1. TP1 Utilisation et création d'API (4H)

1.1 Présentation

Ce TP a pour objectif de vous faire découvrir l'utilisation d'API REST avec Curl et Python puis la création d'API avec FastAPI.

1.2 Travail à rendre

Faire un compte rendu par binôme de toutes les actions réalisées (commandes/captures) en suivant les recommandations suivantes

Vous devez utiliser le fichier au format Markdown (.md) de ce sujet (dossier /CR) et l'éditer avec VSCode afin d'y mettre vos réponses.

```
Pour le compte rendu sous VSCode :
-- Changez le thème de VSCode : File -> Preferences -> Color Theme --> Light+
-- Renommez le dossier CR en CR_ETU1_ETU2 qui contient le fichier .md et le
dossier images
-- Renommez le fichier index.md en CR_ETU1_ETU2
-- Mettez vos réponses en **gras** après un > suivi d'un espace :
> **Ma réponse**
>
> **Suite de ma réponse**
>
-- Insérez les captures écran ainsi  ![capture1](images/capture1.png)
-- Prévisualisez le résultat : se positioner dans fenêtre du fichier .md  puis
CTRL+SHIFT+V
   ou bien pour avoir le texte et la prévisualisation côte à côte  : CTRL+SHIFT+P
et sélectionner Markdown : Open preview to the side
```

> **Ma réponse**
>
> **Suite de ma réponse**

Capture 0 : voici comment insérer une capture écran

2. QCM

Commencez par faire **individuellement** le QCM sous Moodle en utilisant les PC Gn et Dn sous Windows.

- avec un adressage IP par DHCP
- en configurant sous Firefox le proxy 10.254.0.254:3128 pour tous les protocoles

3. Configuration du Proxy

- Démarrez `Sn` sous Windows avec
  - avec un adressage IP par DHCP (vous devez obtenir une @IP dans le réseau `10.254.200.YYY/16`)
  - en configurant sous Firefox le proxy `10.254.0.254:3128` pour tous les protocoles
- Récupérez ce sujet sur l'ENT
- Depuis `Sn`, récupérez la VM fournie avec la commande suivante:

```
pscp -r root@10.254.XXX.YYY:/VMAPI/* D:\Vm
```

> Dans le dossier D:\Vm, lancez à la VM à partir du fichier `.vmx`

- Vérfiez que la VM dispose bien d'une adresse IP dans le réseau `10.254.200.YYY/16`
- Créez ou utilisez le script `script_proxy.sh` pour accéder à Internet depuis un terminal en passant proxy

```
#!/bin/bash
#author : P. Durand 2022
# script_proxy_docker.sh à lancer avec source et pas en ./
# 10.254.0.254 en salle B10-002 ...
# 172.16.0.1 en salle I014
# Au lieu de saisir votre mot de passe dans le fichier, utilisez la commande read

login='testeuras1r'
#Demandez le mot de passe de la semaine
mdp='aaaaaaa'

#vous pouvez échapper un caractère spécial de votre mot de passe avec \

export HTTP_PROXY="http://$login:$mdp@10.254.0.254:3128"
export HTTPS_PROXY="http://$login:$mdp@10.254.0.254:3128"
export http_proxy="http://$login:$mdp@10.254.0.254:3128"
export https_proxy="http://$login:$mdp@10.254.0.254:3128"
```

Ce script sera à lancer dans chaque nouveau terminal qui doit se connecter à Internet

Testez l'accès à l'extérieur du réseau depuis un terminal

- Lancez le script afin que les variables d'environnement soient accessibles depuis le terminal où il est lancé `source script_proxy.sh`
- Vérifiez la modification des variables d'environnement avec `export`
- Testez l'accès Internet (HTTP) avec une commande de votre choix `wget` ou `curl`!

VISA 0 Faites valider l'accès HTTP depuis un terminal par votre enseignant

4. EX1-Utilisation de CURL pour accèder à une API

On va utiliser ici l'API REST factice https://fakestoreapi.com/ qui propose plusieurs méthodes:

- GET/products
- GET/products/1
- GET/products/categories
- GET/products/category/jewelery
- GET/carts?userId=1
- GET/products?limit=5
- POST/products
- PUT/products/1
- PATCH/products/1
- DELETE/products/1

Donnez et testez les commandes CURL acceptant une réponse au format json qui permettent de:

1. lister tous les produits,
2. lister le produit N°5,
3. lister les catégories,
4. créer un produit de votre choix dans la catégorie electronic,
5. modifier un produit de votre choix.

Consignez les résultats

- **Refaites le même travail avec le plugin RESTED de Firefox**

> Variante possible avec les API suivantes en cas de problème d'accès

- https://jsonplaceholder.typicode.com/
- https://dummyjson.com/
- https://fakeapi.platzi.com/

VISA 1 Faites valider par votre enseignant

5. EX2-Utilisation des classes Python d'accès HTTP

5.1 Introduction

La bibliothèque Python `requests` permet d'interagir avec les services Web.

5.2 Création d'un environnement Python virtuel

- Créez un dossier `/home/etudiant/Nom1_Nom2`
- Dans ce dossier, créez un environnement virtuel nommé `vNOM`
- Dans ce dossier, créez un dossier `TP1`
- Activez l'environnement virtuel
- Mettez à jour la base dépot Python
- Installez la bibliothèque `requests` en utilisant la commande pip comme ceci : `pip3.9 install requests`

5.3 Travail à réaliser

1. Créez dans le dossier TP1 le fichier `EX2.py` et testez le programme Python avec `python` suivant qui interroge l'API REST suivante:

```
import requests
BASE_URL = 'https://fakestoreapi.com'
response = requests.get(f"{BASE_URL}/products")
print(response.status_code)
print(response.json())
```

2. listez le produit N°10
3. Testez la limitation à l'affichage pour 5 produits avec des paramètres passés à la requête

```
query_params = {
    "limit": 5
}
response = requests.get(f"{BASE_URL}/products", params=query_params)
```

4. Compléter le programme afin de n'afficher que la `description` des produits et la valeur `count` des 5 produits précédents
5. Créez un produit de votre choix dans la catégorie electronic, le produit crée doit être déclaré avant au format `JSON`

```
response = requests.post(f"{BASE_URL}/products", json=new_product)
```

6. Vérifiez l'insertion du produit en interrogeant à nouveau l'API pour ce produit en particulier (https://fakestoreapi.com/)

7. Modifiez un produit de votre choix

VISA 2 Faites valider par votre enseignant

6. EX3-Création d'une API avec FastAPI

6.1 **Préparer l'environnement**

Pour installer FastAPI, exécutez la commande suivante dans l'environnement virtuel Python :

```
pip install fastapi
pip install "uvicorn[standard]"

#nous aurons besoin ensuite des modules suivants

pip install sqlite3worker
pip install PyJWT
pip install requests
```

6.2 Version N°1 utilisant des requêtes SQL

Cette API va manipuler les données de la base de données fournies `IUT.sqlite` (déjà utilisée en première année).

Cette API devra proposer un CRUD sur la table `etudiants`

Voici un rappel du schéma relationnel:

```
CREATE TABLE `etudiants` (
    `mail`  VARCHAR NOT NULL DEFAULT (null),
    `mdp`   VARCHAR DEFAULT (null),
    `adresse`   VARCHAR DEFAULT (null),
    `insee` INTEGER DEFAULT (null),
    PRIMARY KEY(`mail`)
);

CREATE TABLE `villes` (
    `commune`   varchar ( 30 ) NOT NULL DEFAULT (''),
    `cp`    varchar ( 5 ) NOT NULL DEFAULT (''),
    `dept`  varchar ( 30 ) NOT NULL DEFAULT (''),
    `insee` varchar ( 5 ) DEFAULT (null),
    PRIMARY KEY(`insee`)
);
```

- Créez un dossier EX3-Api-V1 et mettez en place l'arborescence suivante

```
├── EX3-Api-V1
    ├── main.py
    └── bdd
        └──IUT.sqlite
```

- Complétez le code source suivant pour créez l'API en suivant les commentaires

```
import os
from fastapi import FastAPI
from pydantic import BaseModel
import sqlite3

#Créez un objet FastAPI

# Créez une classe Etudiant qui hérite de BaseModel

# Créez une classe Ville qui hérite de BaseModel

# Créez une fonction qui permet la connection à la BDD

#Route (endpoint) avec la méthode GET pour obtenir tous les etudiants en faisant
une jointure

#Route (endpoint) avec la méthode POST pour ajouter un étudiant

#Route (endpoint) avec la méthode DELETE pour supprimer un etudiant

#Route (endpoint) avec la méthode PUT pour modifier un etudiant
```

- Lancez votre API avec `uvicorn`
- Testez progressivement votre API avec l'outil FastAPI intégré avec l'URL `http://127.0.0.1:8000/docs`
- Capturez et consignez tous les résultats.

VISA 3 Faites valider par votre enseignant

6.3 Version N°2 utilisant l'ORM SQLModel

- Créez un dossier EX3-Api-V2 et mettez en place l'arborescence suivante:

```
├── EX3-Api-V2
    ├── main.py
    └── bdd
        └──IUT.sqlite
```

- Modifiez l'API précédente afin d'utiliser l'ORM SQLModel pour créer le CRUD de la table `Etudiants`
- Lancez votre API avec `uvicorn`
- Testez progressivement votre API avec l'outil FastAPI intégré avec l'URL `http://127.0.0.1:8000/docs`
- Capturez et consignez tous les résultats.

VISA 4 Faites valider par votre enseignant