

## INSTALL GUIDE:

### How to Create a Virtual Environment with Python 3.10 on Ubuntu 18.04

Creating a virtual environment in Python is an essential practice for managing dependencies and isolating project environments. Here's a step-by-step guide to creating a virtual environment using Python 3.10 on Ubuntu 18.04.

#### 1. Install Python 3.10 (if not already installed)

Before creating a virtual environment, you need to ensure that Python 3.10 is installed on your system:

*Make sure your system is updated:*

```
sudo apt update && sudo apt upgrade
```

*Install required dependencies:*

```
sudo apt install wget build-essential libreadline-gplv2-dev libncursesw5-dev \
libssl-dev libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev libffi-dev
zlib1g-dev
```

*Download the archive from Python's official site:*

```
wget https://www.python.org/ftp/python/3.10.0/Python-3.10.0.tgz
```

*Extract the archive:*

```
tar xzf Python-3.10.0.tgz
```

*Move to extracted directory:*

```
cd Python-3.10.0
```

*Compile on your system*

```
./configure --enable-optimizations
```

*Finally install on your system, make sure you are using altinstall command since this might replace your current Python version if you don't use this command:*

```
sudo make altinstall
```

*If that is successfully installed, you can check the Python version:*

```
python3.10 -V
```

#### 2. Install venv module (if not already installed)

```
pip3.10 install virtualenv
```

#### 3. Create a Virtual Environment

Now that Python 3.10 and the venv module are installed, you can create a virtual environment.

First, navigate to the directory where you want to create your virtual environment:

```
cd /path/to/your/project
```

Use the following command to create a virtual environment. You can name the environment anything you like; in this example, it's named venv:

```
python3.10 -m venv venv
```

This command will create a directory named **venv** in your project directory containing the virtual environment.

#### 4. Activate the Virtual Environment

To start using your virtual environment, you need to activate it:

```
source venv/bin/activate
```

Once activated, your terminal prompt will change to indicate that you're now working within the virtual environment. For example:

```
(venv) yourname@yourmachine:~/path/to/your/project$
```

#### 5. Copy the project files

Now copy the project files inside the venv folder (~/.path/to/your/project).

### Install Dependencies:

With the virtual environment activated, you can now install your project's dependencies.

## NPM and NodeJS installation:

To install specific versions of Node.js and npm on Linux, you can use the Node.js version manager called **nvm** (Node Version Manager). Here are the steps to install Node.js version 17.9.1 and npm version 8.11.0:

1. `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash`
2. `source ~/.nvm/nvm.sh`
3. `nvm install 17.9.1` (Install Node.js version 17.9.1)
4. `npm install -g npm@8.11.0`
5. Check versions with:  
`node --version`  
`npm --version`

**Note:** using the nvm command, we do not alter the global dependency of node, because with nvm we simply can change the version of node in use with the command “`nvm use <version_number>`”. When you switch the node version, the npm version will automatically update as well. There's no need to worry about the npm version installed, as it won't affect the npm versions associated with other Node.js versions. Of course, if node 17.9.1 is already used by another project, you might need to reconsider the installation of npm.

Now, we need to install other dependencies. Type the following instructions into the terminal:

1. `sudo apt-get install nodejs-dev node-gyp libssl1.0-dev`
2. `npm install express`
3. `npm install body-parser`

## Python project requirements installation:

Go to the project folder and type into a terminal the following instruction:

```
pip3.10 install -r requirements.txt
```

## NPM dependencies installation:

Go to the project folder and type into a terminal the following instruction:

```
npm install
```

 (this will install the global dependencies for npm, you can see them in the `npm-root-dependencies.json` file)

```
cd pythonProject
```

 (move to the pythonProject folder)

```
npm install
```

 (this will install the local dependencies for npm, you can see them in the `npm-project-dependencies.json` file)

## USER GUIDE:

### Preliminaries:

Regarding the **start\_servers.sh** script, you need to update the paths to the files. In the provided scripts, it is assumed that the project is in the home folder (~/.HanabiBackend). You need to change this to point to the project in the venv.

In the project, inside the pythonProject folder, there is a “**properties.env**” file. You need to update this file. Particularly, you need to update the **FRONTEND\_HOST\_IP** variable with the correct IP address. Use the command “`curl ifconfig.me`” in a Linux terminal to get the IP. Then, you also need to update the **BACKEND\_HOST\_IP** variable. Use the command “`hostname -I | cut -d' ' -f1`” to get the IP.

Also, do not forget to update the environment variables regarding the furhats IPs.

### Start the project:

To start the project, first, you need to open a browser and go to the Furhat's IP address and enable the remote API (password: admin). Do this for both Furhats and keep these pages open.

Now, you need to run the start\_servers.sh script. To do this, open a terminal and locate the file; then, run the following instruction:

```
./start_servers.sh
```

**Note:** if this is the first time you execute this script, then, you'll need to give it the execution permission. To do this, simply type “`chmod +x start_server.sh`”.

Now, we need to have three different devices ready to open a web page (theoretically, you could use one single device and open, in incognito mode, three different pages).

In the **first device**, open the following address:

<http://localhost:3000/tests/lobby.html?name=Player1&room=PriscaLab&isboard=false>

**Note:** update localhost with the host IP, and always start with the first player address, do not open this after opening the second player address. Order is important.

Now, in the **second device**, open the following address:

<http://localhost:3000/tests/lobby.html?name=Player2&room=PriscaLab&isboard=false>

Again, change your host ip accordingly.

Now, you should be able to press the “**Ready**” button in the webpage. Please proceed to press “Ready” FIRST in the first player device webpage, and only then in the second player device. The first player needs to be the first to be ready.

After pressing “Ready” for both the players, we need to start the backend project.

Now, you need to go inside the venv project. To start using your virtual environment, you need to activate it:

```
source venv/bin/activate
```

Once activated, your terminal prompt will change to indicate that you're now working within the virtual environment. For example:

```
(venv) yourname@yourmachine:~/path/to/your/project$
```

Once you are inside your venv project folder, open another terminal and type (update with your path):

```
~/path/to/your/project/HanabiBackend/pythonProject python3.10 main.py
```

Once executed this instruction, the program will ask you to type several information. Please proceed until no other information is asked and wait until the console log stops printing any information (so you need to wait for the system to load).

Now, with the **third device**, open the following address:

<http://localhost:3000/tests/flex-reactive.html?name=board&room=PriscaLab&isboard=true&persistentId=board>

This device represents the game board, so it's not a real player.

Now, the game will start according to the game mode you chose (Tom+Human or Tom+Tom).