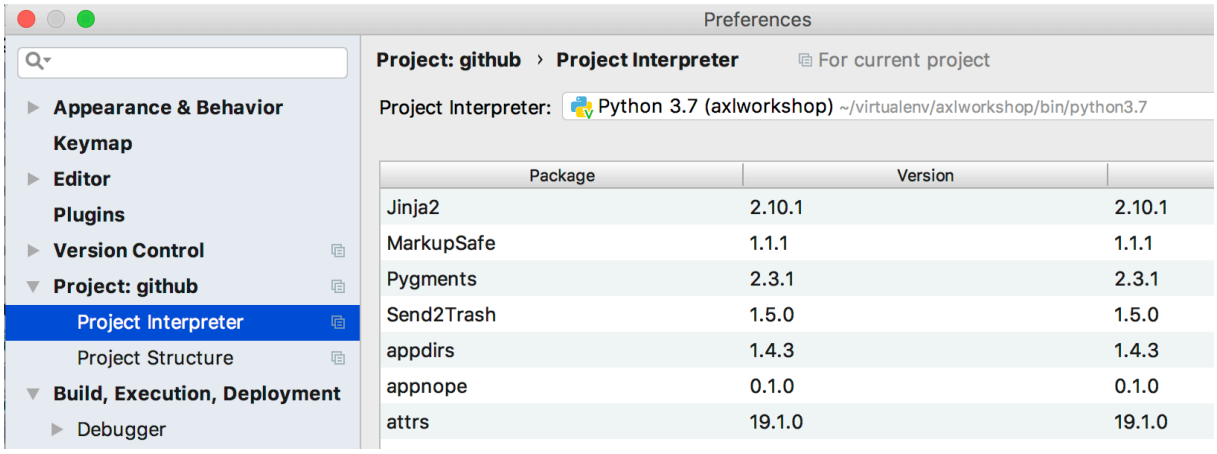
- On Windows: pip install -r requirements.txt
- On Mac: pip3 install -r requirements.txt

With this the requirements defined for the project will be installed.

## 7 Open the Project in PyCharm

In PyCharm using “File/Open” open the project directory you created by unzipping the archive downloaded from GitHub.

In the PyCharm preferences you then need to make sure to select Python 3.7 as the interpreter for your project:




After opening the preference, navigate to the “Project Interpreter” section under your project. To add a new project interpreter you can select the gear icon at the right.

If you chose to create a virtual environment above then make sure to select the Python interpreter in the virtual environment you created.

## 8 Running Python Code

## 8.1 Runnning Code in PyCharm

If you installed PyCharm then you can open a Python file in the project directory an execute that file from within PyCharm using the “Run” menu. In that menu you have the option to “Run” or “Debug” file. If you choose to “Debug” a file then the execution stops at breakpoints you can set in the Python source file by clicking on the grey area next to a line in the Python source code. As soon as code execution stops at a breakpoint you can inspect variable values and step through your code.

## 8.2 Using the Docker image

If you chose to use the predefined Docker container then you can start the container using this command:

```
docker run -it --rm -p 8888:8888 jeokrohn/axl_workshop
```

The image is downloaded when the docker container is executed for the first time. Subsequent executions don’t require an image download.

After the Docker container is started you can connect to the container by pointing your web browser to http://localhost:8888

**The password to access the Jupyter server is: axl**

**Note: any changes you make to notebooks using this method are lost as soon as the container is stopped!!**

If you want to be able to modify the project files from within the Jupyter notebook you need to run this command instead to start the container:

```
docker run -it --rm -p 8888:8888 -v "$(pwd)":/home/jovyan/
jeokrohn/axl_india
```

**On Windows instead of $(pwd) you’ll need to insert the absolute path of your project directory.**

After authenticating to the Jupyter server you should see something like this:


To open a Jupyter notebook you can select any IPYNB file.

Within the notebook you can execute code cells by hitting Shift+Enter.

# 9 Accessing the Lab

The easiest option to get access to a full blown UC setup to play with is to schedule an instance of the "Cisco Preferred Architecture for Enterprise Collaboration 11.6 Lab v1" on dcloud.cisco.com.

To connect to your dCloud lab use the procedure described here: https://dcloud-cms.cisco.com/help/install_anyconnect_pc_mac and the credentials received via email in response to scheduling the lab.

The connection to the lab needs to be established before trying to send AXL requests to Unified CM.

# 10 Lab Topology

For the lab Topology, information to access and credentials pls. refer to the information available at the dCloud site.

After your VPN connection is up you can then connect to all lab systems. The main host you will need is:

UCM at 198.18.1.13
User: administrator
Password: dCloud123!

These credentials can be used to login and also to authenticate AXL requests.

# 11 AXL Introduction

From the Docker image open "01 AXL Introduction.ipynb".
In PyCharm use "01 AXL Introduction.py"

You can always also open the Jupyter notebook on GitHub in read-only mode to follow the steps.

# 12 Using zeep

From the Docker image open "02 Zeep.ipynb".
In PyCharm use "02 zeep.py"

You can always also open the Jupyter notebook on GitHub in read-only mode to follow the steps.

# 13 Solving a real world problem: Mexican numbering plan

From the Docker image open "03 Mexico numbering plan.ipynb".
In PyCharm use "03 mxnumplan.py"

You can always also open the Jupyter notebook on GitHub in read-only mode to follow the steps.