06

Do as I did: virtualization best practices

Creating and managing virtual environments in Python is a fundamental practice for isolating project dependencies and ensuring consistency in development. Let's go through the step-by-step process to create a new virtual environment and the corresponding commands to activate and deactivate it on Mac, Windows, and Linux operating systems.

Instructor's opinion

```
# Criar um novo ambiente virtual
python -m venv ./venv

# Criar o ambiente virtual (Mac)
python3 -m venv ./venv

# Ativar o ambiente virtual (Mac/Linu
source nome_do_ambiente/bin/activate

# Ativar o ambiente virtual (Windows)
.\nome_do_ambiente\Scripts\activate

# Desativar o ambiente virtual
```

```
deactivate
```

COPY CODE

Additionally, it is highly recommended to create a requirements.txt file to list the project dependencies. This makes it easier to replicate the environment across different machines and environments. The requirements.txt file can be generated with the following command:

```
pip freeze > requirements.txt
```

COPY CODE

The file `requirements.txt` is vital for documenting and sharing project dependencies. By explicitly listing the versions of the libraries used, we ensure that other developers or systems can reproduce the exact same development environment. This avoids conflicts and ensures that the project runs consistently across different contexts. Virtualization in Python development is an essential practice. By using virtual environments and the requirements.txt file, you ensure the cohesion of the development environment, facilitate collaboration between team members, and promote project portability.

Remember to `criar` and `ativar` your virtual environment before starting any development,

ensuring a stable and consistent coding experience.