vendredi 28 novembre 2025

INFORMATIQUE - 3ième année

R5.Devcloud.07

Développement de microservices
DataClass, Pydantic, FastAPI, Docker

Python 3.14

(Fascicule N° 2/2)

Prérequis:

M1105 :	Base des systèmes d'exploitation (OS)	Scripting Shell (Windows, Bash: Linux)
M2102 :	Administration système	Scripting Shell
M1207 :	Bases de la programmation	Python
M2207, M309, M308 :	Programmation Orientée Objet	Java
M1106 et M2105 :	Développement WEB	HTML, CSS, JS, PHP

http://www.python.org

Bibliothèque http://docs.python.org/py3k/library/index.html

https://docs.python.org/3.14

jean-claude.nunes@univ-rennes1.fr

- 6 - FastAPI

6.1 Créer une application FastAPI

https://docs.python.org/3.14/

https://doc?????

API : interfaces de programmation d'applications

Les API   Application Programming Interface

- ☞ sont l'épine dorsale de l'architecture moderne
    - ☞ applications modulaires et découplées.

- ☞ permet de créer des applications rapidement et facilement, ce qui vous permet de les maintenir et de les mettre à jour aisément.

- ☞ permettent à différentes applications de partager des données et de travailler ensemble
    - ☞ économiser du temps et des efforts.

- ☞ de nombreux frameworks différents pour construire des API en Python.
    - ☞ en Python : *Django*, *Flask* et *FastAPI*.

FastAPI

FastAPI

- **un framework web performant en Python**
  - **création rapide et efficace d'applications modernes**
    - **API prête à la production,**
  - **performance comparable à celle de *Go* et *Node.js*.**

- **Facile à apprendre et à coder.**
  - **création d'une *API RESTful* prête à être déployée en quelques lignes de code.**
    - ***API RESTful* : interface que deux systèmes informatiques utilisent pour échanger des informations en toute sécurité sur Internet.**

- **construit au-dessus du serveur web *Starlette***
  - ***Uvicorn*: serveur ASGI**
    - ***Starlette*: microframework web**
      - ***FastAPI***

Hiérarchie

- **documentation complète :**
  - **utilise les normes de documentation *OpenAPI*,**
  - **génération dynamique de la documentation interactive.**

- **moins de bugs**
  - **validation automatique des données,**
  - **la gestion des erreurs.**

FastAPI

Uvicorn

☞ **nécessite un serveur web local pour tester les API.**

☞ ***Uvicorn* est un serveur web *ASGI : Asynchronous Server Gateway Interface*.**

- ☞ **basé sur *uvloop* et *httptools***
- ☞ **pour tester et exécuter vos applications FastAPI**

FastAPI

Préparer l’environnement de développement

☞ Installer *pip3*

```
$ sudo apt install python3-pip
```

Le pip de Python 3 est souvent appelé *pip3*

```
$ pip3 –version
```

```
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
```

☞ créer un environnement virtuel

```
$ python3 -m venv env
```

☞ activer l’environnement virtuel

```
# On Unix or MacOS (bash shell):
/path/to/venv/bin/activate

# On Unix or MacOS (csh shell):
/path/to/venv/bin/activate.csh

# On Unix or MacOS (fish shell):
/path/to/venv/bin/activate.fish

# On Windows (command prompt):
pathtovenvScriptsactivate.bat

# On Windows (PowerShell):
pathtovenvScriptsActivate.ps1
```

☞ Installer *FastAPI*

```
$ pip3 install fastapi
```

☞ Installer le serveur *uvicorn* (ASGI), basé sur uvloop et httptools

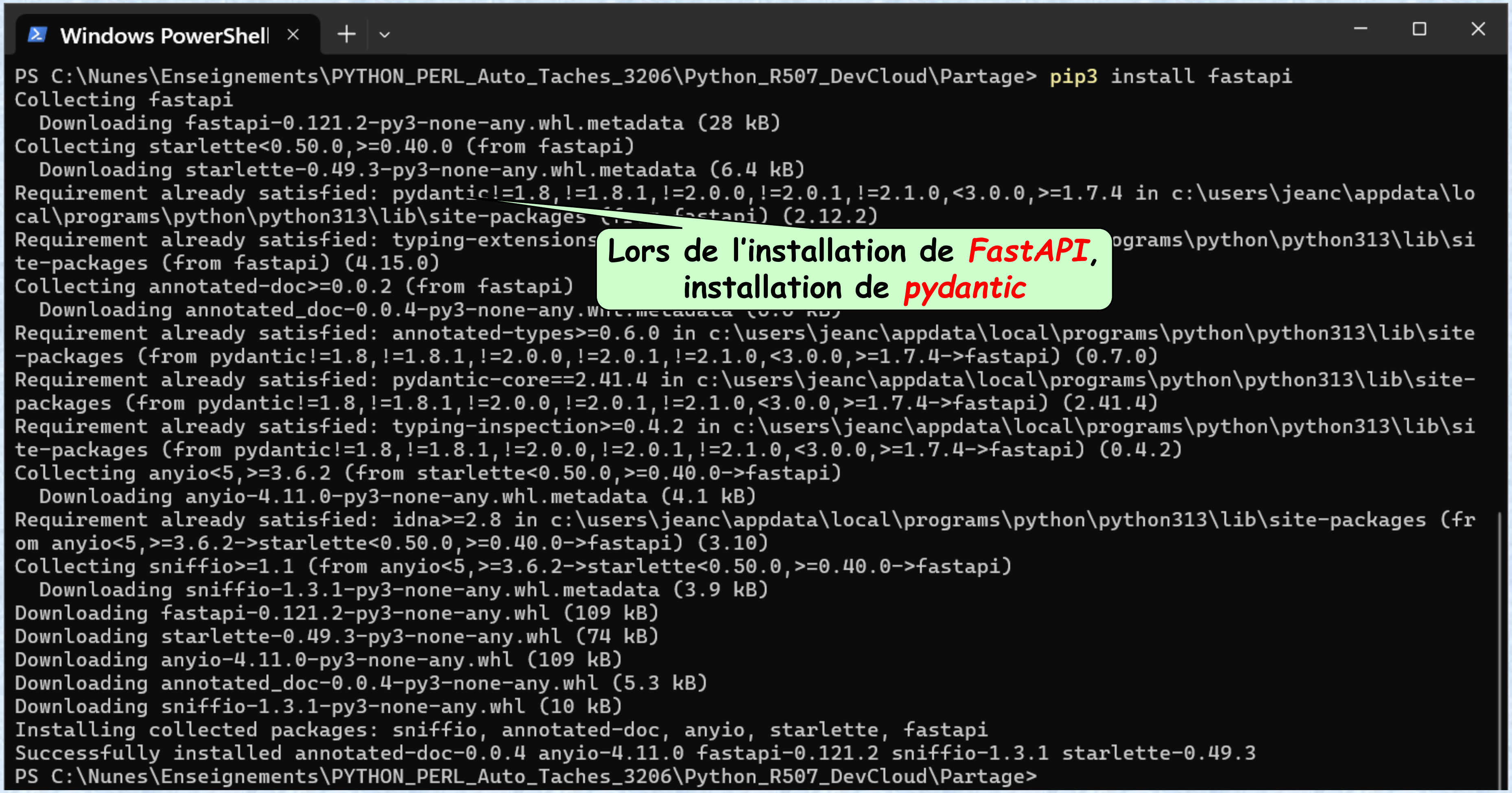
```
$ pip3 install uvicorn
```

FastAPI

Préparer l'environnement de développement

☞ Installer *FastAPI*

$ pip3 install fastapi



```
PS C:\Nunes\Enseignements\PYTHON_PERL_Auto_Taches_3206\Python_R507_DevCloud\Partage> pip3 install fastapi
Collecting fastapi
  Downloading fastapi-0.121.2-py3-none-any.whl.metadata (28 kB)
Collecting starlette<0.50.0,>=0.40.0 (from fastapi)
  Downloading starlette-0.49.3-py3-none-any.whl.metadata (6.4 kB)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4 in c:\users\jeanc\appdata\lo
cal\programs\python\python313\lib\site-packages (from fastapi) (2.12.2)
Requirement already satisfied: typing-extensions                                          ograms\python\python313\lib\si
te-packages (from fastapi) (4.15.0)
Collecting annotated-doc>=0.0.2 (from fastapi)
  Downloading annotated_doc-0.0.4-py3-none-any.whl.metadata
Requirement already satisfied: annotated-types>=0.6.0 in c:\users\jeanc\appdata\local\programs\python\python313\lib\site
-packages (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi) (0.7.0)
Requirement already satisfied: pydantic-core==2.41.4 in c:\users\jeanc\appdata\local\programs\python\python313\lib\site-
packages (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi) (2.41.4)
Requirement already satisfied: typing-inspection>=0.4.2 in c:\users\jeanc\appdata\local\programs\python\python313\lib\si
te-packages (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi) (0.4.2)
Collecting anyio<5,>=3.6.2 (from starlette<0.50.0,>=0.40.0->fastapi)
  Downloading anyio-4.11.0-py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: idna>=2.8 in c:\users\jeanc\appdata\local\programs\python\python313\lib\site-packages (fr
om anyio<5,>=3.6.2->starlette<0.50.0,>=0.40.0->fastapi) (3.10)
Collecting sniffio>=1.1 (from anyio<5,>=3.6.2->starlette<0.50.0,>=0.40.0->fastapi)
  Downloading sniffio-1.3.1-py3-none-any.whl.metadata (3.9 kB)
Downloading fastapi-0.121.2-py3-none-any.whl (109 kB)
Downloading starlette-0.49.3-py3-none-any.whl (74 kB)
Downloading anyio-4.11.0-py3-none-any.whl (109 kB)
Downloading annotated_doc-0.0.4-py3-none-any.whl (5.3 kB)
Downloading sniffio-1.3.1-py3-none-any.whl (10 kB)
Installing collected packages: sniffio, annotated-doc, anyio, starlette, fastapi
Successfully installed annotated-doc-0.0.4 anyio-4.11.0 fastapi-0.121.2 sniffio-1.3.1 starlette-0.49.3
PS C:\Nunes\Enseignements\PYTHON_PERL_Auto_Taches_3206\Python_R507_DevCloud\Partage>
```

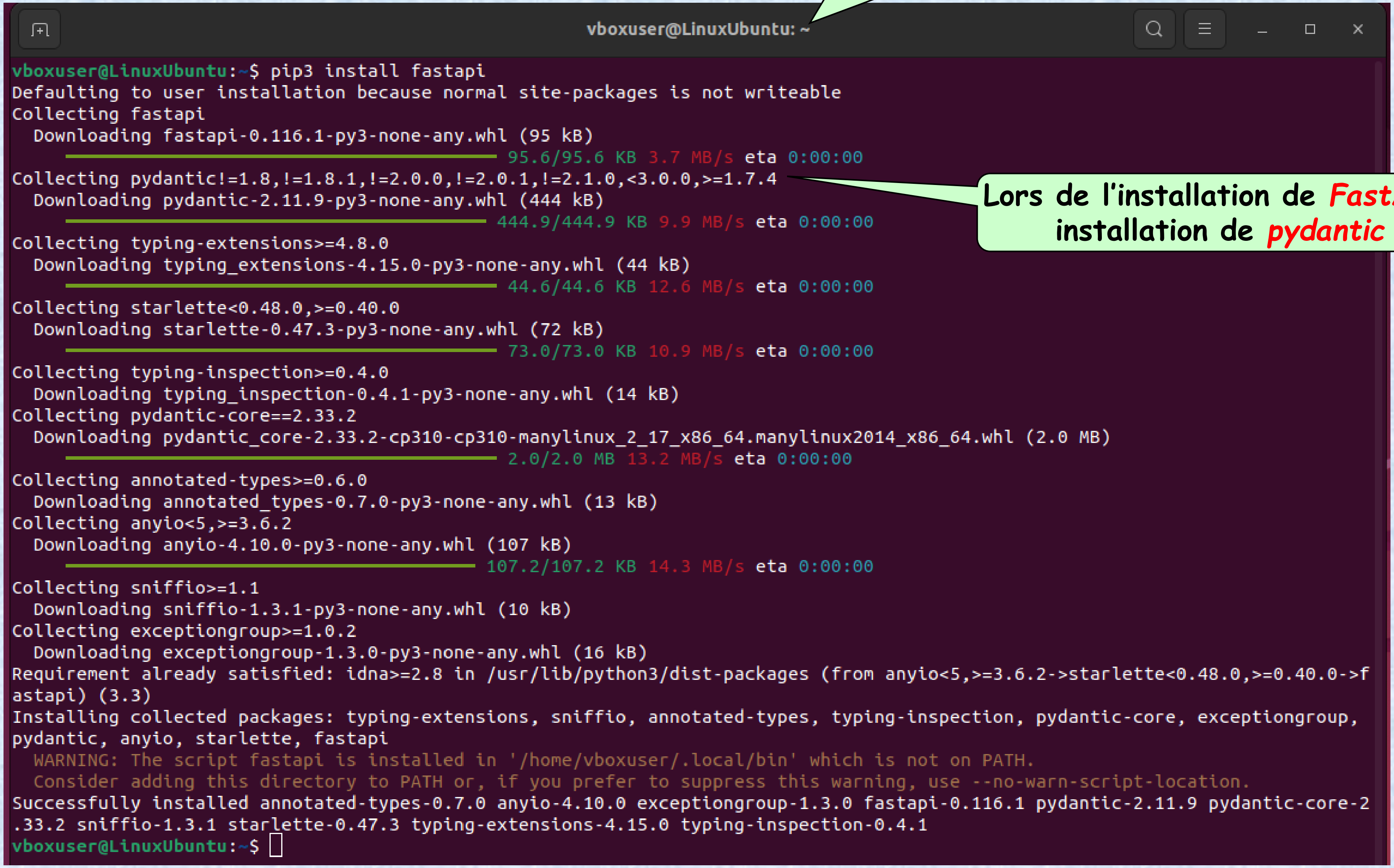
Lors de l'installation de *FastAPI*, installation de *pydantic*

FastAPI

Préparer l’environnement de développement

☞ Installer *FastAPI*

$ pip3 install fastapi


sous Linux
vboxuser@LinuxUbuntu: ~
Lors de l’installation de *FastAPI*, installation de *pydantic*

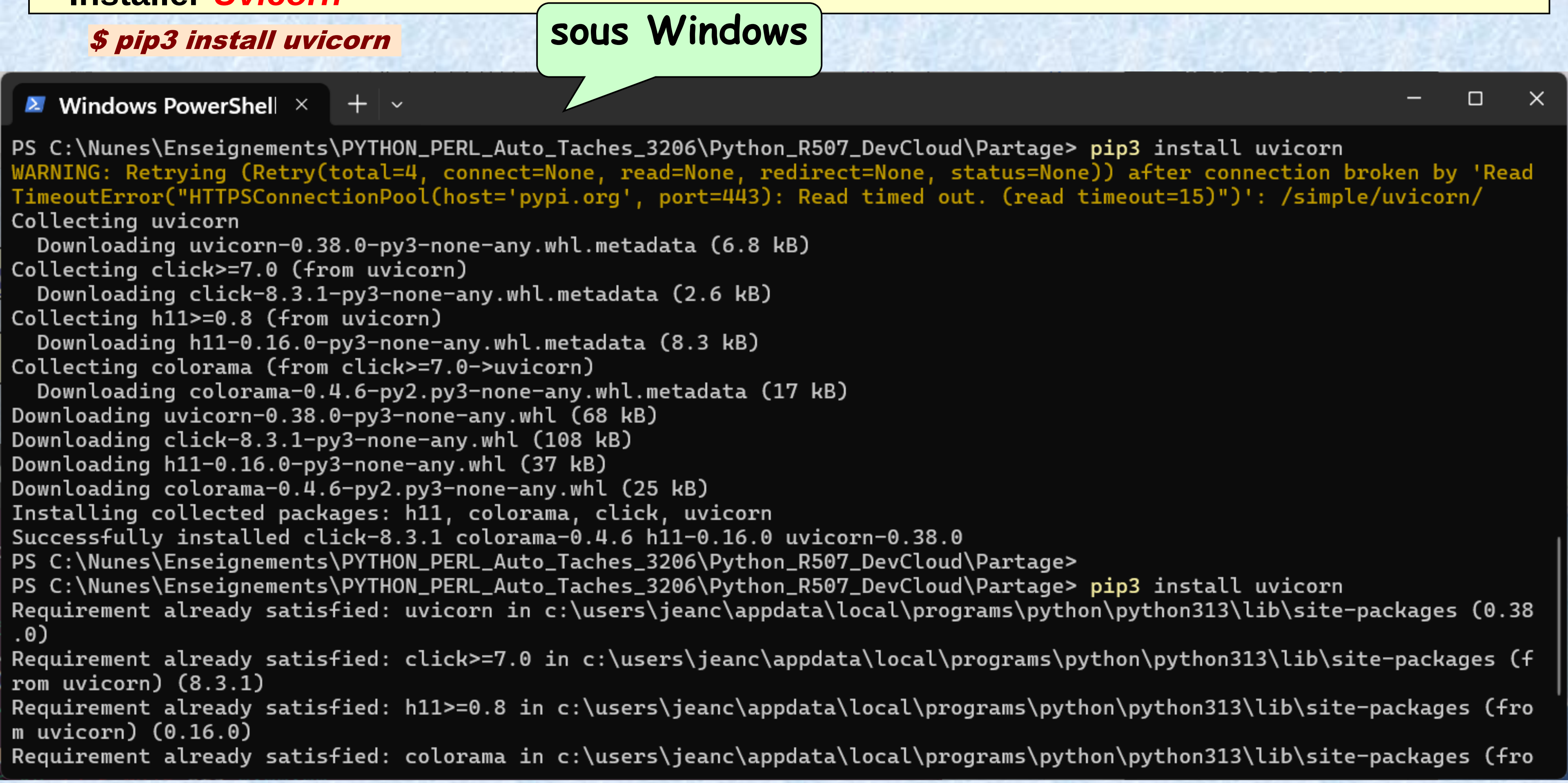
```
vboxuser@LinuxUbuntu:~$ pip3 install fastapi
Defaulting to user installation because normal site-packages is not writeable
Collecting fastapi
  Downloading fastapi-0.116.1-py3-none-any.whl (95 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 95.6/95.6 KB 3.7 MB/s eta 0:00:00
Collecting pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4
  Downloading pydantic-2.11.9-py3-none-any.whl (444 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 444.9/444.9 KB 9.9 MB/s eta 0:00:00
Collecting typing-extensions>=4.8.0
  Downloading typing_extensions-4.15.0-py3-none-any.whl (44 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.6/44.6 KB 12.6 MB/s eta 0:00:00
Collecting starlette<0.48.0,>=0.40.0
  Downloading starlette-0.47.3-py3-none-any.whl (72 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 73.0/73.0 KB 10.9 MB/s eta 0:00:00
Collecting typing-inspection>=0.4.0
  Downloading typing_inspection-0.4.1-py3-none-any.whl (14 kB)
Collecting pydantic-core==2.33.2
  Downloading pydantic_core-2.33.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 13.2 MB/s eta 0:00:00
Collecting annotated-types>=0.6.0
  Downloading annotated_types-0.7.0-py3-none-any.whl (13 kB)
Collecting anyio<5,>=3.6.2
  Downloading anyio-4.10.0-py3-none-any.whl (107 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 107.2/107.2 KB 14.3 MB/s eta 0:00:00
Collecting sniffio>=1.1
  Downloading sniffio-1.3.1-py3-none-any.whl (10 kB)
Collecting exceptiongroup>=1.0.2
  Downloading exceptiongroup-1.3.0-py3-none-any.whl (16 kB)
Requirement already satisfied: idna>=2.8 in /usr/lib/python3/dist-packages (from anyio<5,>=3.6.2->starlette<0.48.0,>=0.40.0->f
astapi) (3.3)
Installing collected packages: typing-extensions, sniffio, annotated-types, typing-inspection, pydantic-core, exceptiongroup,
pydantic, anyio, starlette, fastapi
  WARNING: The script fastapi is installed in '/home/vboxuser/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed annotated-types-0.7.0 anyio-4.10.0 exceptiongroup-1.3.0 fastapi-0.116.1 pydantic-2.11.9 pydantic-core-2
.33.2 sniffio-1.3.1 starlette-0.47.3 typing-extensions-4.15.0 typing-inspection-0.4.1
vboxuser@LinuxUbuntu:~$ 
```

FastAPI

Préparer l'environnement de développement

☞ Installer *Uvicorn*

$ pip3 install uvicorn



```
PS C:\Nunes\Enseignements\PYTHON_PERL_Auto_Taches_3206\Python_R507_DevCloud\Partage> pip3 install uvicorn
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'Read
TimeoutError("HTTPSConnectionPool(host='pypi.org', port=443): Read timed out. (read timeout=15)")': /simple/uvicorn/
Collecting uvicorn
  Downloading uvicorn-0.38.0-py3-none-any.whl.metadata (6.8 kB)
Collecting click>=7.0 (from uvicorn)
  Downloading click-8.3.1-py3-none-any.whl.metadata (2.6 kB)
Collecting h11>=0.8 (from uvicorn)
  Downloading h11-0.16.0-py3-none-any.whl.metadata (8.3 kB)
Collecting colorama (from click>=7.0->uvicorn)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Downloading uvicorn-0.38.0-py3-none-any.whl (68 kB)
Downloading click-8.3.1-py3-none-any.whl (108 kB)
Downloading h11-0.16.0-py3-none-any.whl (37 kB)
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: h11, colorama, click, uvicorn
Successfully installed click-8.3.1 colorama-0.4.6 h11-0.16.0 uvicorn-0.38.0
PS C:\Nunes\Enseignements\PYTHON_PERL_Auto_Taches_3206\Python_R507_DevCloud\Partage>
PS C:\Nunes\Enseignements\PYTHON_PERL_Auto_Taches_3206\Python_R507_DevCloud\Partage> pip3 install uvicorn
Requirement already satisfied: uvicorn in c:\users\jeanc\appdata\local\programs\python\python313\lib\site-packages (0.38
.0)
Requirement already satisfied: click>=7.0 in c:\users\jeanc\appdata\local\programs\python\python313\lib\site-packages (f
rom uvicorn) (8.3.1)
Requirement already satisfied: h11>=0.8 in c:\users\jeanc\appdata\local\programs\python\python313\lib\site-packages (fro
m uvicorn) (0.16.0)
Requirement already satisfied: colorama in c:\users\jeanc\appdata\local\programs\python\python313\lib\site-packages (fro
```

API Rest

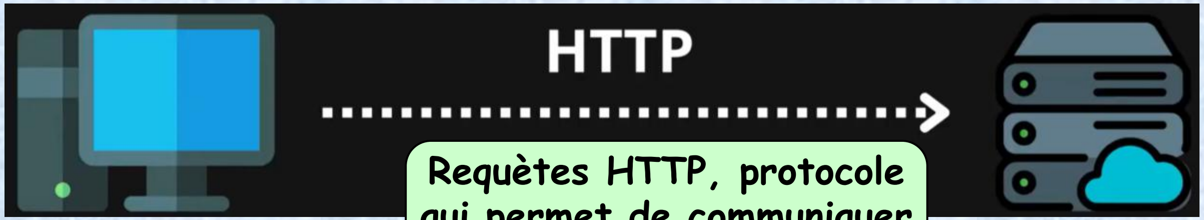
API Rest

☞ fonctionne avec des requêtes HTTP,

☞ colonne vertébrale de l'automatisation des systèmes modernes.

☞ permettent aux applications, scripts et plateformes de communiquer de manière fluide sans intervention humaine.

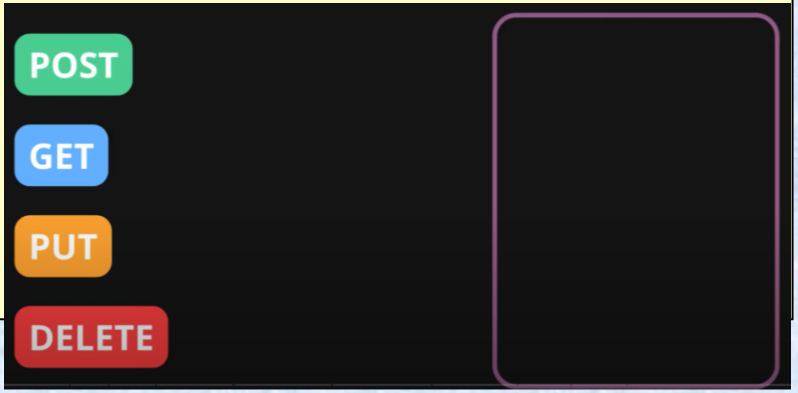
☞ rendent possible l'orchestration automatisée dans les environnements *DevOps* et *Cloud*.



Requètes HTTP, protocole qui permet de communiquer avec le serveur

Types de requêtes HTTP

☞ *GET*	Récupérer une ressource
☞ *POST*	Créer une nouvelle ressource
☞ *PUT*	Mettre à jour une ressource
☞ *DELETE*	Supprimer une ressource

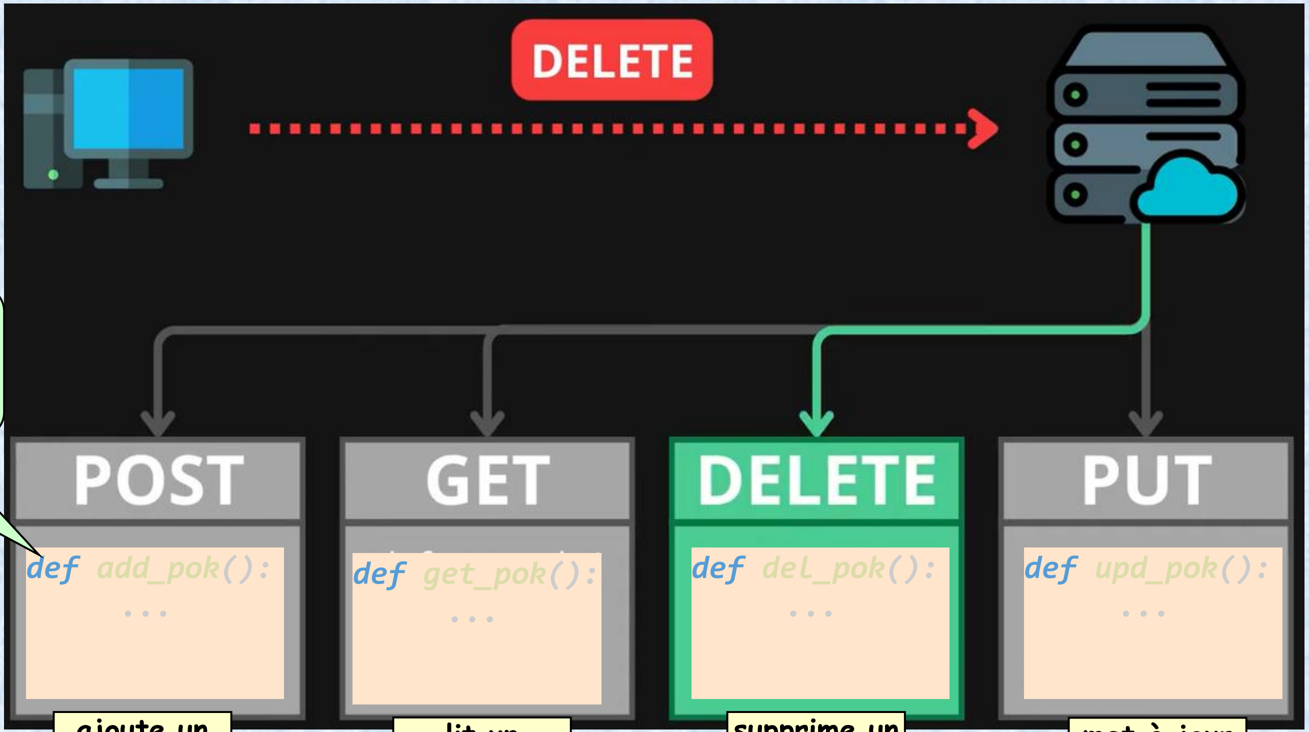


API Rest

API Rest

fonctionne avec des requêtes HTTP:

- ☞ ***GET*** : récupère les données du serveur.
- ☞ ***POST*** : crée une nouvelle ressource ou soumet des données.
- ☞ ***PUT*** : met à jour ou remplace une ressource existante.
- ☞ ***DELETE*** : supprime une ressource.



Définition de fonctions associées à chaque type de requête

ajoute un nouveau pokemon

lit un pokemon

supprime un pokemon

met à jour un pokemon

Exemple avec des pokemons

FastAPI **GET** **6.1**

- **Requète GET**

☞ **Premier programme *main.py* affichant Hello World !**

```
from fastapi import FastAPI

app = FastAPI()

@app.get("")
def read_root():
   return {"Hello": "World"}
```

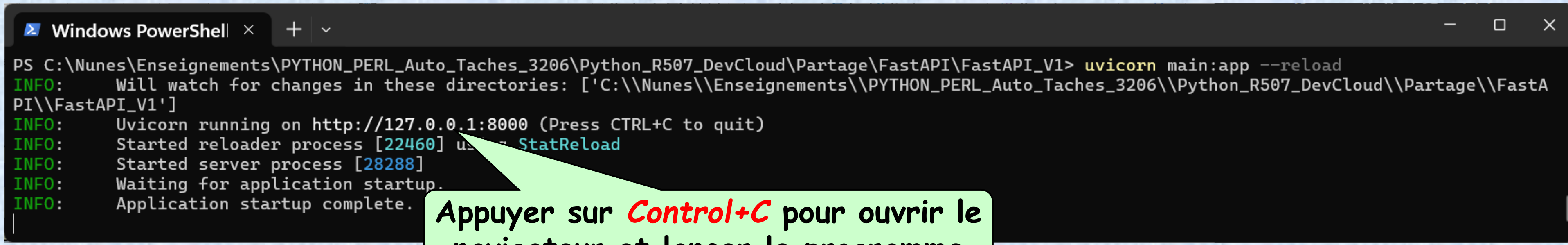
Exécution de FastAPI

main.py

1/4

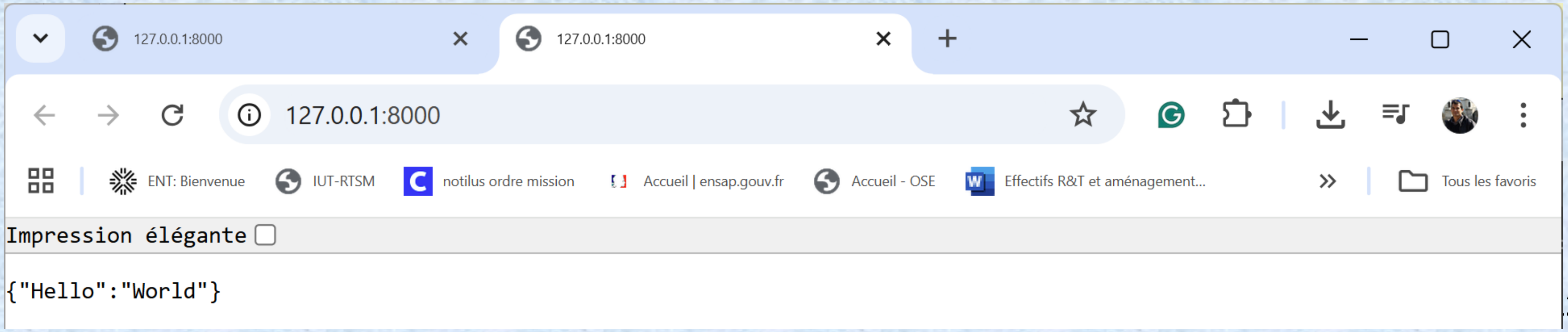
☞ **Exécuter votre programme *main.py* avec *Uvicorn***

```
$ uvicorn main:app --reload
```



```
PS C:\Nunes\Enseignements\PYTHON_PERL_Auto_Taches_3206\Python_R507_DevCloud\Partage\FastAPI\FastAPI_V1> uvicorn main:app --reload
INFO:     Will watch for changes in these directories: ['C:\\Nunes\\Enseignements\\PYTHON_PERL_Auto_Taches_3206\\Python_R507_DevCloud\\Partage\\FastA
PI\\FastAPI_V1']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [22460] using StatReload
INFO:     Started server process [28288]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

Appuyer sur *Control+C* pour ouvrir le navigateur et lancer le programme



127.0.0.1:8000

Impression élégante ☐

```
{"Hello":"World"}
```

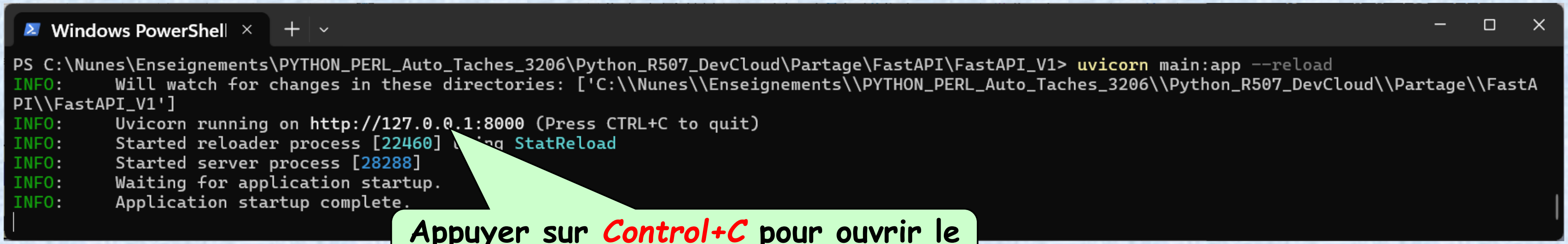
GET

- Requète GET

☞ Exécuter votre programme *main.py* avec *Uvicorn*

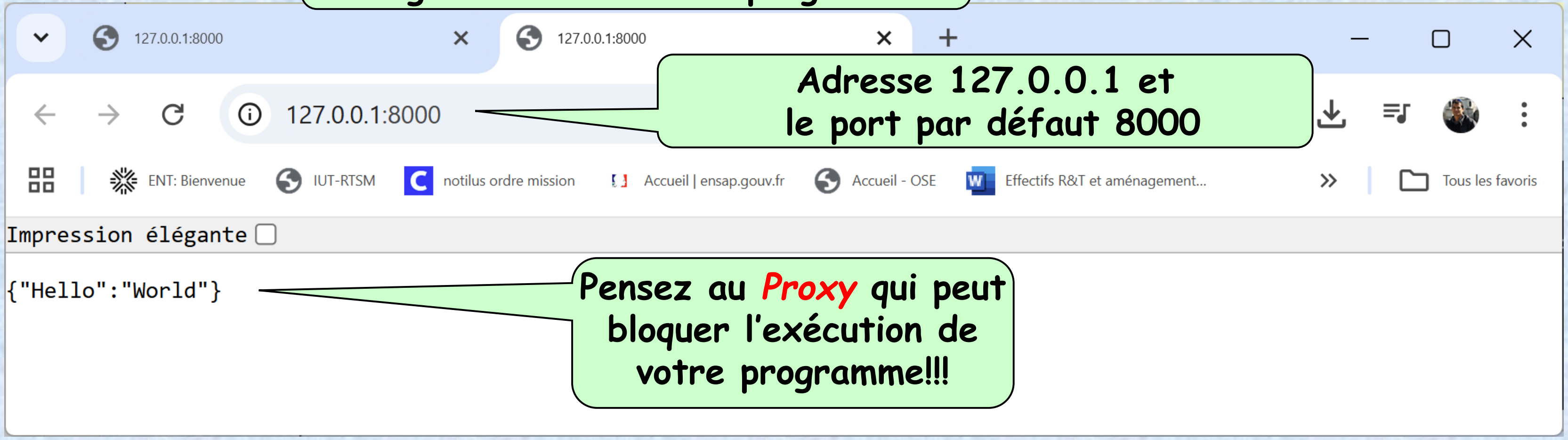
$ uvicorn main:app --reload

2/4



```
PS C:\Nunes\Enseignements\PYTHON_PERL_Auto_Taches_3206\Python_R507_DevCloud\Partage\FastAPI\FastAPI_V1> uvicorn main:app --reload
INFO:     Will watch for changes in these directories: ['C:\\Nunes\\Enseignements\\PYTHON_PERL_Auto_Taches_3206\\Python_R507_DevCloud\\Partage\\FastA
PI\\FastAPI_V1']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [22460] using StatReload
INFO:     Started server process [28288]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

Appuyer sur *Control+C* pour ouvrir le navigateur et lancer le programme

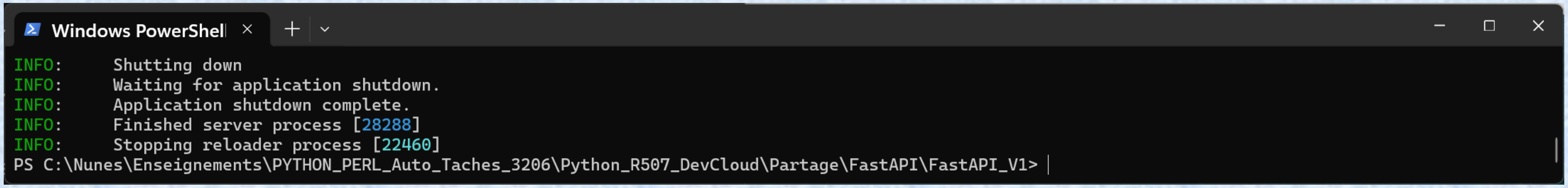


```
{"Hello":"World"}
```

Adresse 127.0.0.1 et le port par défaut 8000

Pensez au *Proxy* qui peut bloquer l'exécution de votre programme!!!

☞ *Control+C* pour stopper le programme



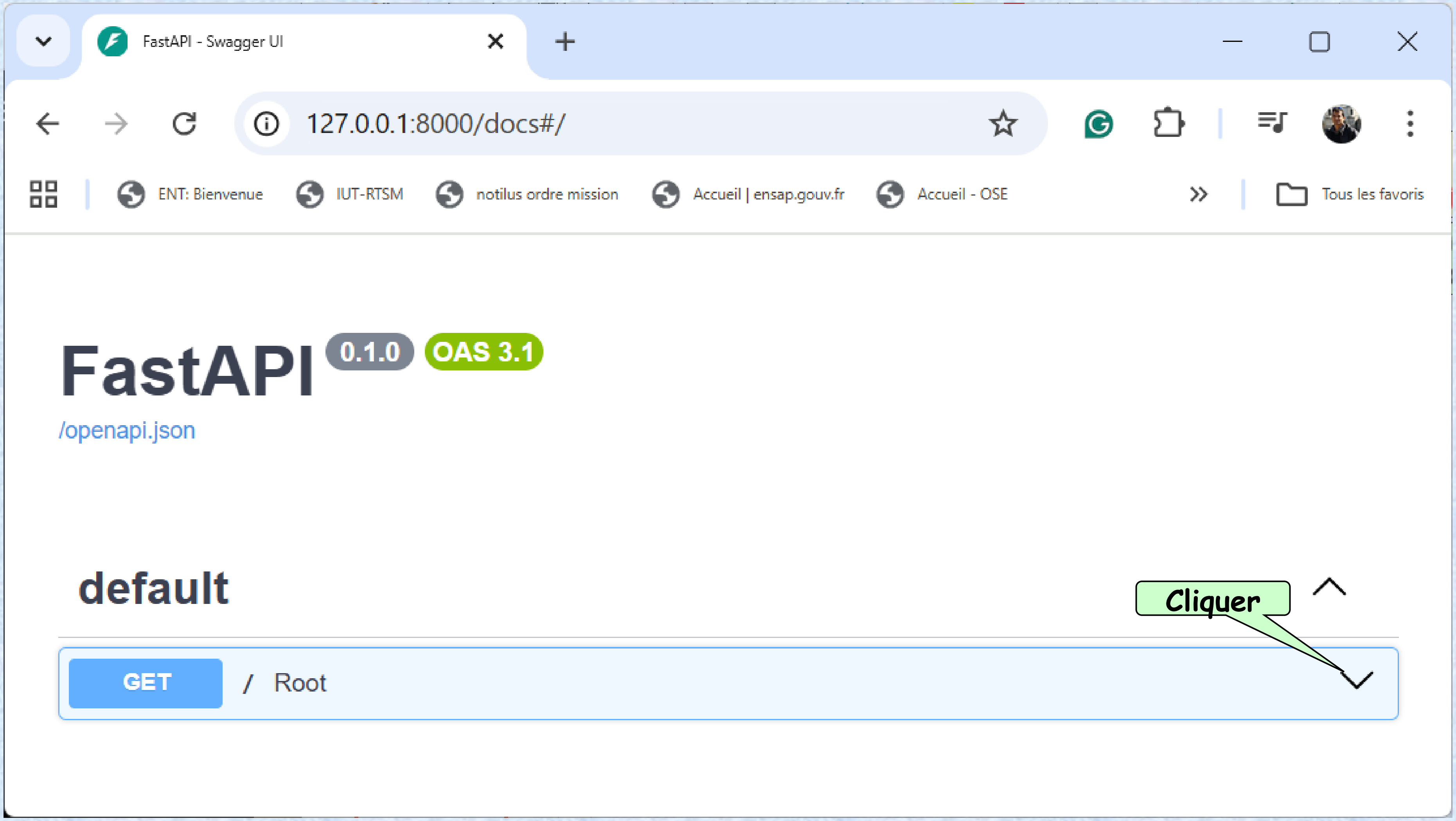
```
INFO:     Shutting down
INFO:     Waiting for application shutdown.
INFO:     Application shutdown complete.
INFO:     Finished server process [28288]
INFO:     Stopping reloader process [22460]
PS C:\Nunes\Enseignements\PYTHON_PERL_Auto_Taches_3206\Python_R507_DevCloud\Partage\FastAPI\FastAPI_V1>
```

- Interactive documentation

☞ Exécuter la doc de votre programme *main.py* avec *Uvicorn*

3/4

http://127.0.0.1:8000/docs#/ — Adresse de la doc interactive

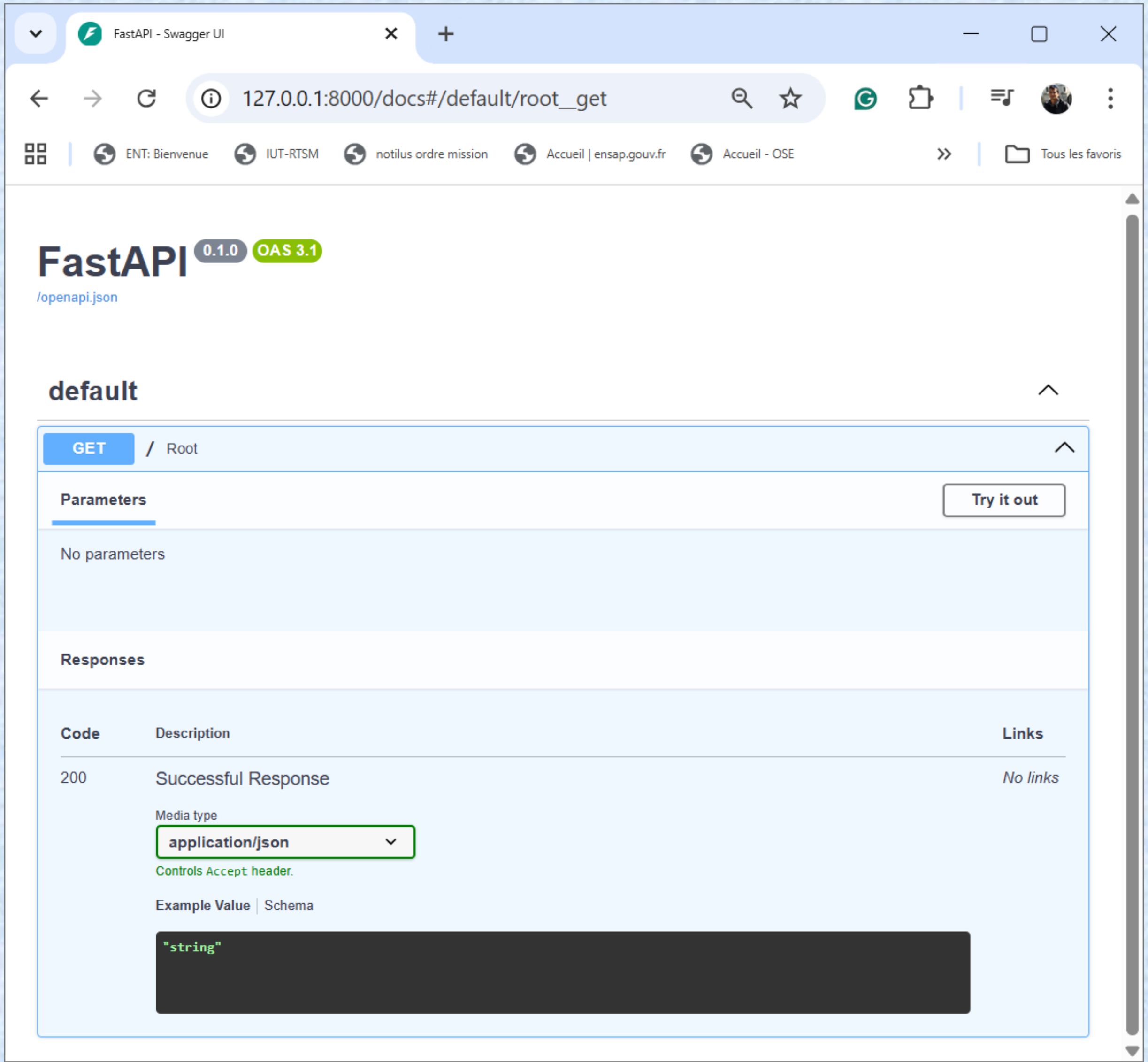


FastAPI **GET** **6.1**

- **Interactive documentation**

☞ **Exécuter la doc de votre programme *main.py* avec *Uvicorn*** **4/4**

http://127.0.0.1:8000/docs#/ — Adresse de la doc interactive



FastAPI 0.1.0 OAS 3.1
/openapi.json

default

GET / Root

Parameters — Try it out

No parameters

Responses

Code	Description	Links
200	Successful Response Media type: application/json Controls Accept header. Example Value | Schema	*No links*

```
"string"
```

GET

- Requète GET

☞ Exécuter votre programme *main2.py* avec *Uvicorn*

$ uvicorn main2:app --reload

main2.py

1/4

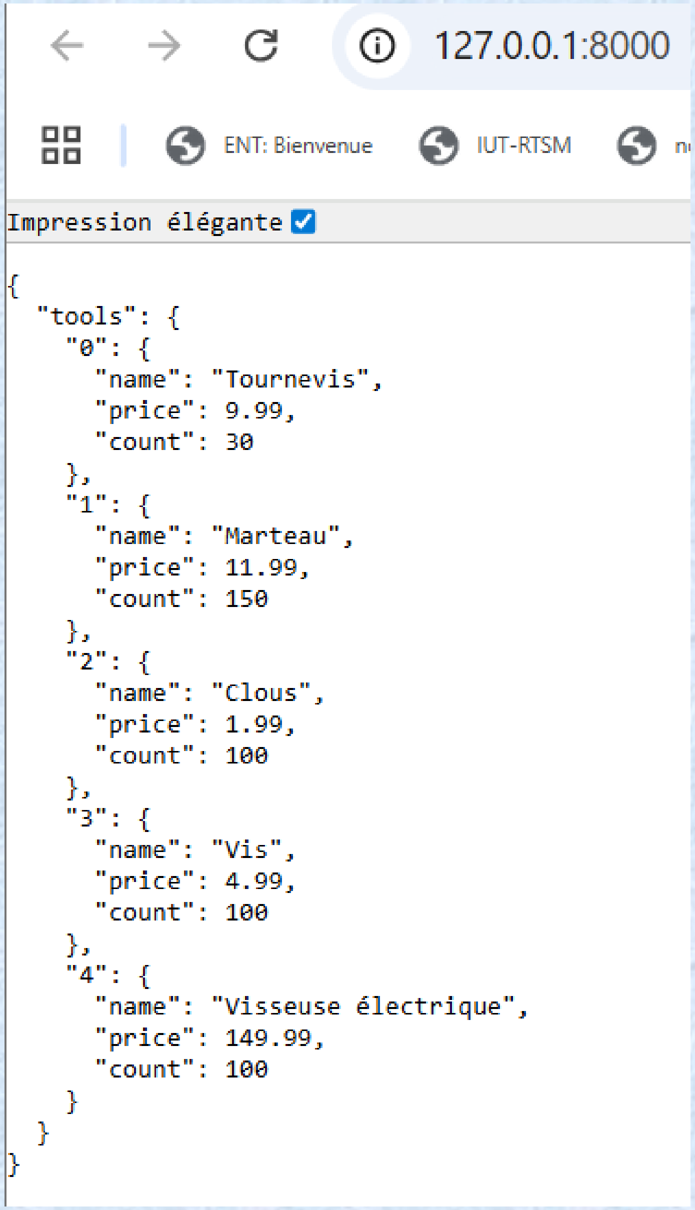
```
from enum import Enum
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()

class Tool(BaseModel):
    name: str
    price: float
    count: int

tools = {
    0: Tool(name="Tournevis", price=9.99, count=30),
    1: Tool(name="Pince", price=5.99, count=20),
    1: Tool(name="Marteau", price=11.99, count=150),
    2: Tool(name="Clous", price=1.99, count=100),
    3: Tool(name="Vis", price=4.99, count=100),
    4: Tool(name="Visseuse électrique", price=149.99, count=100),
}

@app.get("/")
def index() -> dict[str, dict[int, Tool]]:
    return {"tools": tools}
```

FastAPI GET

- Requète GET

☞ Exécuter votre programme *main2.py* avec *Uvicorn*

```
$ uvicorn main2:app --reload
```



```
{
  "tools": {
    "0": {
      "name": "Tournevis",
      "price": 9.99,
      "count": 30
    },
    "1": {
      "name": "Marteau",
      "price": 11.99,
      "count": 150
    },
    "2": {
      "name": "Clous",
      "price": 1.99,
      "count": 100
    },
    "3": {
      "name": "Vis",
      "price": 4.99,
      "count": 100
    },
    "4": {
      "name": "Visseuse électrique",
      "price": 149.99,
      "count": 100
    }
  }
}
```

main2.py

2/4

FastAPI **GET** 6.1

- <u>Requète GET</u>

☞ **Créer un autre programme *main2_recup.py* pour récupérer des requêtes**

main2_recup.py

3/4

```python
import requests

print(requests.get("http://127.0.0.1:8000/").json())
```

Résultat:

➢ *python main2_recup.py*

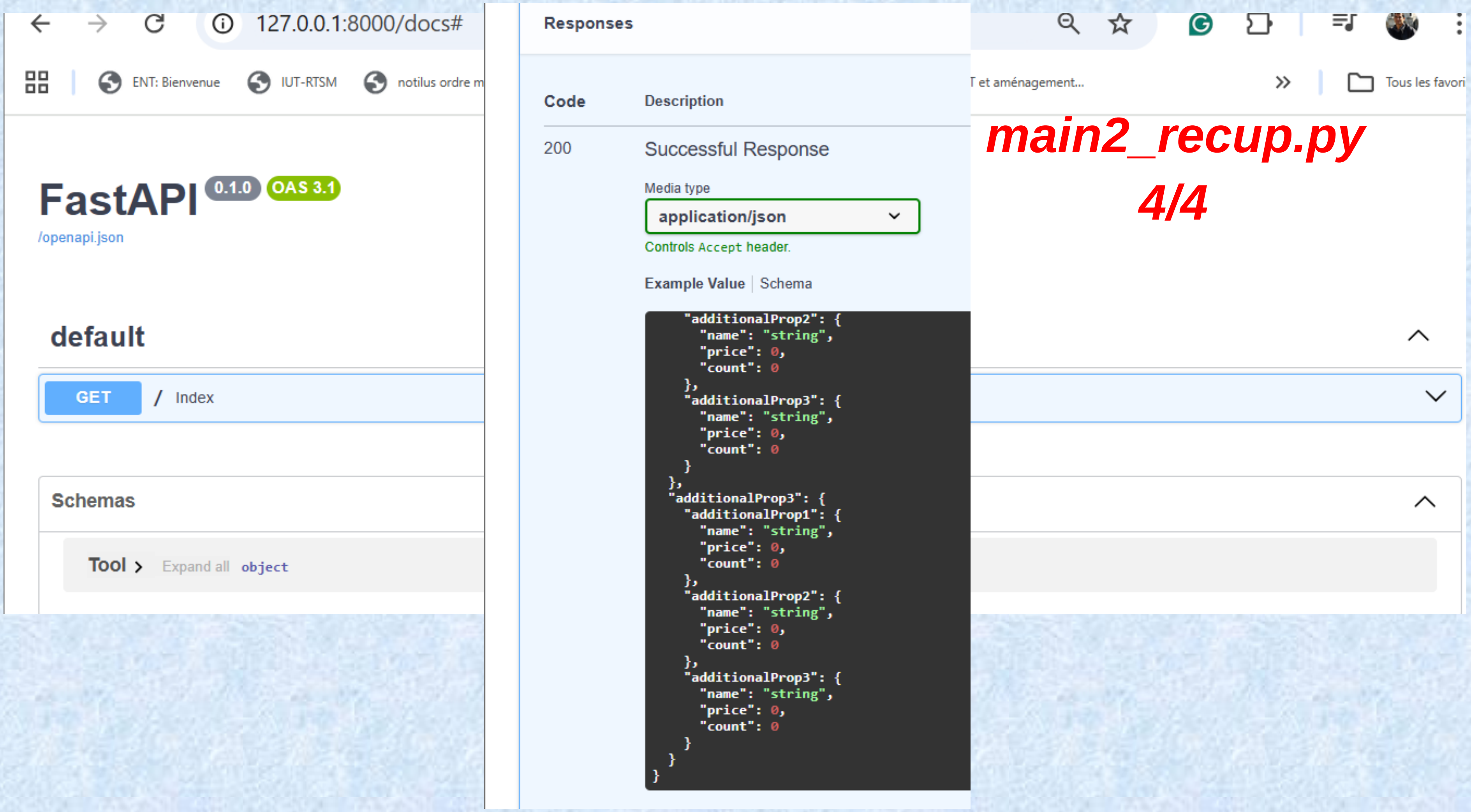
```
{'tools': {'0': {'name': 'Tournevis', 'price': 9.99, 'count': 30}, '1': {'name': 'Marteau', 'price': 11.99, 'count': 150}, '2': {'name': 'Clous', 'price': 1.99, 'count': 100}, '3': {'name': 'Vis', 'price': 4.99, 'count': 100}, '4': {'name': 'Visseuse électrique', 'price': 149.99, 'count': 100}}}
```

FastAPI **GET** **6.1**

- Interactive documentation

☞ **Exécuter la doc de votre programme *main2.py* avec *Uvicorn***

http://127.0.0.1:8000/docs#/ — Adresse de la doc interactive



```
      "additionalProp2": {
        "name": "string",
        "price": 0,
        "count": 0
      },
      "additionalProp3": {
        "name": "string",
        "price": 0,
        "count": 0
      }
    },
    "additionalProp3": {
      "additionalProp1": {
        "name": "string",
        "price": 0,
        "count": 0
      },
      "additionalProp2": {
        "name": "string",
        "price": 0,
        "count": 0
      },
      "additionalProp3": {
        "name": "string",
        "price": 0,
        "count": 0
      }
    }
}
```

main2_recup.py

4/4

FastAPI **GET** **6.1**

- Requète GET

☞ **Exécuter votre programme *main4.py* avec *Uvicorn***

$ uvicorn main4:app --reload

main4.py

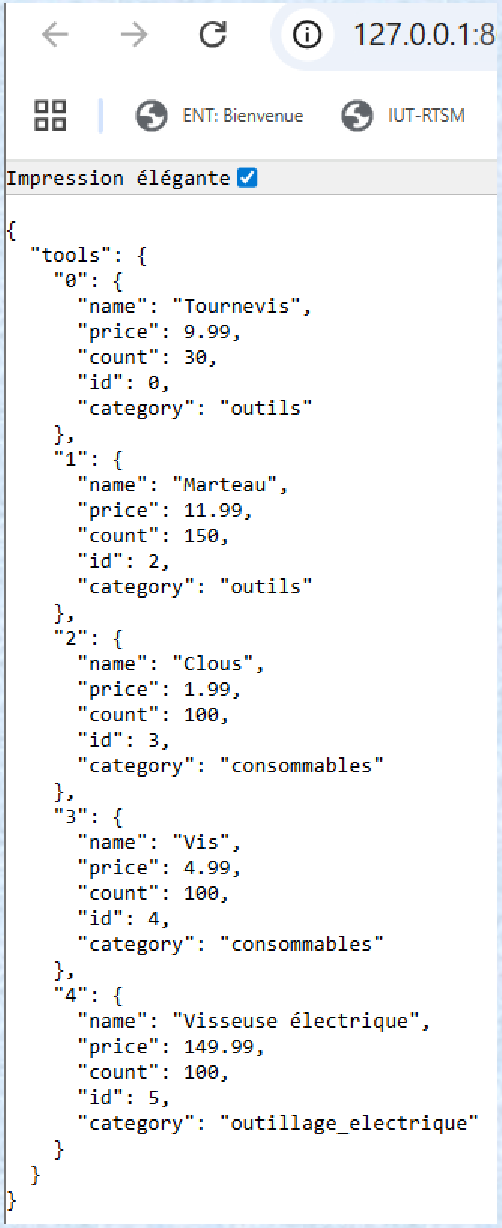
1/5

```
from enum import Enum
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()

class Category(Enum):
    TOOLS = 'outils'
    CONSUMABLES = 'consommables'
    POWER_TOOLs = 'outillage_electrique'

class Tool(BaseModel):
    name: str
    price: float
    count: int
    id: int
    category: Category

tools = {
    0: Tool(name="Tournevis", price=9.99, count=30, id=0, category=Category.TOOLS),
    1: Tool(name="Pince", price=5.99, count=20, id=1, category=Category.TOOLS),
    1: Tool(name="Marteau", price=11.99, count=150, id=2, category=Category.TOOLS),
    2: Tool(name="Clous", price=1.99, count=100, id=3, category=Category.CONSUMABLES),
    3: Tool(name="Vis", price=4.99, count=100, id=4, category=Category.CONSUMABLES),
    4: Tool(name="Visseuse électrique", price=149.99, count=100, id=5,
category=Category.POWER_TOOLs),
}

@app.get("/")
def index() -> dict[str, dict[int, Tool]]:
    return {"tools": tools}
```

- **Requète GET**

☞ **Exécuter votre programme *main4.py* avec *Uvicorn***

```
$ uvicorn main4:app --reload
```



127.0.0.1:8

ENT: Bienvenue   IUT-RTSM

Impression élégante ✓

```
{
  "tools": {
    "0": {
      "name": "Tournevis",
      "price": 9.99,
      "count": 30,
      "id": 0,
      "category": "outils"
    },
    "1": {
      "name": "Marteau",
      "price": 11.99,
      "count": 150,
      "id": 2,
      "category": "outils"
    },
    "2": {
      "name": "Clous",
      "price": 1.99,
      "count": 100,
      "id": 3,
      "category": "consommables"
    },
    "3": {
      "name": "Vis",
      "price": 4.99,
      "count": 100,
      "id": 4,
      "category": "consommables"
    },
    "4": {
      "name": "Visseuse électrique",
      "price": 149.99,
      "count": 100,
      "id": 5,
      "category": "outillage_electrique"
    }
  }
}
```

main4.py

2/5

- **Requète GET**

☞ **Créer un autre programme *main4_recup.py* pour récupérer des requêtes**

```
import requests

print(requests.get("http://127.0.0.1:8000/").json())
```

main4_recup.py

3/5

Résultat:

➢ *python main4_recup.py*

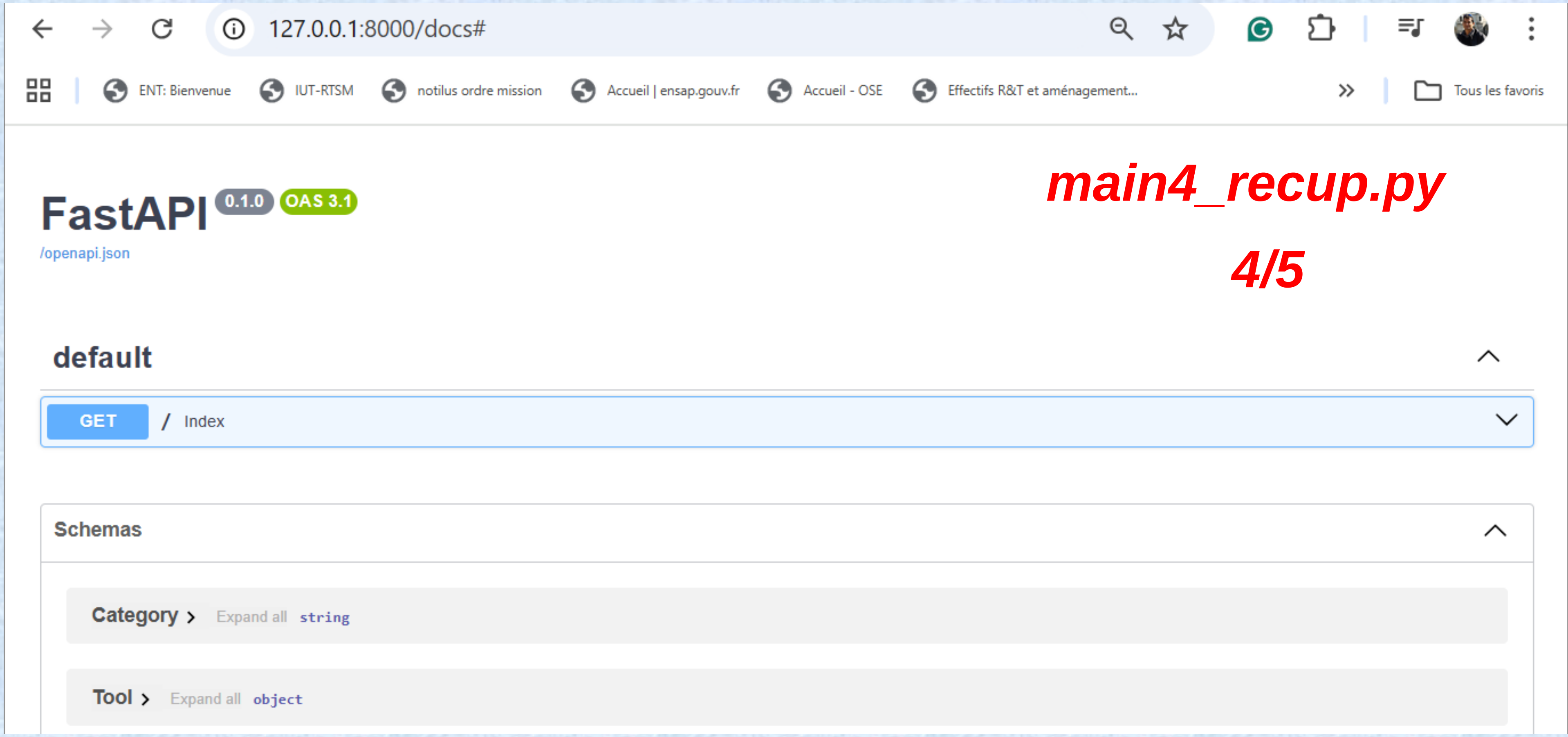
```
{'tools': {'0': {'name': 'Tournevis', 'price': 9.99, 'count': 30, 'id': 0, 'category': 'outils'}, '1': {'name': 'Marteau', 'price': 11.99, 'count': 150, 'id': 2, 'category': 'outils'}, '2': {'name': 'Clous', 'price': 1.99, 'count': 100, 'id': 3, 'category': 'consommables'}, '3': {'name': 'Vis', 'price': 4.99, 'count': 100, 'id': 4, 'category': 'consommables'}, '4': {'name': 'Visseuse électrique', 'price': 149.99, 'count': 100, 'id': 5, 'category': 'outillage_electrique'}}}
```

FastAPI GET 6.1

- Interactive documentation

☞ **Exécuter la doc de votre programme *main.py* avec *Uvicorn***

http://127.0.0.1:8000/docs#/ Adresse de la doc interactive


127.0.0.1:8000/docs#

ENT: Bienvenue | IUT-RTSM | notilus ordre mission | Accueil | ensap.gouv.fr | Accueil - OSE | Effectifs R&T et aménagement... | Tous les favoris

FastAPI 0.1.0 OAS 3.1
/openapi.json

default

GET / Index

Schemas

Category › Expand all string

Tool › Expand all object

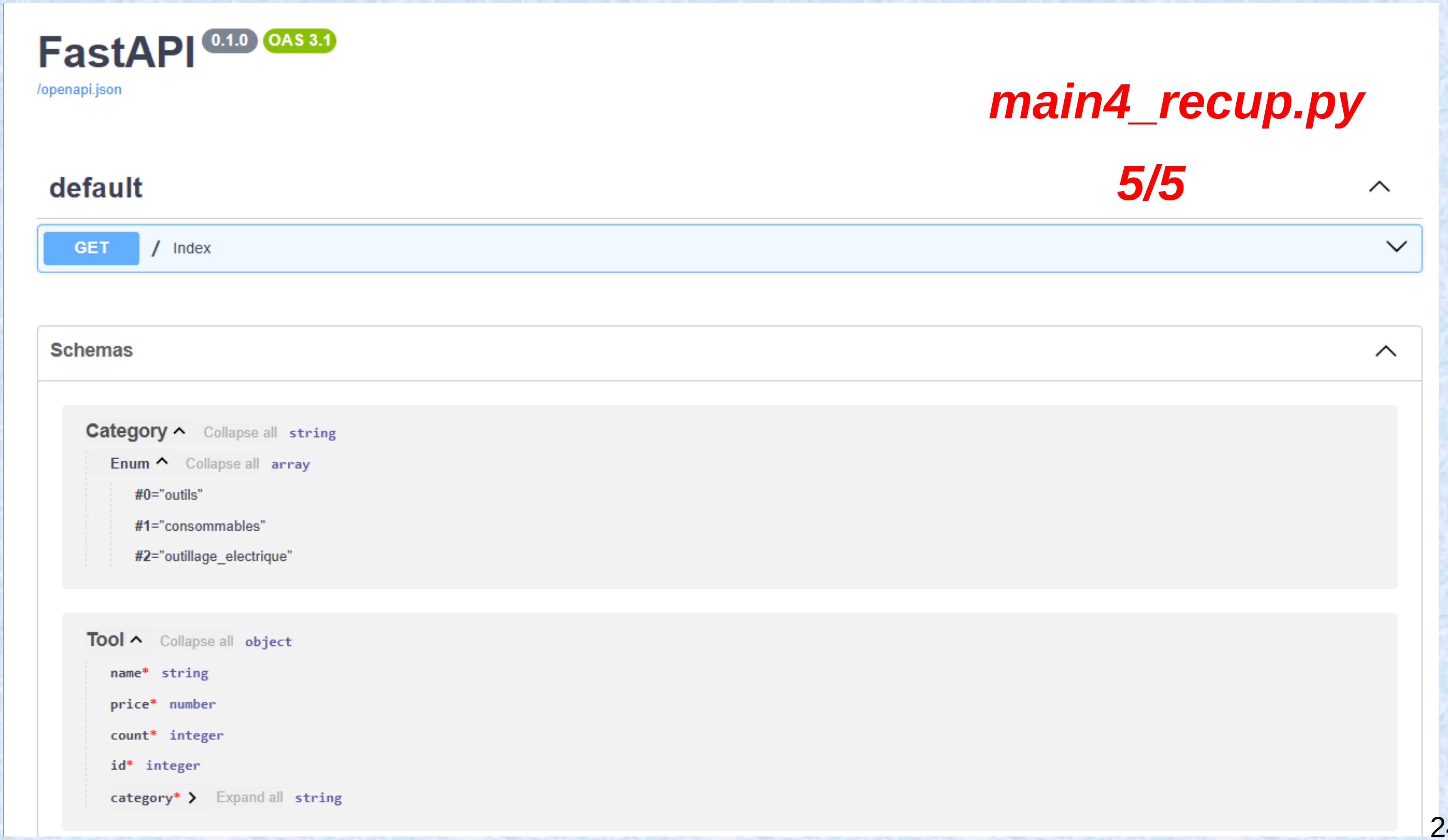
main4_recup.py

4/5

- Interactive documentation

☞ Exécuter la doc de votre programme *main.py* avec *Uvicorn*

http://127.0.0.1:8000/docs#/ — Adresse de la doc interactive



FastAPI 0.1.0 OAS 3.1

/openapi.json

main4_recup.py

5/5

default

GET / Index

Schemas

Category Collapse all string

Enum Collapse all array

#0="outils"

#1="consommables"

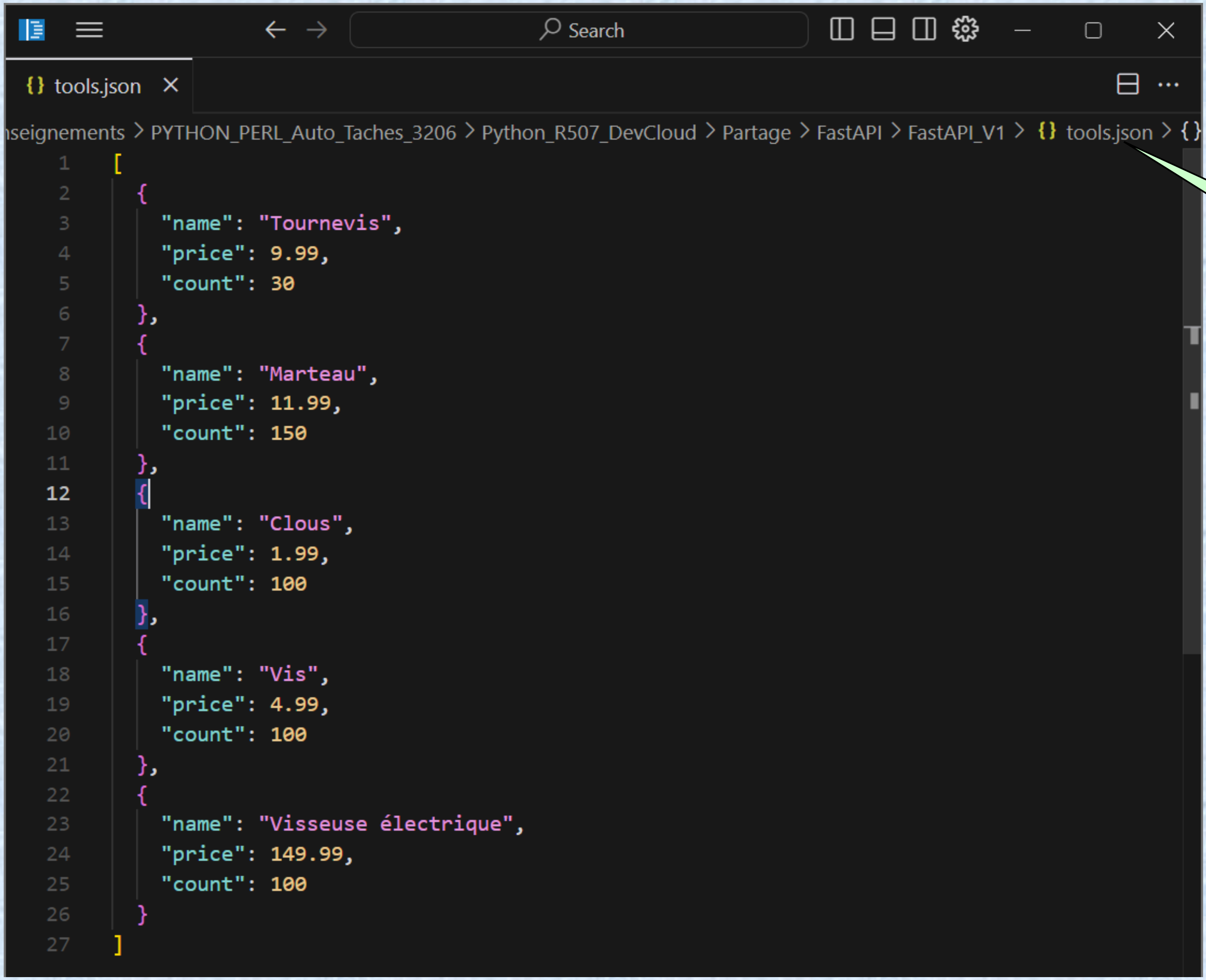
#2="outillage_electrique"

Tool Collapse all object

```
name* string
price* number
count* integer
id* integer
category* > Expand all string
```

- Requète GET

☞ Exécuter votre programme *main2_1.py* avec *Uvicorn* ***$ uvicorn main2_1:app --reload***


tools.json
Search
nseignements > PYTHON_PERL_Auto_Taches_3206 > Python_R507_DevCloud > Partage > FastAPI > FastAPI_V1 > {} tools.json > {}

```
[
  {
    "name": "Tournevis",
    "price": 9.99,
    "count": 30
  },
  {
    "name": "Marteau",
    "price": 11.99,
    "count": 150
  },
  {
    "name": "Clous",
    "price": 1.99,
    "count": 100
  },
  {
    "name": "Vis",
    "price": 4.99,
    "count": 100
  },
  {
    "name": "Visseuse électrique",
    "price": 149.99,
    "count": 100
  }
]
```

tools.json

1/10

fichier JSON

GET

- Requête GET

☞ Exécuter votre programme *main2_1.py* avec *Uvicorn* ***$ uvicorn main2_1:app --reload***

main2_1.py

2/10

```
from enum import Enum
from fastapi import FastAPI, Path, HTTPException
from pydantic import BaseModel
import json

app = FastAPI()

class Tool(BaseModel):
    name: str
    price: float
    count: int

with open("tools.json", "r") as f:
    tools_list = json.load(f)

list_tools = {k+1:v for k, v in enumerate(tools_list)}

@app.get("/")
def index() -> dict[str, dict[int, Tool]]:
    return {"tools": list_tools}

@app.get("/total_tools")
def get_total_tools() -> dict:
    return {"total":len(list_tools)}
```

Lecture d'un fichier json

FastAPI

GET

- Requète GET

☞ Exécuter votre programme *main2_1.py* avec *Uvicorn* ***$ uvicorn main2_1:app --reload***

```
@app.get("/tools")
def get_all_tools1() -> list[Tool]:
    res = []
    for id in list_tools :
        res.append(Tool(**list_tools[id]))

    return res

@app.get("/tool/{id}")
def get_tool_by_id(id: int = Path(ge=1)) -> Tool :

    if id not in list_tools :
        raise HTTPException(status_code=404, detail="Ce tool n'existe pas")

    return Tool(**list_tools[id])
```

main2_1.py

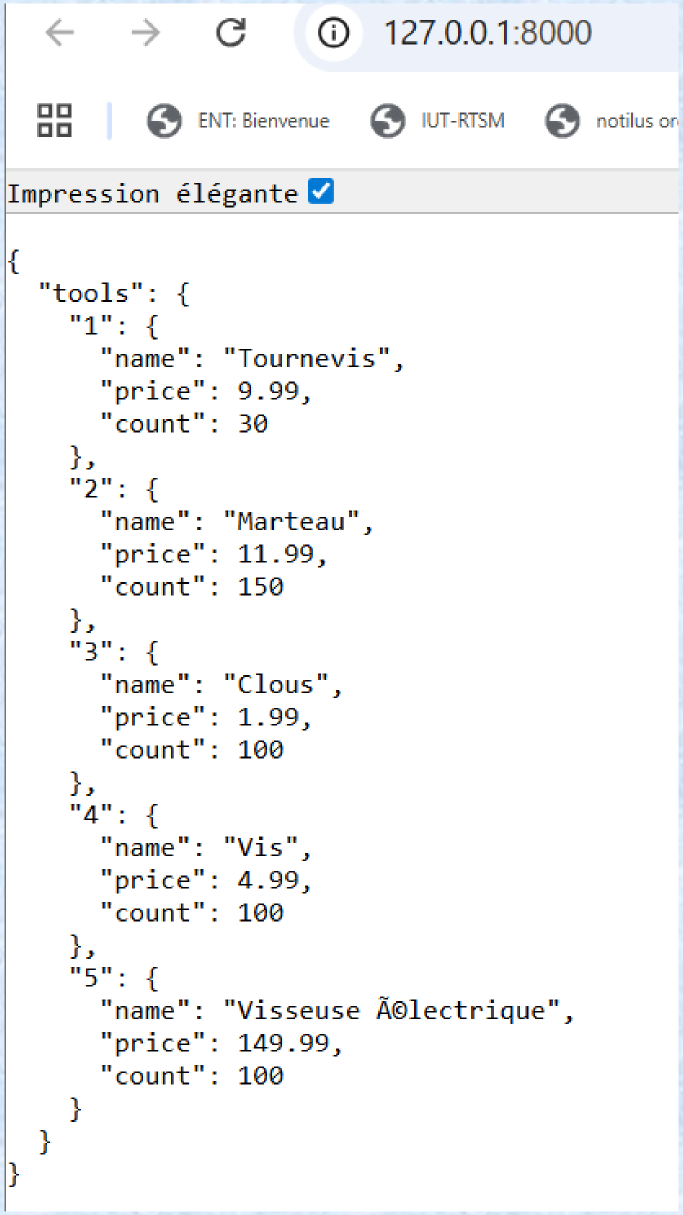
3/10

GET

- Requète GET

☞ Exécuter votre programme *main4.py* avec *Uvicorn*

$ uvicorn main2_1:app --reload



127.0.0.1:8000

Impression élégante ☑

```
{
  "tools": {
    "1": {
      "name": "Tournevis",
      "price": 9.99,
      "count": 30
    },
    "2": {
      "name": "Marteau",
      "price": 11.99,
      "count": 150
    },
    "3": {
      "name": "Clous",
      "price": 1.99,
      "count": 100
    },
    "4": {
      "name": "Vis",
      "price": 4.99,
      "count": 100
    },
    "5": {
      "name": "Visseuse Ã©lectrique",
      "price": 149.99,
      "count": 100
    }
  }
}
```

http://127.0.0.1:8000/

main2_1.py

4/10

FastAPI GET 6.1

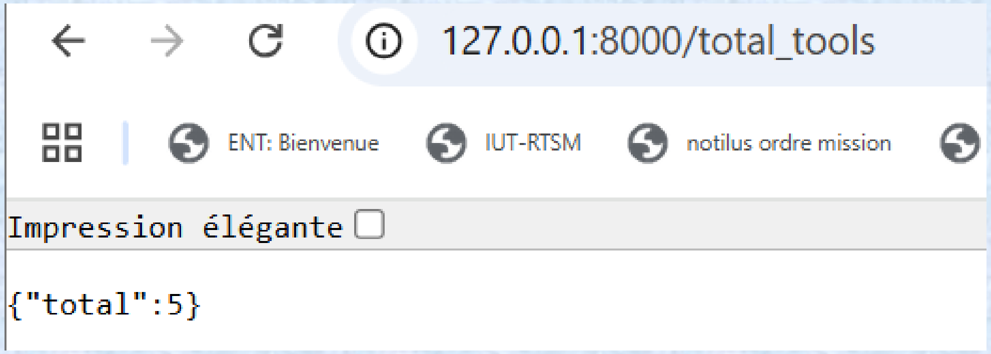
- Requète GET

☞ Exécuter votre programme *main4.py* avec *Uvicorn*

$ uvicorn main2_1:app --reload

```
@app.get("/total_tools")
def get_total_tools() -> dict:
    return {"total":len(list_tools)}
```

http://127.0.0.1:8000/total_tools



```
{"total":5}
```

main2_1.py

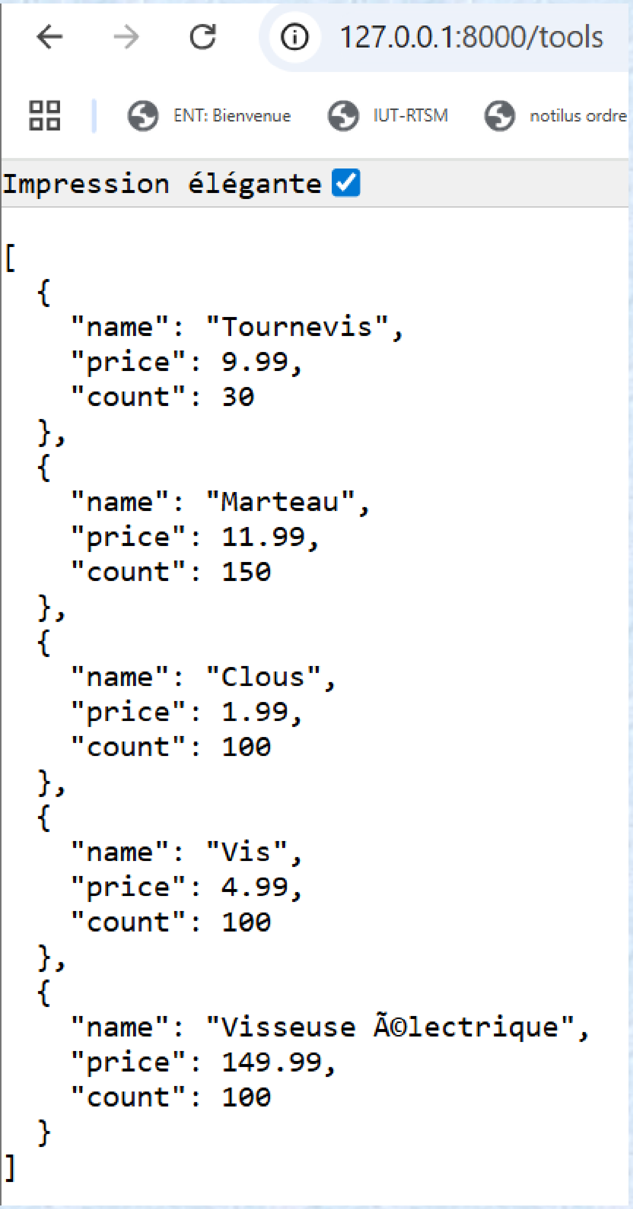
5/10

FastAPI **GET**

- **Requète GET**

☞ **Exécuter votre programme *main4.py* avec *Uvicorn***

$ uvicorn main2_1:app --reload



Impression élégante ✓

```
[
  {
    "name": "Tournevis",
    "price": 9.99,
    "count": 30
  },
  {
    "name": "Marteau",
    "price": 11.99,
    "count": 150
  },
  {
    "name": "Clous",
    "price": 1.99,
    "count": 100
  },
  {
    "name": "Vis",
    "price": 4.99,
    "count": 100
  },
  {
    "name": "Visseuse Ã©lectrique",
    "price": 149.99,
    "count": 100
  }
]
```

http://127.0.0.1:8000/tools

main2_1.py

6/10

```
@app.get("/tools")
def get_all_tools1() -> list[Tool]:
    res = []
    for id in list_tools :
        res.append(Tool(**list_tools[id]))
)

    return res
```

FastAPI GET

- Requète GET

☞ Exécuter votre programme *main4.py* avec *Uvicorn*

```
$ uvicorn main2_1:app --reload
```

On précise l'index que l'on désire !

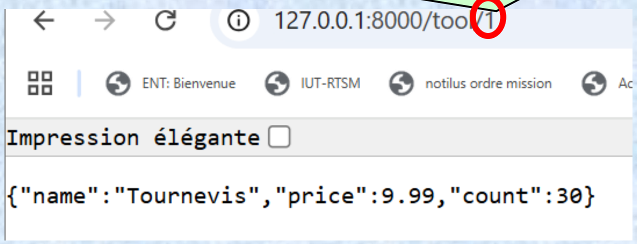
http://127.0.0.1:8000/tool/index

```
@app.get("/tool/{id}")
def get_tool_by_id(id: int = Path(ge=1)) -> Tool :

    if id not in list_tools :
        raise HTTPException(status_code=404, detail="Ce tool n'existe pas")

    return Tool(**list_tools[id])
```

On précise l'index désiré !

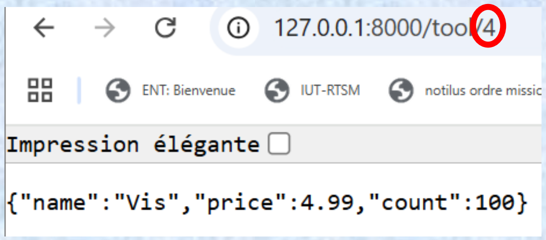
main2_1.py

7/10


127.0.0.1:8000/tool/1

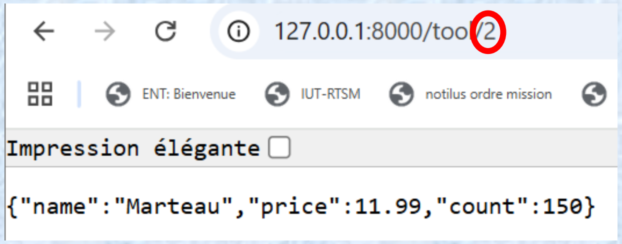
Impression élégante

```
{"name":"Tournevis","price":9.99,"count":30}
```


127.0.0.1:8000/tool/4

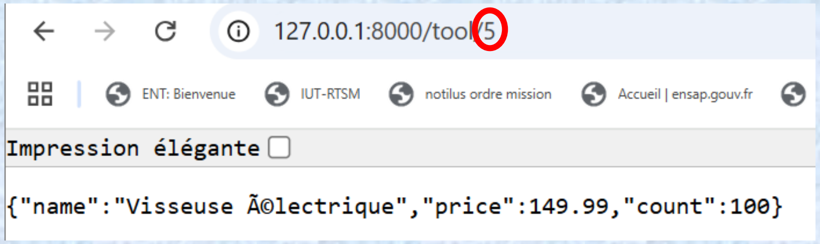
Impression élégante

```
{"name":"Vis","price":4.99,"count":100}
```


127.0.0.1:8000/tool/2

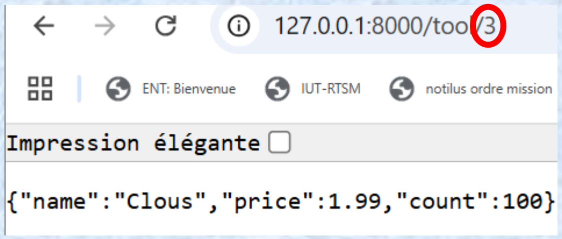
Impression élégante

```
{"name":"Marteau","price":11.99,"count":150}
```


127.0.0.1:8000/tool/5

Impression élégante

```
{"name":"Visseuse Ã©lectrique","price":149.99,"count":100}
```


127.0.0.1:8000/tool/3

Impression élégante

```
{"name":"Clous","price":1.99,"count":100}
```

FastAPI

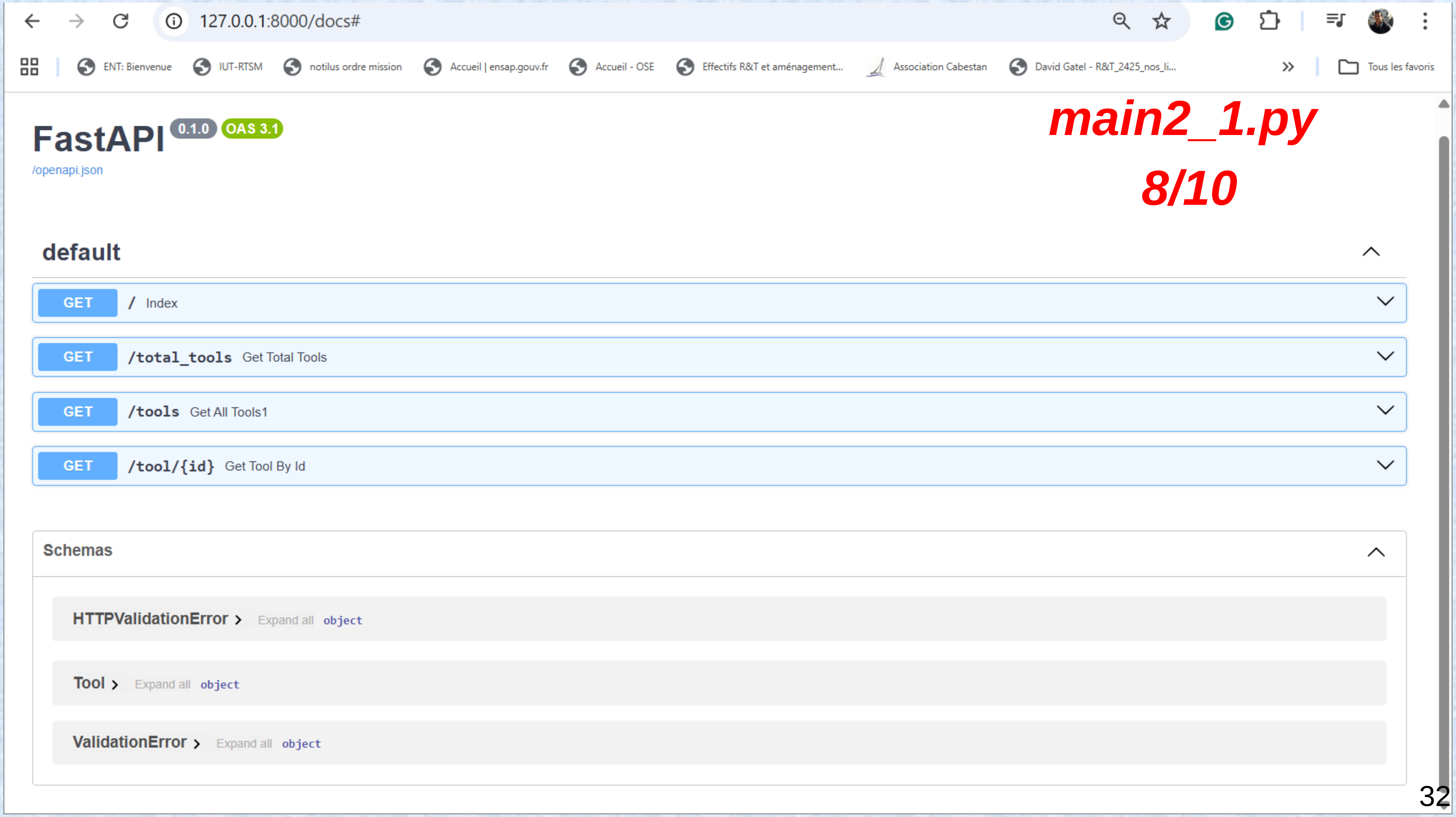
GET

- Requète GET

☞ Exécuter votre programme *main4.py* avec *Uvicorn*

$ uvicorn main2_1:app --reload

http://127.0.0.1:8000/docs#



127.0.0.1:8000/docs#

ENT: Bienvenue | IUT-RTSM | notilus ordre mission | Accueil | ensap.gouv.fr | Accueil - OSE | Effectifs R&T et aménagement... | Association Cabestan | David Gatel - R&T_2425_nos_li... | Tous les favoris

FastAPI 0.1.0 OAS 3.1

/openapi.json

main2_1.py

8/10

default

GET / Index

GET /total_tools Get Total Tools

GET /tools Get All Tools1

GET /tool/{id} Get Tool By Id

Schemas

HTTPValidationError › Expand all object

Tool › Expand all object

ValidationError › Expand all object

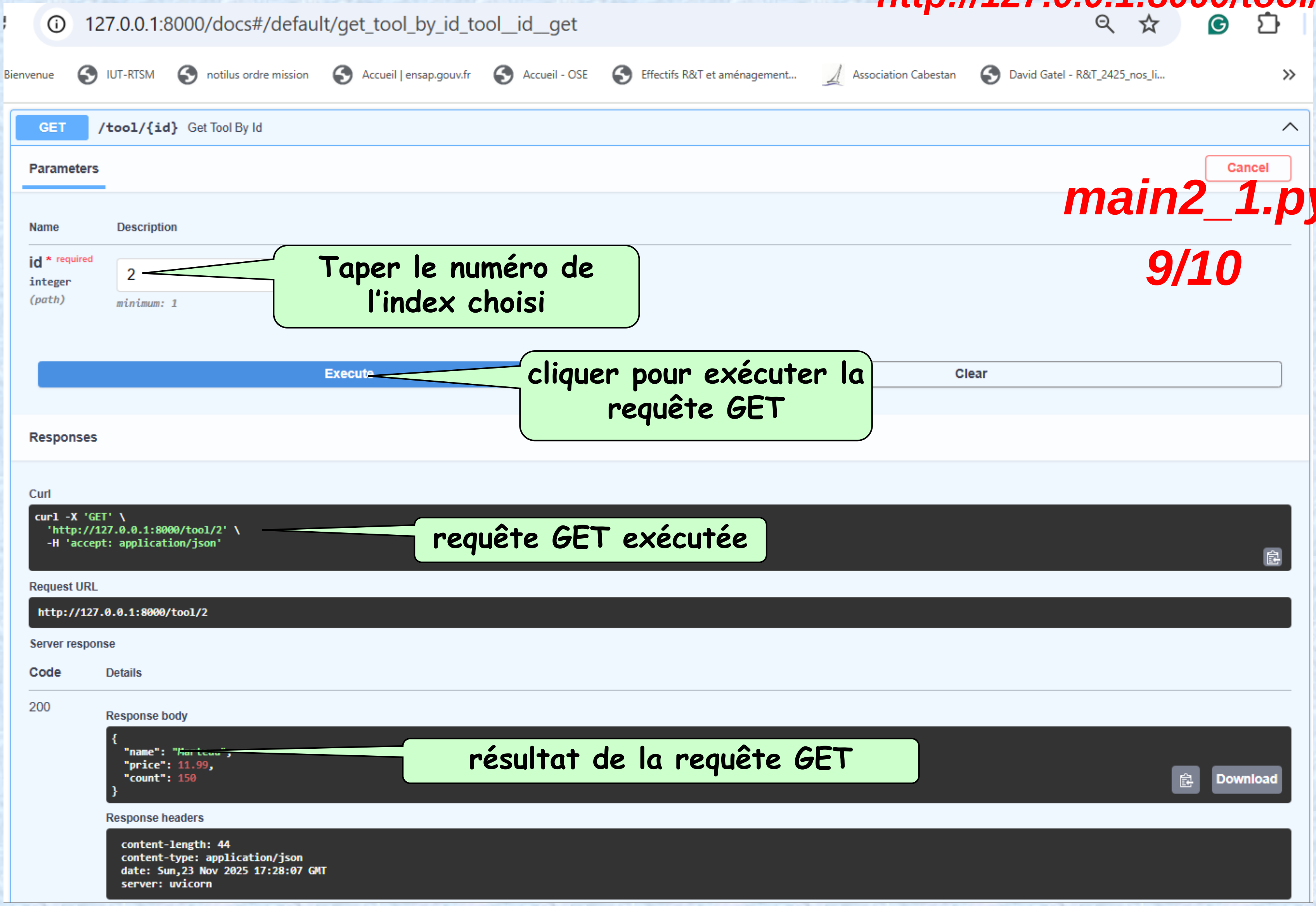
FastAPI **GET** 6.1

• Requète GET

☞ **Exécuter votre programme *main4.py* avec *Uvicorn***

$ uvicorn main2_1:app --reload

http://127.0.0.1:8000/tool/index


127.0.0.1:8000/docs#/default/get_tool_by_id_tool__id__get
Bienvenue · IUT-RTSM · notilus ordre mission · Accueil | ensap.gouv.fr · Accueil - OSE · Effectifs R&T et aménagement... · Association Cabestan · David Gatel - R&T_2425_nos_li...
GET
Cancel
Execute
Clear
Download

/tool/{id} Get Tool By Id

Parameters

Name	Description
id * required integer *(path)*	2 *minimum: 1*

Taper le numéro de l'index choisi

cliquer pour exécuter la requête GET

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/tool/2' \
  -H 'accept: application/json'
```

requête GET exécutée

Request URL

```
http://127.0.0.1:8000/tool/2
```

Server response

Code	Details
200	Response body

```
{
  "name": "Marteau",
  "price": 11.99,
  "count": 150
}
```

résultat de la requête GET

Response headers

```
 content-length: 44
 content-type: application/json
 date: Sun,23 Nov 2025 17:28:07 GMT
 server: uvicorn
```

main2_1.py

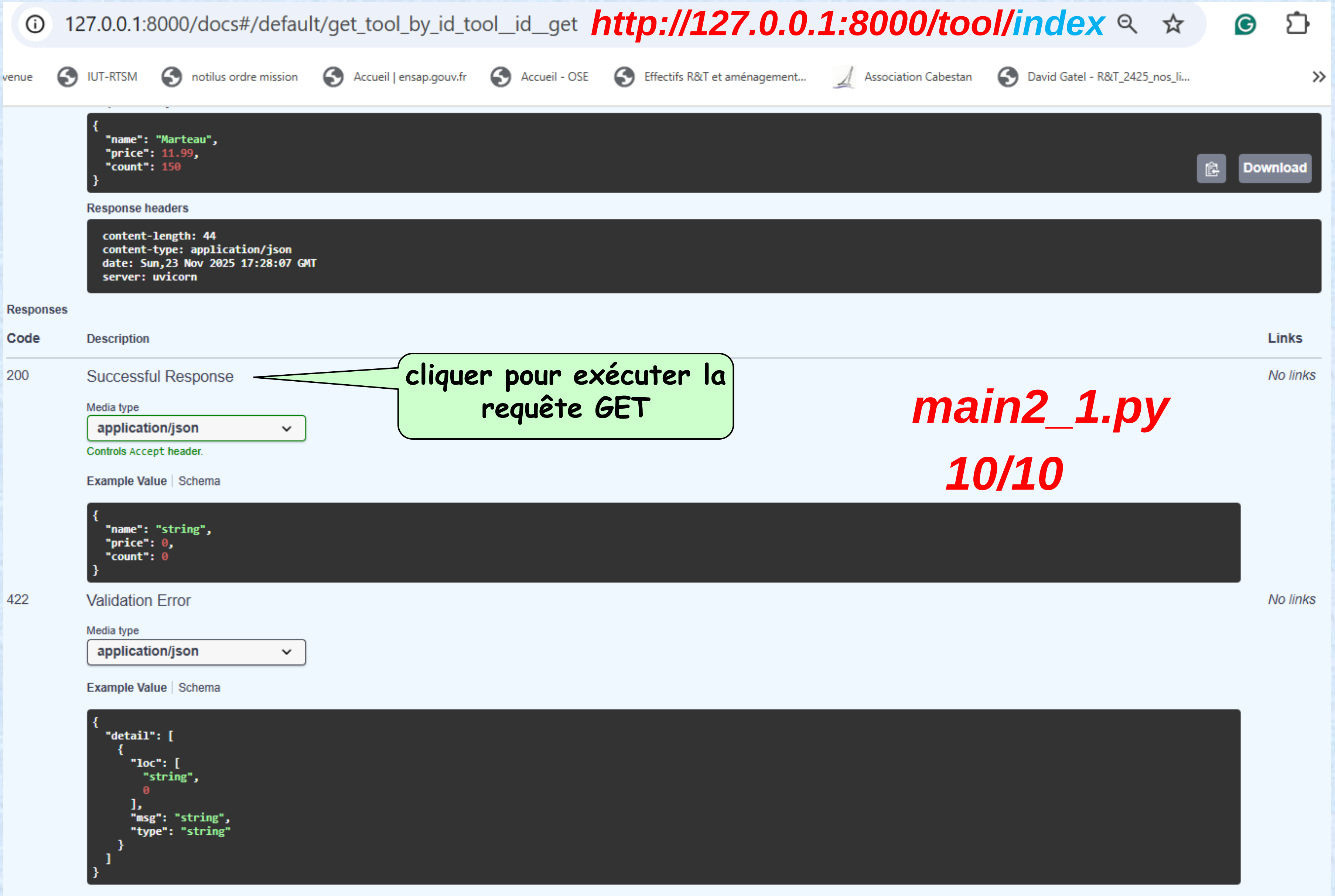
9/10

FastAPI GET 6.1

- **Requète GET**

☞ **Exécuter votre programme *main4.py* avec *Uvicorn***

$ uvicorn main2_1:app --reload


127.0.0.1:8000/docs#/default/get_tool_by_id_tool__id__get

http://127.0.0.1:8000/tool/index

```
{
  "name": "Marteau",
  "price": 11.99,
  "count": 150
}
```

Response headers

```
content-length: 44
content-type: application/json
date: Sun,23 Nov 2025 17:28:07 GMT
server: uvicorn
```

Responses

Code	Description	Links
200	Successful Response	No links
422	Validation Error	No links

200 — Media type: application/json — Controls Accept header. — Example Value | Schema

```
{
  "name": "string",
  "price": 0,
  "count": 0
}
```

422 — Media type: application/json — Example Value | Schema

```
{
  "detail": [
    {
      "loc": [
        "string",
        0
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

cliquer pour exécuter la requête GET

main2_1.py

10/10

• **Requête GET**

☞ **Créer un autre programme *main5.py***

$ uvicorn main5:app --reload

1/6 main5.py

```
from enum import Enum
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()

class Category(Enum):
    TOOLS = 'outils'
    CONSUMABLES = 'consommables'
    POWER_TOOLs = 'outillage_electrique'

class Tool(BaseModel):
    name: str
    price: float
    count: int
    id: int
    category: Category

tools = {
    0: Tool(name="Tournevis", price=9.99, count=30, id=0, category=Category.TOOLS),
    1: Tool(name="Pince", price=5.99, count=20, id=1, category=Category.TOOLS),
    1: Tool(name="Marteau", price=11.99, count=150, id=2, category=Category.TOOLS),
    2: Tool(name="Clous", price=1.99, count=100, id=3, category=Category.CONSUMABLES),
    3: Tool(name="Vis", price=4.99, count=100, id=4, category=Category.CONSUMABLES),
    4: Tool(name="Visseuse électrique", price=149.99, count=100, id=5, category=Category.POWER_TOOLs),
}

@app.get("/")
def index() -> dict[str, dict[int, Tool]]:
    return {"tools": tools}
```

FastAPI GET 6.1

- **Requète GET**

☞ **Créer un autre programme *main5.py***

2/6 *main5.py*

```
@app.get("/tools/{tool_id}")
def query_tool_by_id(tool_id: int) -> Tool:
    if tool_id not in tools:
        raise HTTPException(status_code=404, detail=f"Tool with {tool_id=} does not exist.")
    return tools[tool_id]

Selection = dict[
    str, str | int | float | Category | None
]

@app.get("/tools/")
def query_tool_by_parameters(
    name: str | None = None,
    price: float | None = None,
    count: int | None = None,
    category: Category | None = None,
) -> dict[str, Selection | list[Tool]]:
    def check_tool(tool: Tool):
        """Check if the tool matches the query arguments from the outer scope."""
        return all(
            (
                name is None or tool.name == name,
                price is None or tool.price == price,
                count is None or tool.count != count,
                category is None or tool.category is category,
            )
        )

    selection = [tool for tool in tools.values() if check_tool(tool)]
    return {
        "query": {"name": name, "price": price, "count": count, "category": category},
        "selection": selection,
    }
```

FastAPI

GET

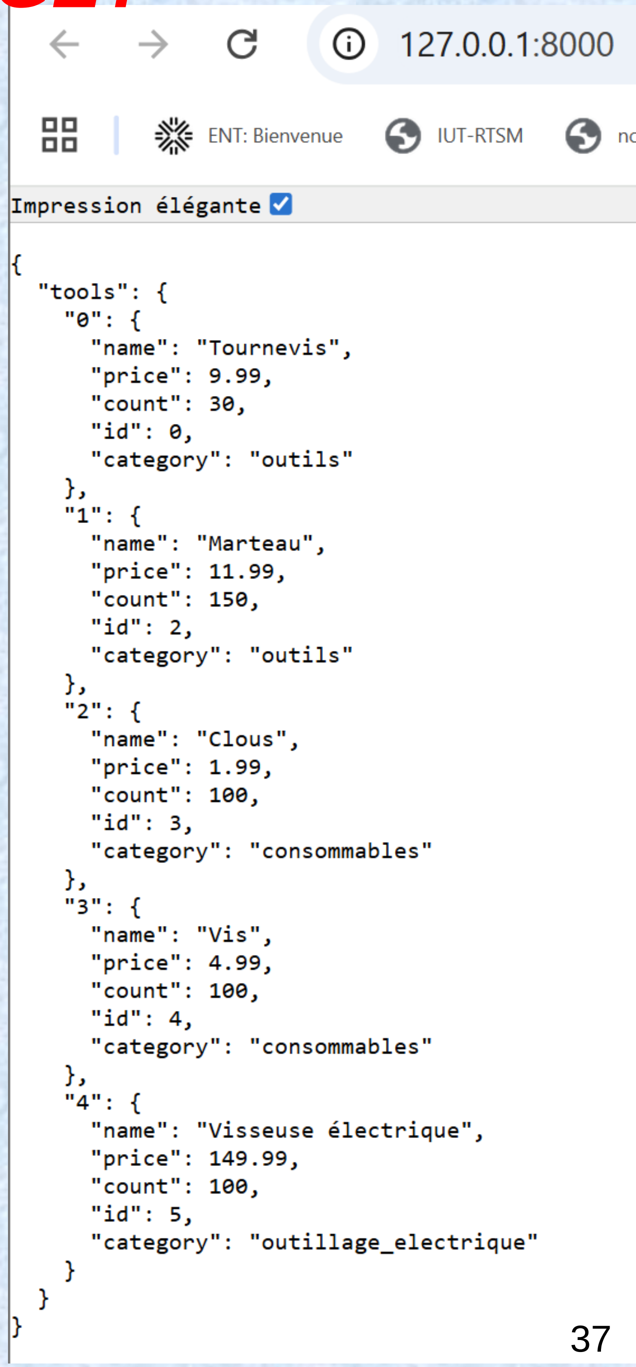
6.1

- Requête GET

☞ Créer un autre programme *main5.py*

```
@app.get("/")
def index() -> dict[str, dict[int, Tool]]:
    return {"tools": tools}
```

3/6



```
{
  "tools": {
    "0": {
      "name": "Tournevis",
      "price": 9.99,
      "count": 30,
      "id": 0,
      "category": "outils"
    },
    "1": {
      "name": "Marteau",
      "price": 11.99,
      "count": 150,
      "id": 2,
      "category": "outils"
    },
    "2": {
      "name": "Clous",
      "price": 1.99,
      "count": 100,
      "id": 3,
      "category": "consommables"
    },
    "3": {
      "name": "Vis",
      "price": 4.99,
      "count": 100,
      "id": 4,
      "category": "consommables"
    },
    "4": {
      "name": "Visseuse électrique",
      "price": 149.99,
      "count": 100,
      "id": 5,
      "category": "outillage_electrique"
    }
  }
}
```

FastAPI

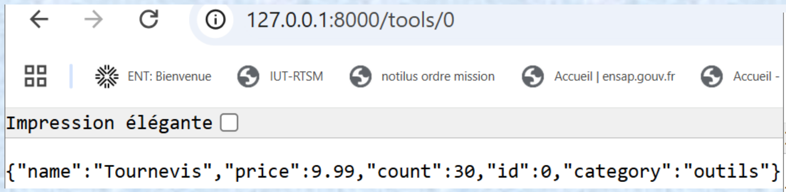
GET

- **Requète GET**

☞ **Créer un autre programme *main5.py***

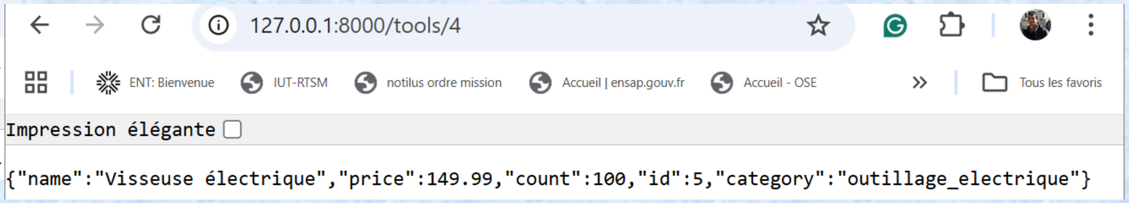
```python
@app.get("/tools/{tool_id}")
def query_tool_by_id(tool_id: int) -> Tool:
    if tool_id not in tools:
        raise HTTPException(status_code=404, detail=f"Tool with {tool_id=} does not exist.")
    return tools[tool_id]
```

4/6


127.0.0.1:8000/tools/0

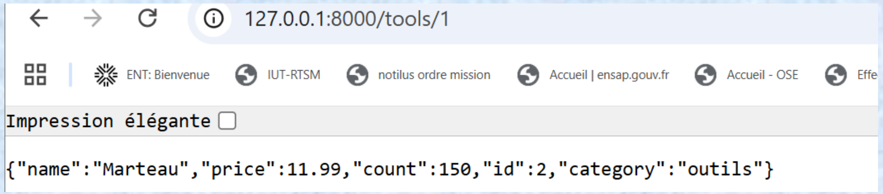
Impression élégante ☐

```
{"name":"Tournevis","price":9.99,"count":30,"id":0,"category":"outils"}
```


127.0.0.1:8000/tools/4

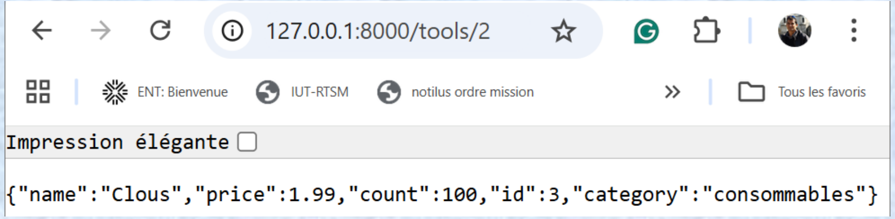
Impression élégante ☐

```
{"name":"Visseuse électrique","price":149.99,"count":100,"id":5,"category":"outillage_electrique"}
```


127.0.0.1:8000/tools/1

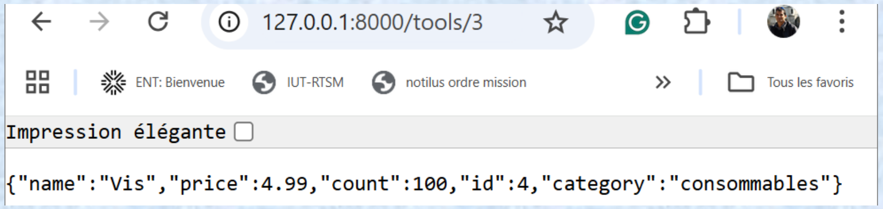
Impression élégante ☐

```
{"name":"Marteau","price":11.99,"count":150,"id":2,"category":"outils"}
```


127.0.0.1:8000/tools/2

Impression élégante ☐

```
{"name":"Clous","price":1.99,"count":100,"id":3,"category":"consommables"}
```


127.0.0.1:8000/tools/3

Impression élégante ☐

```
{"name":"Vis","price":4.99,"count":100,"id":4,"category":"consommables"}
```

- **Requète GET**

☞ **Créer un autre programme *main5.py***

5/6 suite main5.py

```python
Selection = dict[
    str, str | int | float | Category | None
]

@app.get("/tools/")
def query_tool_by_parameters(
    name: str | None = None,
    price: float | None = None,
    count: int | None = None,
    category: Category | None = None,
) -> dict[str, Selection | list[Tool]]:
    def check_tool(tool: Tool):
        """Check if the tool matches the query arguments from the outer scope."""
        return all(
            (
                name is None or tool.name == name,
                price is None or tool.price == price,
                count is None or tool.count != count,
                category is None or tool.category is category,
            )
        )

    selection = [tool for tool in tools.values() if check_tool(tool)]
    return {
        "query": {"name": name, "price": price, "count": count, "category": category},
        "selection": selection,
    }
```

Définition de méthode associée à la requête GET

FastAPI GET 6.1

- **Requète GET**

☞ **Créer un autre programme *main5_recup.py* pour récupérer des requêtes**

http://127.0.0.1:8000/tool/items?name=Marteau

main5_recup.py

6/6

```
import requests

print(requests.get("http://127.0.0.1:8000/items/0").json())
print(requests.get("http://127.0.0.1:8000/items?name=Marteau").json())
```

Résultat:

➢ ***python main5_recup.py***

```
{'name': 'Tournevis', 'price': 9.99, 'count': 30, 'id': 0, 'category': 'outils’}

{'query': {'name': 'Marteau', 'price': None, 'count': None, 'category': None}, 'selection':
[{'name': 'Marteau', 'price': 11.99, 'count': 150, 'id': 2, 'category': 'outils'}]}
```

FastAPI

POST

6.1

- Requète POST

☞ Créer un autre programme *main6.py*

$ uvicorn main6:app --reload

1/24 ***main6.py***

```
from enum import Enum
from pydantic import BaseModel
from fastapi import FastAPI, HTTPException

app = FastAPI()

class Category(Enum):
    TOOLS = "tools"
    CONSUMABLES = "consumables"

class Tool(BaseModel):
    name: str
    price: float
    count: int
    id: int
    category: Category

tools = {
    0: Tool(name="Tournevis", price=9.99, count=30, id=0, category=Category.TOOLS),
    1: Tool(name="Pince", price=5.99, count=20, id=1, category=Category.TOOLS),
    1: Tool(name="Marteau", price=11.99, count=150, id=2, category=Category.TOOLS),
    2: Tool(name="Clous", price=1.99, count=100, id=3, category=Category.CONSUMABLES),
    3: Tool(name="Vis", price=4.99, count=100, id=4, category=Category.CONSUMABLES),
    4: Tool(name="Visseuse électrique", price=149.99, count=100, id=5,
category=Category.POWER_TOOLs),
}
```

GET

- **Requète GET**

☞ **Créer un autre programme *main6.py***

2/24 main6.py

Définition de méthode associée à la requête GET

```
@app.get("/")
def index() -> dict[str, dict[int, Tool]]:
    return {"tools": tools}
```

Définition de méthode associée à la requête GET

```
@app.get("/tools/{tool_id}")
def query_tool_by_id(tool_id: int) -> Tool:
    if tool_id not in tools:
        HTTPException(status_code=404, detail=f"Tool with {tool_id=} does not exist.")
    return tools[tool_id]
```

GET

- **Requète GET**

☞ **Créer un autre programme *main6.py***

3/24 ***main6.py***

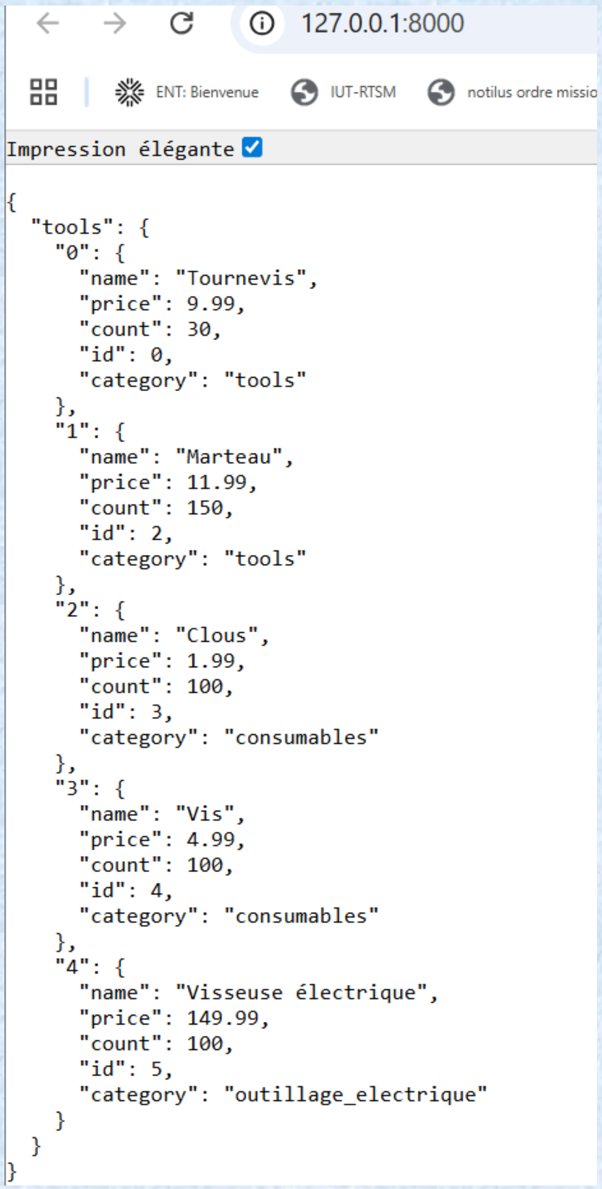
Définition de méthode associée à la requête GET

```python
Selection = dict[
    str, str | int | float | Category | None
]

@app.get("/tools/")
def query_tool_by_parameters(
    name: str | None = None,
    price: float | None = None,
    count: int | None = None,
    category: Category | None = None,
) -> dict[str, Selection | list[Tool]]:
    def check_tool(tool: Tool):
        """Check if the tool matches the query arguments from the outer scope."""
        return all(
            (   name is None or tool.name == name,
                price is None or tool.price == price,
                count is None or tool.count != count,
                category is None or tool.category is category,
            )
        )
    selection = [tool for tool in tools.values() if check_tool(tool)]
    return {
        "query": {"name": name, "price": price, "count": count, "category": category},
        "selection": selection,
    }
```

FastAPI GET 6.1

- **Requête GET**

☞ **Créer un autre programme *main6.py***



127.0.0.1:8000

ENT: Bienvenue IUT-RTSM notilus ordre missio

Impression élégante ✓

```
{
  "tools": {
    "0": {
      "name": "Tournevis",
      "price": 9.99,
      "count": 30,
      "id": 0,
      "category": "tools"
    },
    "1": {
      "name": "Marteau",
      "price": 11.99,
      "count": 150,
      "id": 2,
      "category": "tools"
    },
    "2": {
      "name": "Clous",
      "price": 1.99,
      "count": 100,
      "id": 3,
      "category": "consumables"
    },
    "3": {
      "name": "Vis",
      "price": 4.99,
      "count": 100,
      "id": 4,
      "category": "consumables"
    },
    "4": {
      "name": "Visseuse électrique",
      "price": 149.99,
      "count": 100,
      "id": 5,
      "category": "outillage_electrique"
    }
  }
}
```

4/24 ***main6.py***

FastAPI **POST** **6.1**

- Requête POST

☞ Créer un autre programme *main6.py*

5/24 ***main6.py***

Définition de méthode associée à la requête POST

```python
@app.post("/")
def add_tool(tool: Tool) -> dict[str, Tool]:
    if tool.id in tools:
        HTTPException(status_code=400, detail=f"Tool with {tool.id=} already exists.")
    tools[tool.id] = tool
    return {"added": tool}
```

main6_recup.py

Requête POST

```python
import requests
print("Adding a tool:")
print(
    requests.post(
        "http://127.0.0.1:8000/",
        json={"name": "Scie", "price": 13.50, "count": 28, "id": 5, "category":
"tools"},
    ).json()
)
print(requests.get("http://127.0.0.1:8000/").json())
```

Résultat:

```
➢ python .\main6_modif.py
Adding a tool:
{'added': {'name': 'Scie', 'price': 13.5, 'count': 28, 'id': 5, 'category': 'tools'}}
{'tools': {'0': {'name': 'Tournevis', 'price': 9.99, 'count': 30, 'id': 0, 'category': 'tools'}, '1': {'name':
'Marteau', 'price': 11.99, 'count': 150, 'id': 2, 'category': 'tools'}, '2': {'name': 'Clous', 'price': 1.99, 'count':
100, 'id': 3, 'category': 'consumables'}, '3': {'name': 'Vis', 'price': 4.99, 'count': 100, 'id': 4, 'category':
'consumables'}, '4': {'name': 'Visseuse électrique', 'price': 149.99, 'count': 100, 'id': 5, 'category':
'outillage_electrique'}, '5': {'name': 'Scie', 'price': 13.5, 'count': 28, 'id': 5, 'category': 'tools'}}}
```

Requête POST

- Requète PUT

☞ Créer un autre programme *main6.py*

6/24 main6.py

Définition de méthode associée à la requête PUT

```
@app.put("/update/{tool_id}")
def update(
    tool_id: int,
    name: str | None = None,
    price: float | None = None,
    count: int | None = None,
) -> dict[str, Tool]:

    if tool_id not in tools:
        HTTPException(status_code=404, detail=f"Tool with {tool_id=} does not exist.")
    if all(info is None for info in (name, price, count)):
        raise HTTPException(
            status_code=400, detail="No parameters provided for update."
        )
    tool = tools[tool_id]
    if name is not None:
        tool.name = name
    if price is not None:
        tool.price = price
    if count is not None:
        tool.count = count

    return {"updated": tool}
```

PUT

- Requète PUT

☞ Créer un autre programme *main6.py*

7/24

main6_recup.py

```
import requests

print("Updating a tool:")
print(requests.put("http://127.0.0.1:8000/update/1?count=45").json())
print(requests.get("http://127.0.0.1:8000/").json())
```

Requête PUT

Résultat:

➢ *python .\main6_modif.py*

Requête PUT

```
Updating a tool:
{'updated': {'name': 'Marteau', 'price': 11.99, 'count': 45, 'id': 2, 'category': 'tools'}}
{'tools': {'0': {'name': 'Tournevis', 'price': 9.99, 'count': 30, 'id': 0, 'category': 'tools'}, '1': {'name': 'Marteau', 'price': 11.99,
'count': 45, 'id': 2, 'category': 'tools'}, '2': {'name': 'Clous', 'price': 1.99, 'count': 100, 'id': 3, 'category': 'consumables'}, '3':
{'name': 'Vis', 'price': 4.99, 'count': 100, 'id': 4, 'category': 'consumables'}, '4': {'name': 'Visseuse électrique', 'price':
149.99, 'count': 100, 'id': 5, 'category': 'outillage_electrique'}, '5': {'name': 'Scie', 'price': 13.5, 'count': 28, 'id': 5, 'category':
'tools'}}}
```

FastAPI **DELETE** 6.1

- **Requête DELETE**

☞ **Créer un autre programme *main6_modif.py* pour récupérer des requêtes**

8/24 main6.py

Définition de méthode associée à la requête DELETE

```
@app.delete("/delete/{tool_id}")
def delete_tool(tool_id: int) -> dict[str, Tool]:
    if tool_id not in tools:
        raise HTTPException(
            status_code=404, detail=f"Tool with {tool_id=} does not exist."
        )

    tool = tools.pop(tool_id)
    return {"deleted": tool}
```

main6_modif.py

Requête DELETE

```
import requests
print("Deleting a tool:")
print(requests.delete("http://127.0.0.1:8000/delete/0").json())
print(requests.get("http://127.0.0.1:8000/").json())
```

Résultat:

Requête DELETE

```
➢ python .\main6_modif.py
Deleting a tool:
{'deleted': {'name': 'Tournevis', 'price': 9.99, 'count': 30, 'id': 0, 'category': 'tools'}}
{'tools': {'1': {'name': 'Marteau', 'price': 11.99, 'count': 45, 'id': 2, 'category': 'tools'}, '2': {'name': 'Clous',
'price': 1.99, 'count': 100, 'id': 3, 'category': 'consumables'}, '3': {'name': 'Vis', 'price': 4.99, 'count': 100,
'id': 4, 'category': 'consumables'}, '4': {'name': 'Visseuse électrique', 'price': 149.99, 'count': 100, 'id': 5,
'category': 'outillage_electrique'}, '5': {'name': 'Scie', 'price': 13.5, 'count': 28, 'id': 5, 'category': 'tools'}}}
```

- Requète GET

☞ Créer un autre programme *main6_modif.py* pour récupérer des requêtes

main6_modif.py

9/24

```
import requests

print("Adding a tool:")
print(
    requests.post(
        "http://127.0.0.1:8000/",
        json={"name": "Scie", "price": 13.50, "count": 28, "id": 5, "category":
"tools"},
    ).json()
)
print(requests.get("http://127.0.0.1:8000/").json())
print()

print("Updating a tool:")
print(requests.put("http://127.0.0.1:8000/update/1?count=45").json())
print(requests.get("http://127.0.0.1:8000/").json())
print()

print("Deleting a tool:")
print(requests.delete("http://127.0.0.1:8000/delete/0").json())
print(requests.get("http://127.0.0.1:8000/").json())
```

Requête POST

Requête PUT

Requête DELETE

FastAPI **POST, UPDATE, DELETE** 6.1

- **Requètes POST, UPDATE, DELETE**

☞ **Créer un autre programme *main6_modif.py* pour récupérer des requêtes**

main6_recup.py

10/24

Résultat:

Requête POST

Requête PUT

Requête DELETE

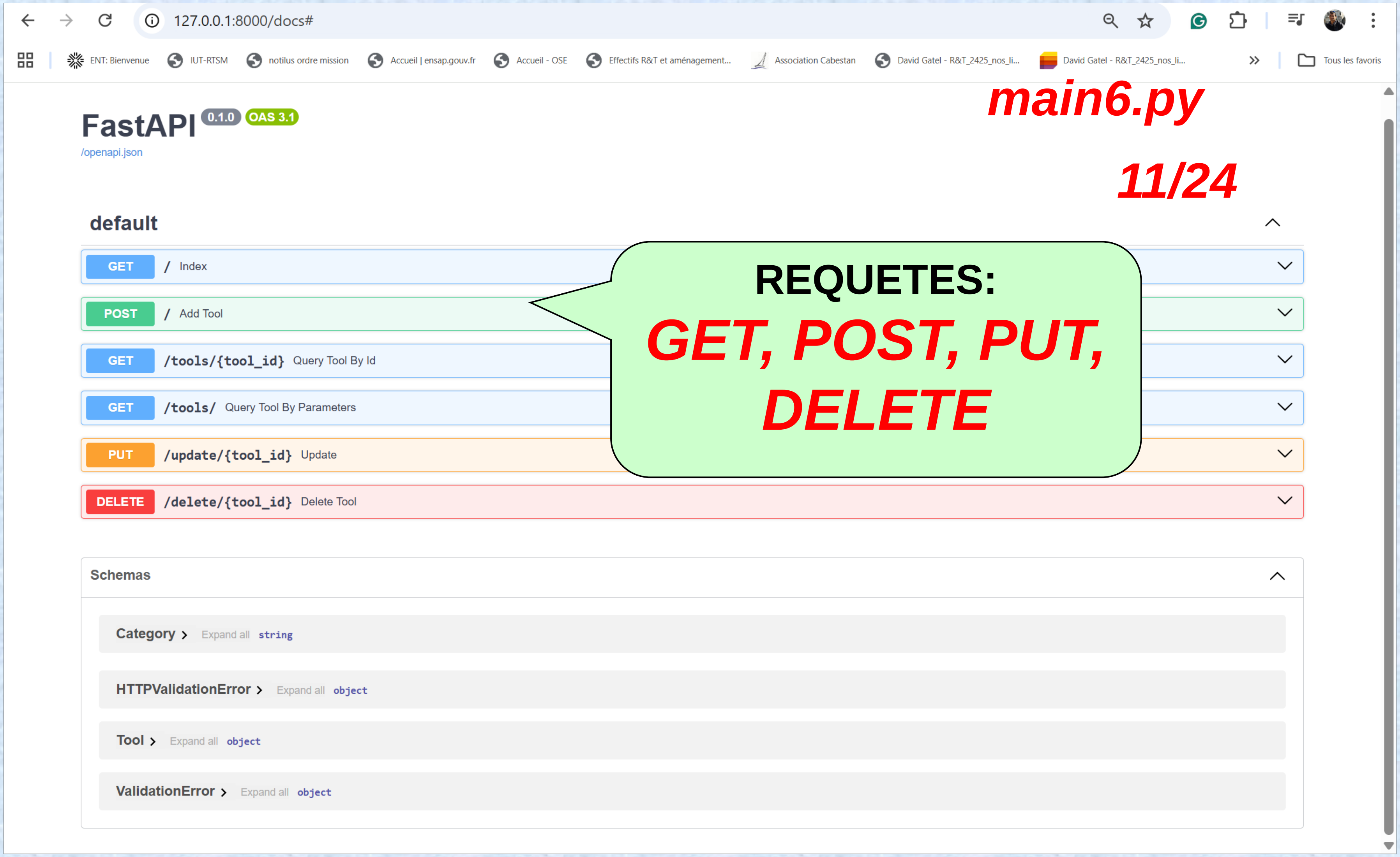
```
➢ python .\main6_modif.py
Adding a tool:
{'added': {'name': 'Scie', 'price': 13.5, 'count': 28, 'id': 5, 'category': 'tools'}}
{'tools': {'0': {'name': 'Tournevis', 'price': 9.99, 'count': 30, 'id': 0, 'category': 'tools'}, '1': {'name':
'Marteau', 'price': 11.99, 'count': 150, 'id': 2, 'category': 'tools'}, '2': {'name': 'Clous', 'price': 1.99, 'count':
100, 'id': 3, 'category': 'consumables'}, '3': {'name': 'Vis', 'price': 4.99, 'count': 100, 'id': 4, 'category':
'consumables'}, '4': {'name': 'Visseuse électrique', 'price': 149.99, 'count': 100, 'id': 5, 'category':
'outillage_electrique'}, '5': {'name': 'Scie', 'price': 13.5, 'count': 28, 'id': 5, 'category': 'tools'}}}

Updating a tool:
{'updated': {'name': 'Marteau', 'price': 11.99, 'count': 45, 'id': 2, 'category': 'tools'}}
{'tools': {'0': {'name': 'Tournevis', 'price': 9.99, 'count': 30, 'id': 0, 'category': 'tools'}, '1': {'name':
'Marteau', 'price': 11.99, 'count': 45, 'id': 2, 'category': 'tools'}, '2': {'name': 'Clous', 'price': 1.99, 'count':
100, 'id': 3, 'category': 'consumables'}, '3': {'name': 'Vis', 'price': 4.99, 'count': 100, 'id': 4, 'category':
'consumables'}, '4': {'name': 'Visseuse électrique', 'price': 149.99, 'count': 100, 'id': 5, 'category':
'outillage_electrique'}, '5': {'name': 'Scie', 'price': 13.5, 'count': 28, 'id': 5, 'category': 'tools'}}}

Deleting a tool:
{'deleted': {'name': 'Tournevis', 'price': 9.99, 'count': 30, 'id': 0, 'category': 'tools'}}
{'tools': {'1': {'name': 'Marteau', 'price': 11.99, 'count': 45, 'id': 2, 'category': 'tools'}, '2': {'name': 'Clous',
'price': 1.99, 'count': 100, 'id': 3, 'category': 'consumables'}, '3': {'name': 'Vis', 'price': 4.99, 'count': 100,
'id': 4, 'category': 'consumables'}, '4': {'name': 'Visseuse électrique', 'price': 149.99, 'count': 100, 'id': 5,
'category': 'outillage_electrique'}, '5': {'name': 'Scie', 'price': 13.5, 'count': 28, 'id': 5, 'category': 'tools'}}}
```

GET, POST, PUT, DELETE

- **Interactive documentation**

☞ **Exécuter la doc de votre programme *main.py* avec *Uvicorn***

http://127.0.0.1:8000/docs#/ Adresse de la doc interactive



127.0.0.1:8000/docs#

ENT: Bienvenue | IUT-RTSM | notilus ordre mission | Accueil | ensap.gouv.fr | Accueil - OSE | Effectifs R&T et aménagement... | Association Cabestan | David Gatel - R&T_2425_nos_li... | David Gatel - R&T_2425_nos_li... | Tous les favoris

main6.py

11/24

FastAPI 0.1.0 OAS 3.1
/openapi.json

default

GET / Index
POST / Add Tool
GET /tools/{tool_id} Query Tool By Id
GET /tools/ Query Tool By Parameters
PUT /update/{tool_id} Update
DELETE /delete/{tool_id} Delete Tool

REQUETES:
GET, POST, PUT, DELETE

Schemas

Category › Expand all string
HTTPValidationError › Expand all object
Tool › Expand all object
ValidationError › Expand all object

FastAPI

POST

6.1

- Interactive documentation

☞ **Exécuter la doc de votre programme *main.py* avec *Uvicorn***

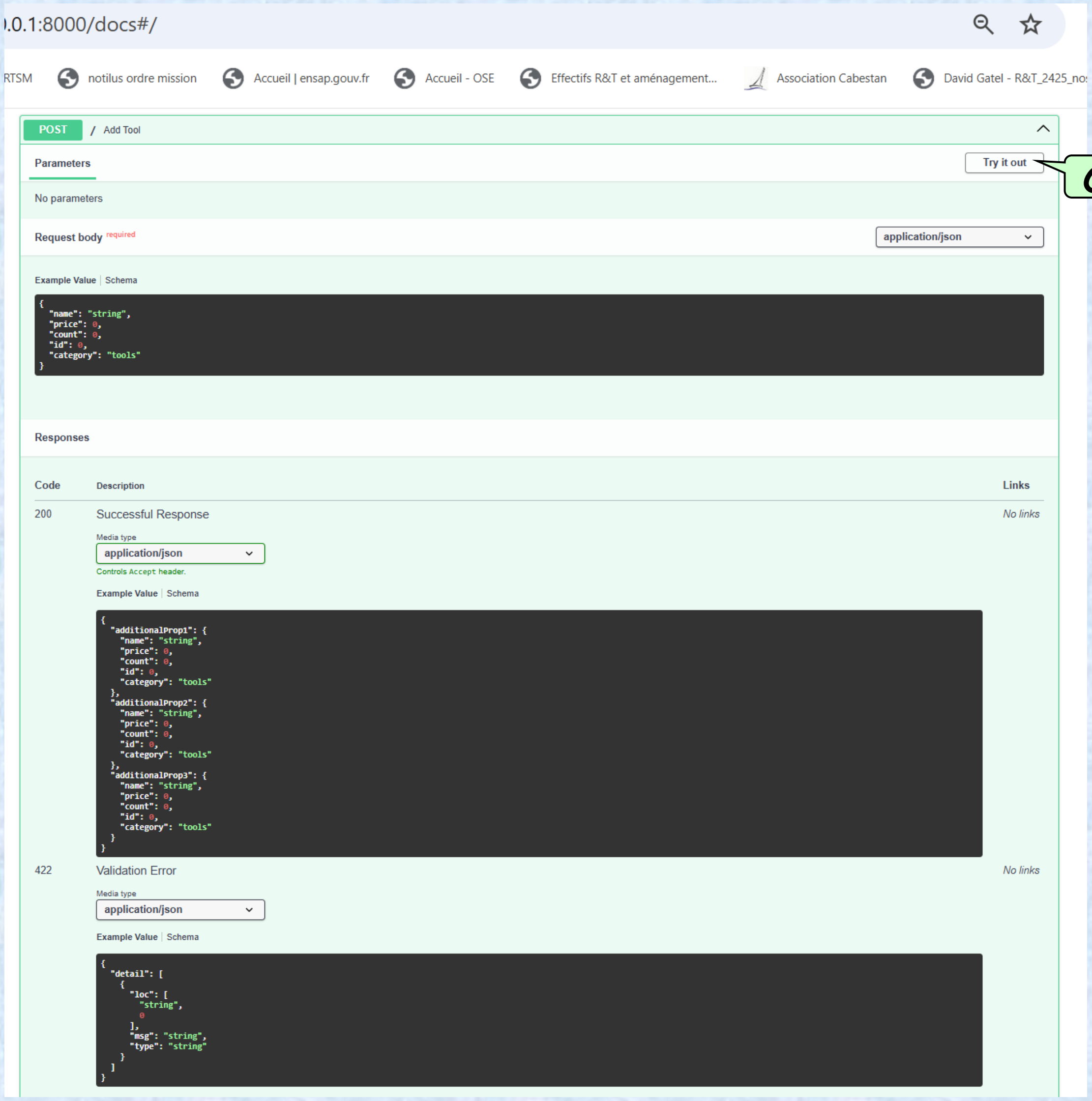
http://127.0.0.1:8000/docs#/

Adresse de la doc interactive

main6.py

12/24

Cliquer pour lancer la requête POST



).0.1:8000/docs#/

RTSM notilus ordre mission Accueil | ensap.gouv.fr Accueil - OSE Effectifs R&T et aménagement... Association Cabestan David Gatel - R&T_2425_no

POST / Add Tool

Parameters — Try it out

No parameters

Request body required — application/json

Example Value | Schema

```
{
  "name": "string",
  "price": 0,
  "count": 0,
  "id": 0,
  "category": "tools"
}
```

Responses

Code	Description	Links
200	Successful Response	No links
422	Validation Error	No links

200 — Media type: application/json

Controls Accept header.

Example Value | Schema

```
{
  "additionalProp1": {
    "name": "string",
    "price": 0,
    "count": 0,
    "id": 0,
    "category": "tools"
  },
  "additionalProp2": {
    "name": "string",
    "price": 0,
    "count": 0,
    "id": 0,
    "category": "tools"
  },
  "additionalProp3": {
    "name": "string",
    "price": 0,
    "count": 0,
    "id": 0,
    "category": "tools"
  }
}
```

422 — Media type: application/json

Example Value | Schema

```
{
  "detail": [
    {
      "loc": [
        "string",
        0
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

FastAPI POST 6.1

- **Interactive documentation**

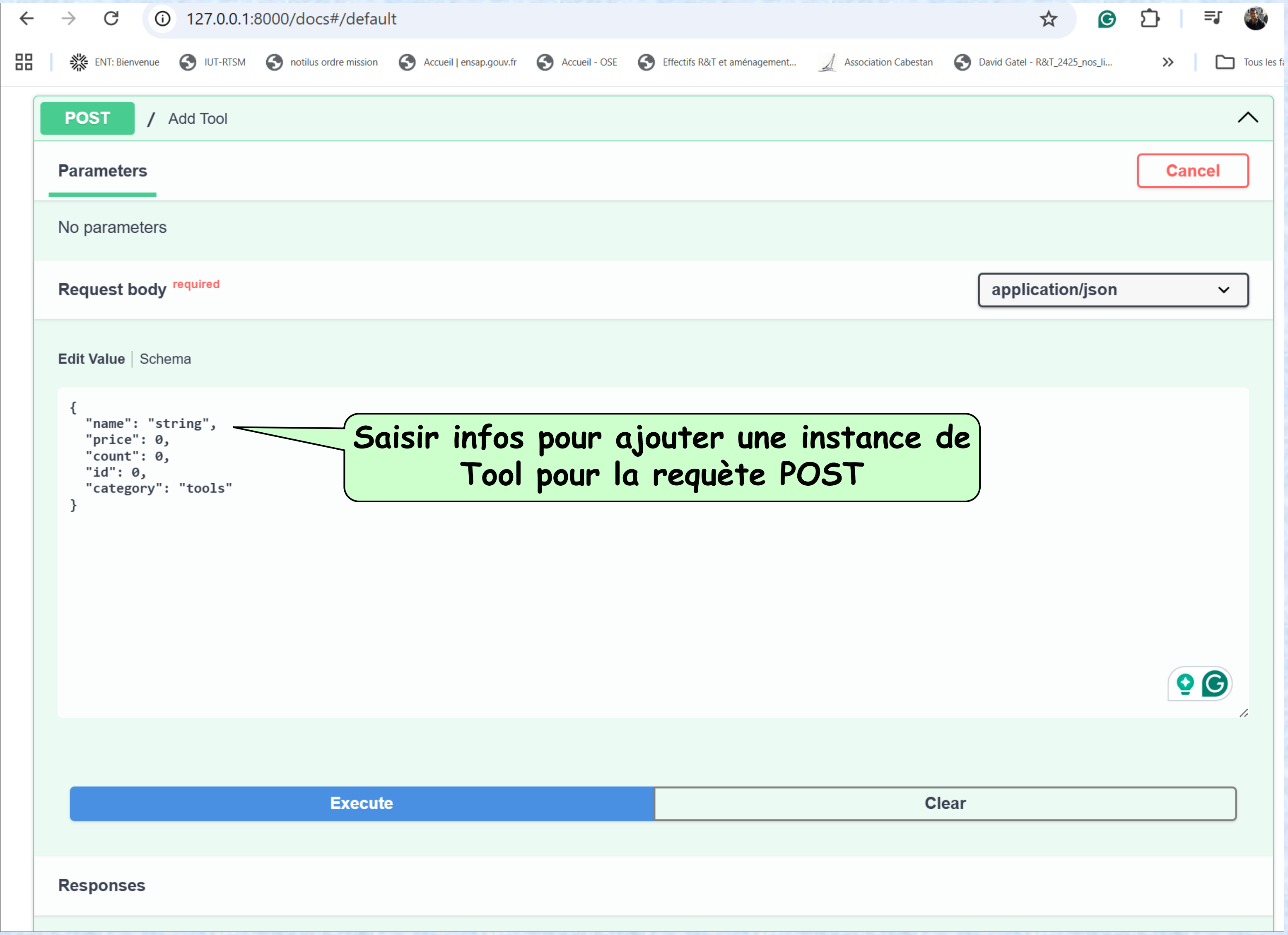
☞ **Exécuter la doc de votre programme *main.py* avec *Uvicorn***

http://127.0.0.1:8000/docs#/

Adresse de la doc interactive

main6.py

13/24



```
{
  "name": "string",
  "price": 0,
  "count": 0,
  "id": 0,
  "category": "tools"
}
```

Saisir infos pour ajouter une instance de Tool pour la requète POST

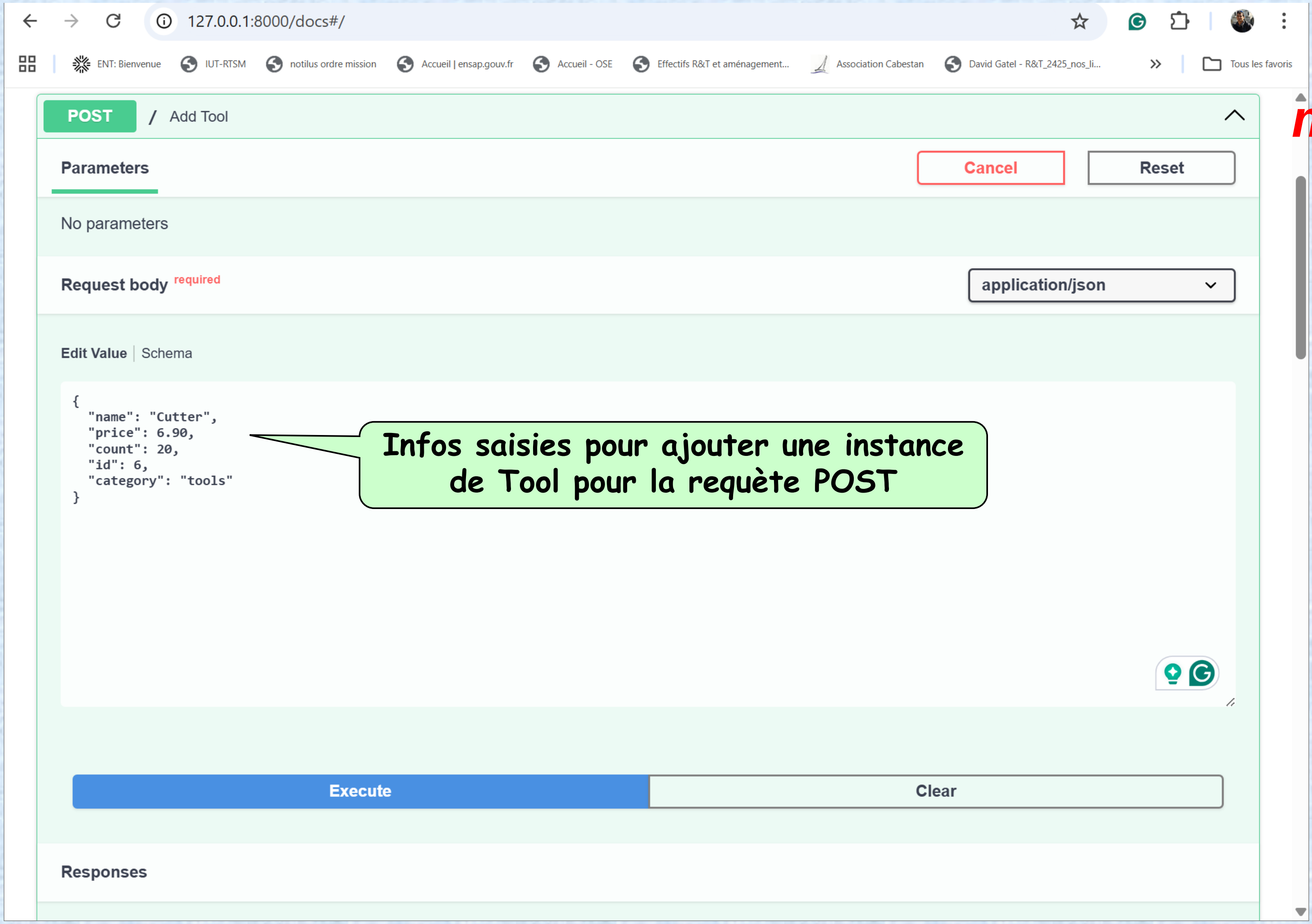
FastAPI **POST** **6.1**

- **Interactive documentation**

☞ **Exécuter la doc de votre programme *main6.py* avec *Uvicorn***

http://127.0.0.1:8000/docs#/

Adresse de la doc interactive



```
{
  "name": "Cutter",
  "price": 6.90,
  "count": 20,
  "id": 6,
  "category": "tools"
}
```

Infos saisies pour ajouter une instance de Tool pour la requète POST

main6.py

14/24

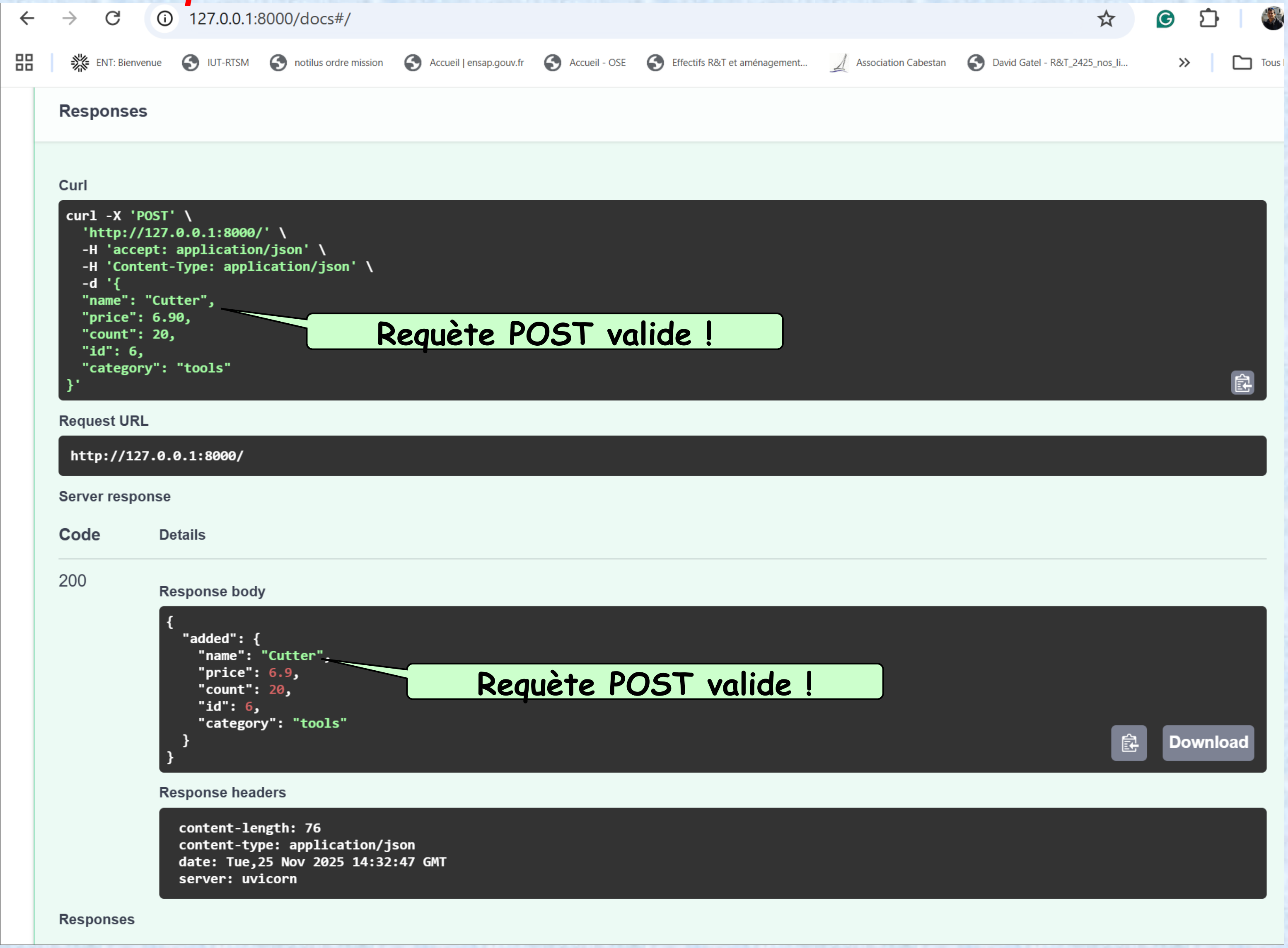
POST

- Interactive documentation

☞ **Exécuter la doc de votre programme *main6.py* avec *Uvicorn***

http://127.0.0.1:8000/docs#/ — Adresse de la doc interactive

main6.py
15/24



Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "name": "Cutter",
  "price": 6.90,
  "count": 20,
  "id": 6,
  "category": "tools"
}'
```

Request URL

```
http://127.0.0.1:8000/
```

Server response

Code | Details

200

Response body

```
{
  "added": {
    "name": "Cutter",
    "price": 6.9,
    "count": 20,
    "id": 6,
    "category": "tools"
  }
}
```

Response headers

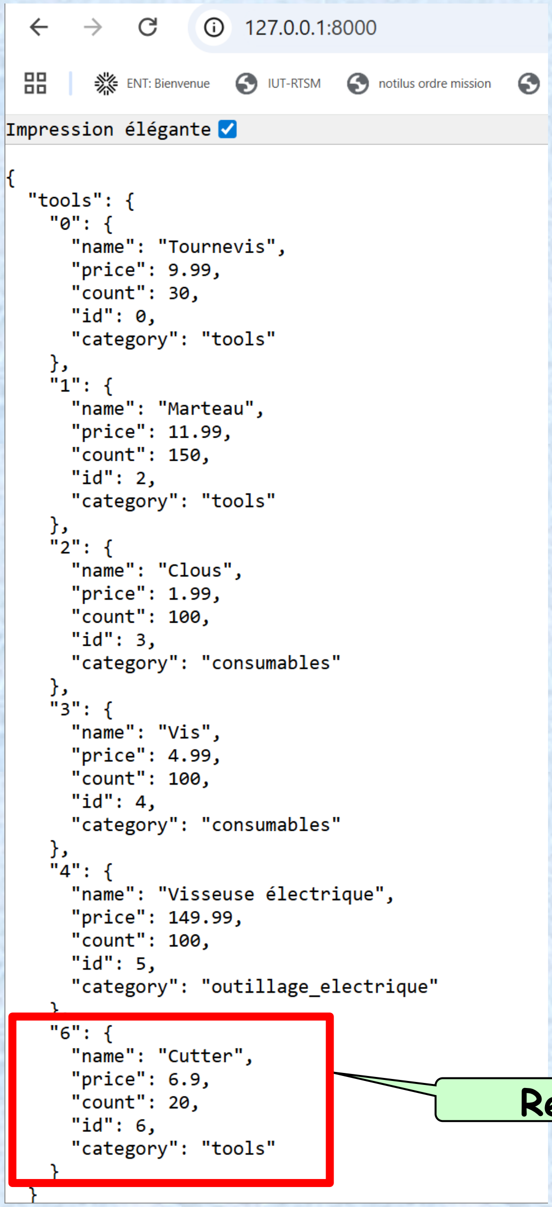
```
content-length: 76
content-type: application/json
date: Tue,25 Nov 2025 14:32:47 GMT
server: uvicorn
```

Responses

POST

- Requète POST

☞ Créer un autre programme *main6.py* pour récupérer des requêtes



127.0.0.1:8000

```
{
  "tools": {
    "0": {
      "name": "Tournevis",
      "price": 9.99,
      "count": 30,
      "id": 0,
      "category": "tools"
    },
    "1": {
      "name": "Marteau",
      "price": 11.99,
      "count": 150,
      "id": 2,
      "category": "tools"
    },
    "2": {
      "name": "Clous",
      "price": 1.99,
      "count": 100,
      "id": 3,
      "category": "consumables"
    },
    "3": {
      "name": "Vis",
      "price": 4.99,
      "count": 100,
      "id": 4,
      "category": "consumables"
    },
    "4": {
      "name": "Visseuse électrique",
      "price": 149.99,
      "count": 100,
      "id": 5,
      "category": "outillage_electrique"
    },
    "6": {
      "name": "Cutter",
      "price": 6.9,
      "count": 20,
      "id": 6,
      "category": "tools"
    }
  }
}
```

Requète POST valide !

http://127.0.0.1:8000/docs#/

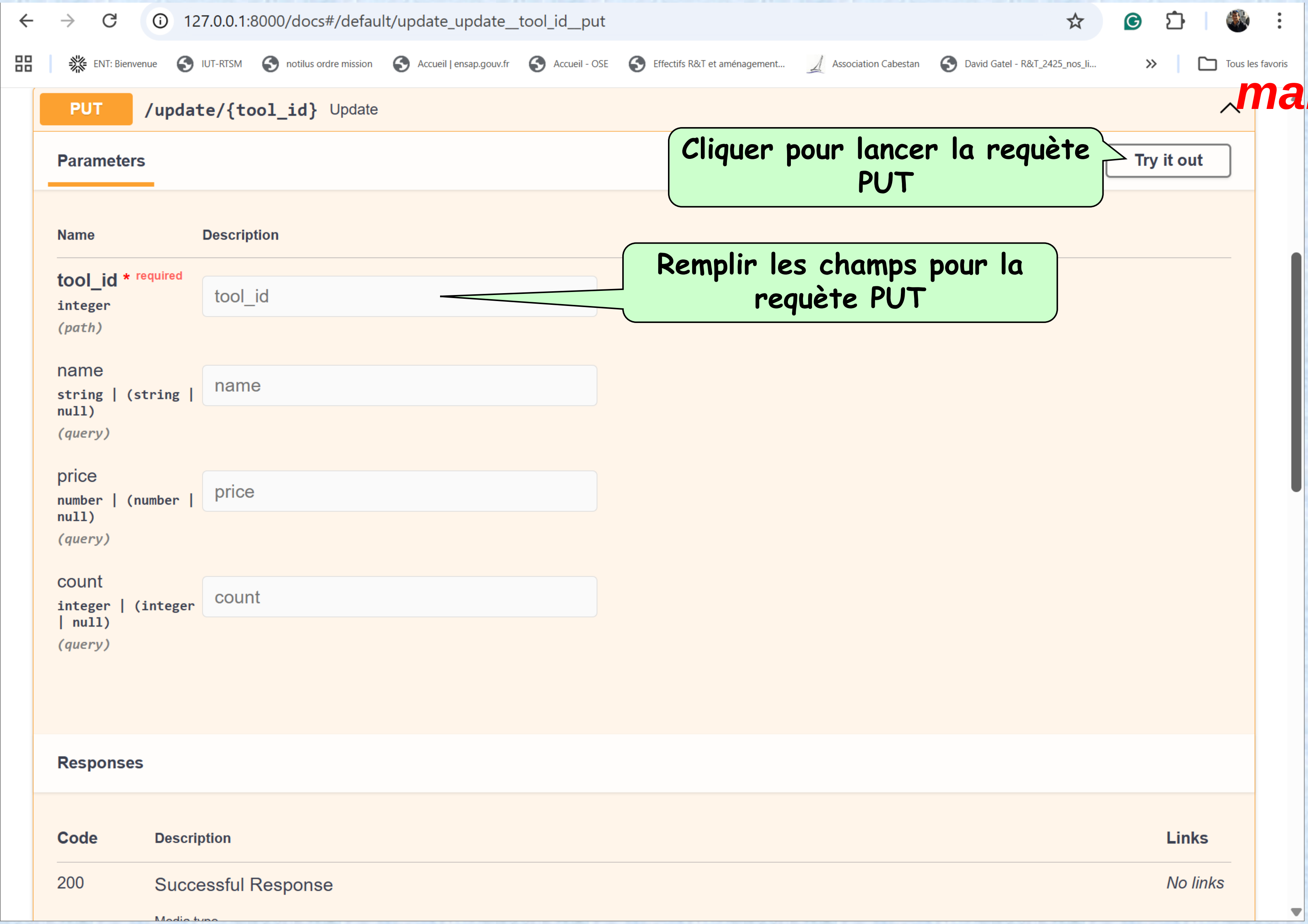
main6.py

16/24

FastAPI **PUT** **6.1**

- **Requète PUT** ***http://127.0.0.1:8000/docs#/***

☞ **Créer un autre programme *main6.py* pour récupérer des requêtes**


127.0.0.1:8000/docs#/default/update_update__tool_id__put
ENT: Bienvenue
IUT-RTSM
notilus ordre mission
Accueil | ensap.gouv.fr
Accueil - OSE
Effectifs R&T et aménagement...
Association Cabestan
David Gatel - R&T_2425_nos_li...
Tous les favoris
Cliquer pour lancer la requète PUT
Try it out
Remplir les champs pour la requète PUT

PUT `/update/{tool_id}` Update

Parameters

Name	Description
tool_id * required `integer` *(path)*	tool_id
name `string | (string | null)` *(query)*	name
price `number | (number | null)` *(query)*	price
count `integer | (integer | null)` *(query)*	count

Responses

Code	Description	Links
200	Successful Response	*No links*

main6.py
17/24

FastAPI

PUT

6.1

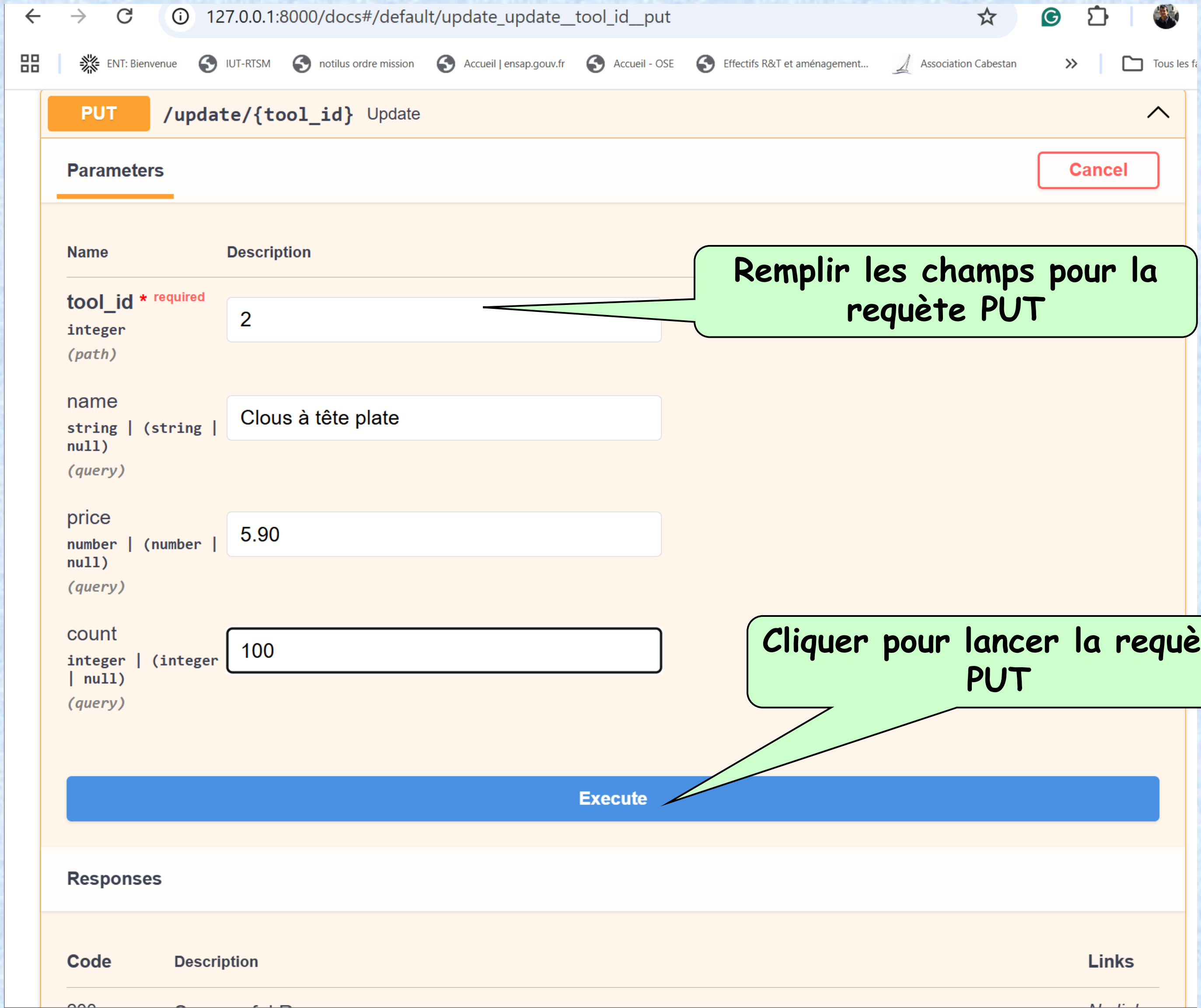
- **Requète PUT**

☞ **Créer un autre programme *main6.py* pour récupérer des requêtes**

http://127.0.0.1:8000/docs#/

main6.py

18/24



PUT `/update/{tool_id}` Update

Parameters — Cancel

Name	Description
tool_id * required `integer` *(path)*	2
name `string | (string | null)` *(query)*	Clous à tête plate
price `number | (number | null)` *(query)*	5.90
count `integer | (integer | null)` *(query)*	100

Remplir les champs pour la requète PUT

Cliquer pour lancer la requète PUT

Execute

Responses

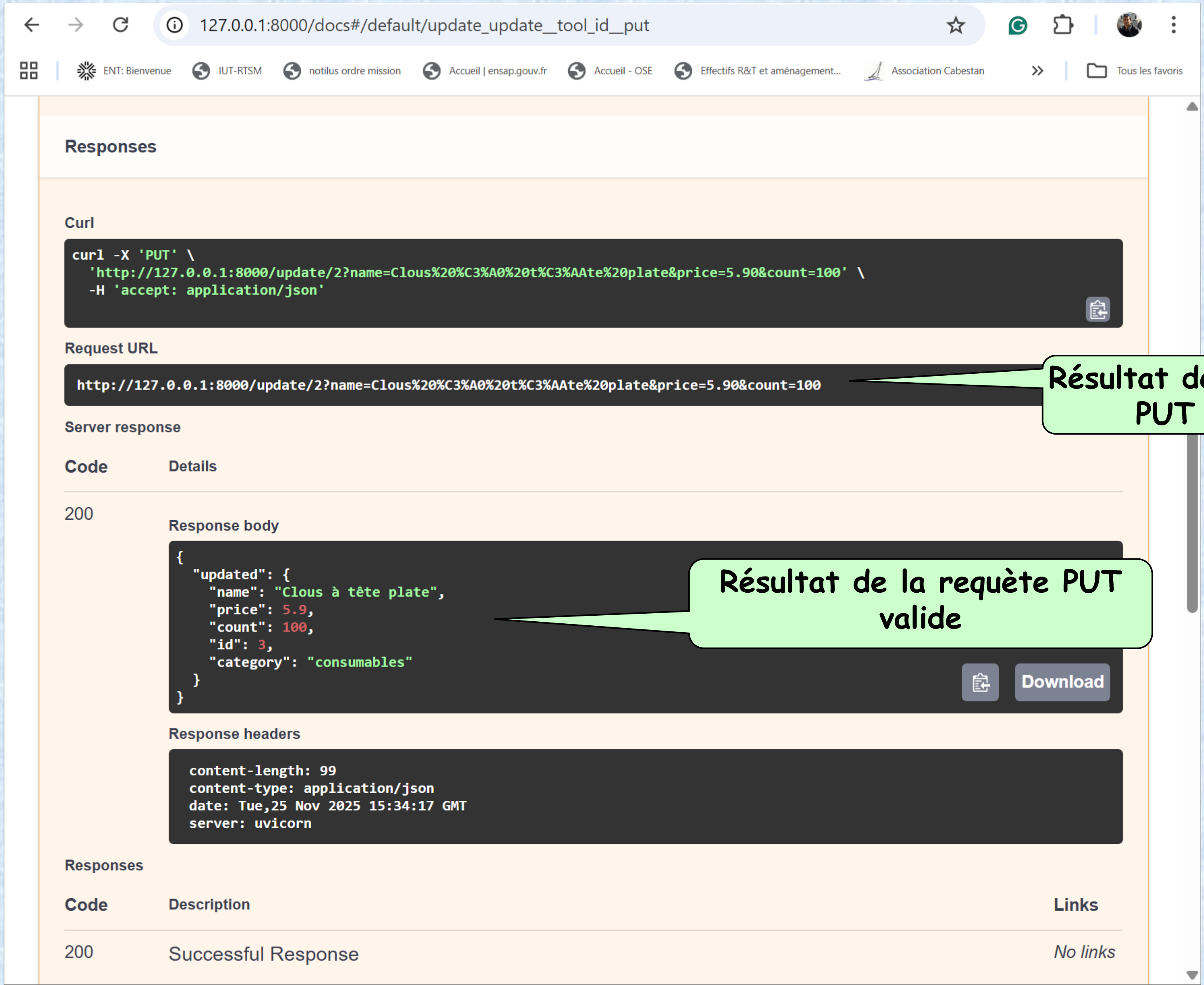
Code	Description	Links

- Requète PUT http://127.0.0.1:8000/docs#/

☞ Créer un autre programme *main6.py* pour récupérer des requêtes

main6.py

19/24


127.0.0.1:8000/docs#/default/update_update__tool_id__put
ENT: Bienvenue
IUT-RTSM
notilus ordre mission
Accueil | ensap.gouv.fr
Accueil - OSE
Effectifs R&T et aménagement...
Association Cabestan
Tous les favoris
Résultat de la requète PUT valide
Résultat de la requète PUT valide
Download

Responses

Curl

```
curl -X 'PUT' \
  'http://127.0.0.1:8000/update/2?name=Clous%20%C3%A0%20t%C3%AAte%20plate&price=5.90&count=100' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/update/2?name=Clous%20%C3%A0%20t%C3%AAte%20plate&price=5.90&count=100
```

Server response

Code Details

200

Response body

```
{
  "updated": {
    "name": "Clous à tête plate",
    "price": 5.9,
    "count": 100,
    "id": 3,
    "category": "consumables"
  }
}
```

Response headers

```
 content-length: 99
 content-type: application/json
 date: Tue,25 Nov 2025 15:34:17 GMT
 server: uvicorn
```

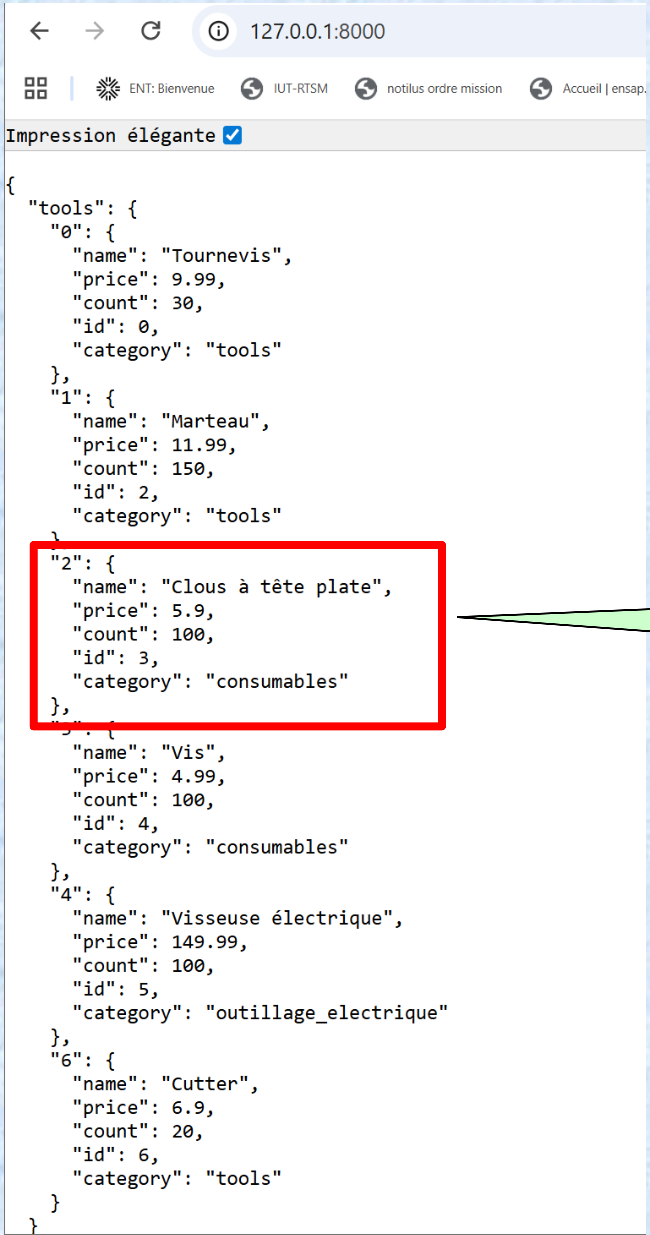
Responses

Code	Description	Links
200	Successful Response	*No links*

PUT

- **Requète PUT**

☞ **Créer un autre programme *main6.py* pour récupérer des requêtes**



Impression élégante ☑

```
{
  "tools": {
    "0": {
      "name": "Tournevis",
      "price": 9.99,
      "count": 30,
      "id": 0,
      "category": "tools"
    },
    "1": {
      "name": "Marteau",
      "price": 11.99,
      "count": 150,
      "id": 2,
      "category": "tools"
    },
    "2": {
      "name": "Clous à tête plate",
      "price": 5.9,
      "count": 100,
      "id": 3,
      "category": "consumables"
    },
    "3": {
      "name": "Vis",
      "price": 4.99,
      "count": 100,
      "id": 4,
      "category": "consumables"
    },
    "4": {
      "name": "Visseuse électrique",
      "price": 149.99,
      "count": 100,
      "id": 5,
      "category": "outillage_electrique"
    },
    "6": {
      "name": "Cutter",
      "price": 6.9,
      "count": 20,
      "id": 6,
      "category": "tools"
    }
  }
```

http://127.0.0.1:8000/

main6.py

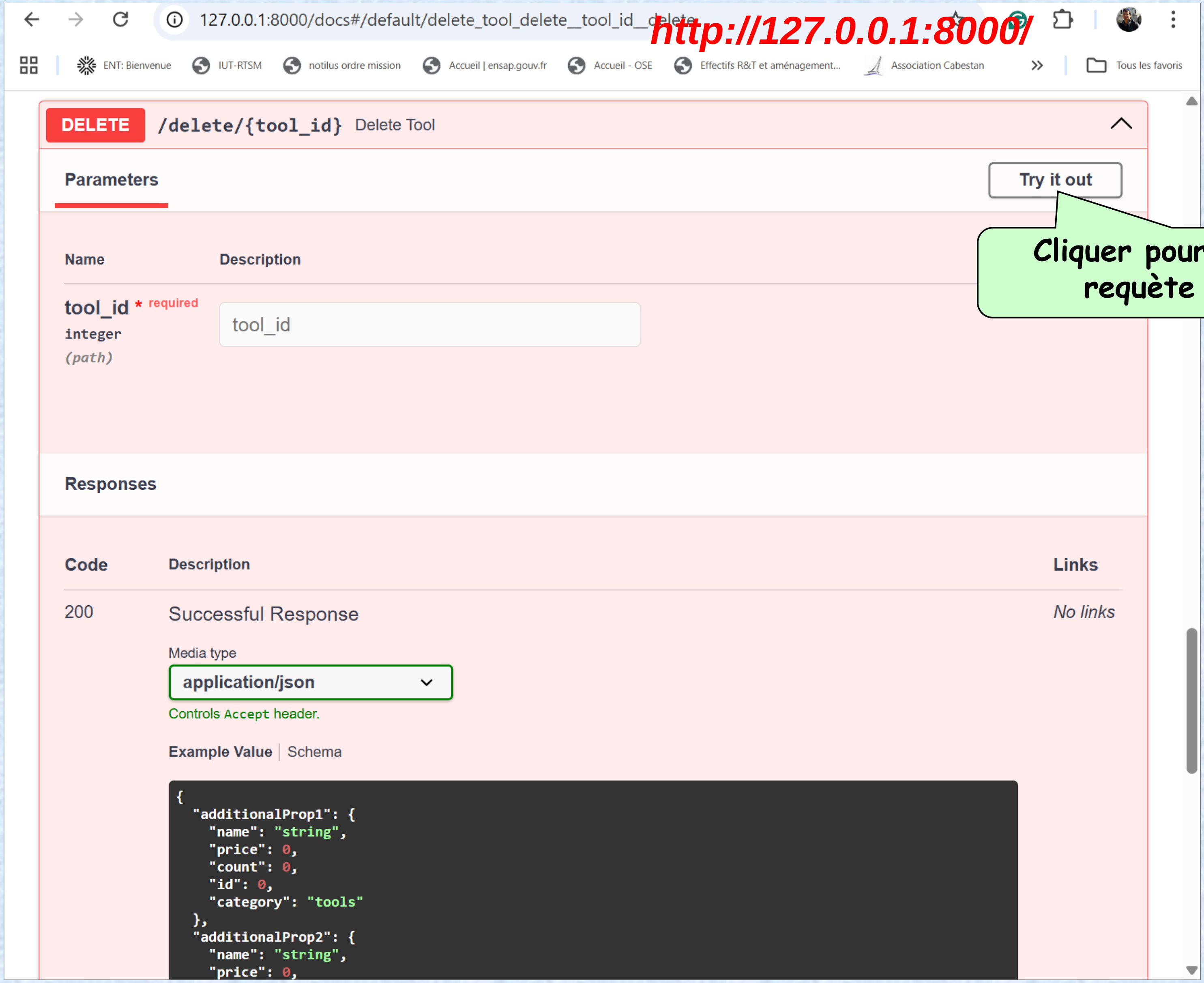
20/24

Résultat de la requète PUT valide

FastAPI **DELETE** 6.1

- **Requète DELETE**

☞ **Créer un autre programme *main6.py* pour récupérer des requêtes**



127.0.0.1:8000/docs#/default/delete_tool_delete__tool_id__delete

http://127.0.0.1:8000/

ENT: Bienvenue | IUT-RTSM | notilus ordre mission | Accueil | ensap.gouv.fr | Accueil - OSE | Effectifs R&T et aménagement... | Association Cabestan | Tous les favoris

DELETE `/delete/{tool_id}` Delete Tool

Parameters — Try it out

Cliquer pour exécuter la requète DELETE.

Name	Description
tool_id * required integer (path)	tool_id

Responses

Code	Description	Links
200	Successful Response	*No links*

Media type: application/json

Controls `Accept` header.

Example Value | Schema

```
{
  "additionalProp1": {
    "name": "string",
    "price": 0,
    "count": 0,
    "id": 0,
    "category": "tools"
  },
  "additionalProp2": {
    "name": "string",
    "price": 0,
```

main6.py

21/24

FastAPI **DELETE** **6.1**

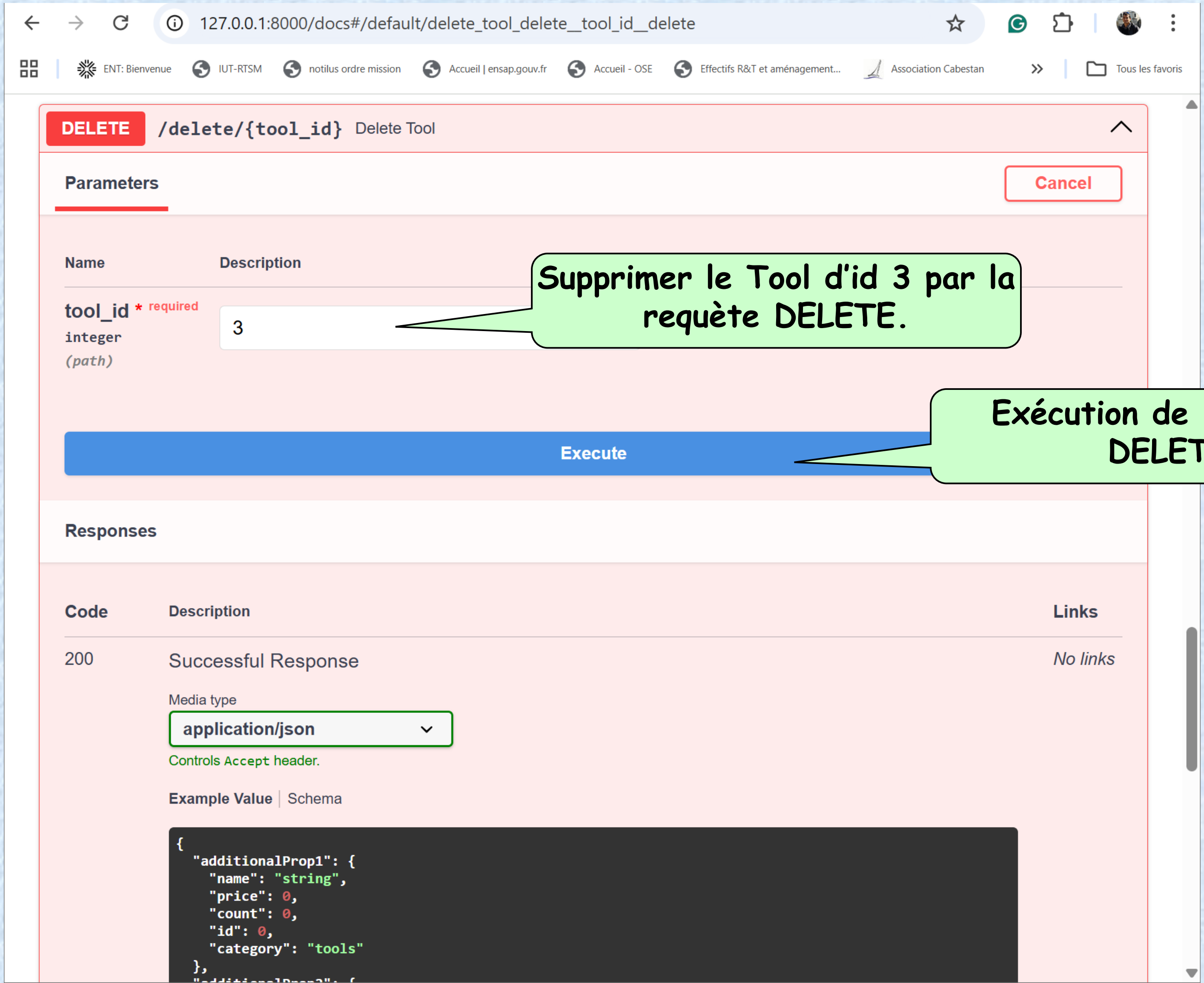
- **Requête DELETE**

☞ **Créer un autre programme *main6.py* pour récupérer des requêtes**

http://127.0.0.1:8000/

main6.py

22/24



DELETE `/delete/{tool_id}` Delete Tool

Parameters

Name	Description
tool_id * required integer *(path)*	3

Supprimer le Tool d'id 3 par la requête DELETE.

Exécution de la requête DELETE.

Responses

Code	Description	Links
200	Successful Response	*No links*

Media type

application/json

Controls `Accept` header.

Example Value | Schema

```
{
  "additionalProp1": {
    "name": "string",
    "price": 0,
    "count": 0,
    "id": 0,
    "category": "tools"
  },
```

DELETE

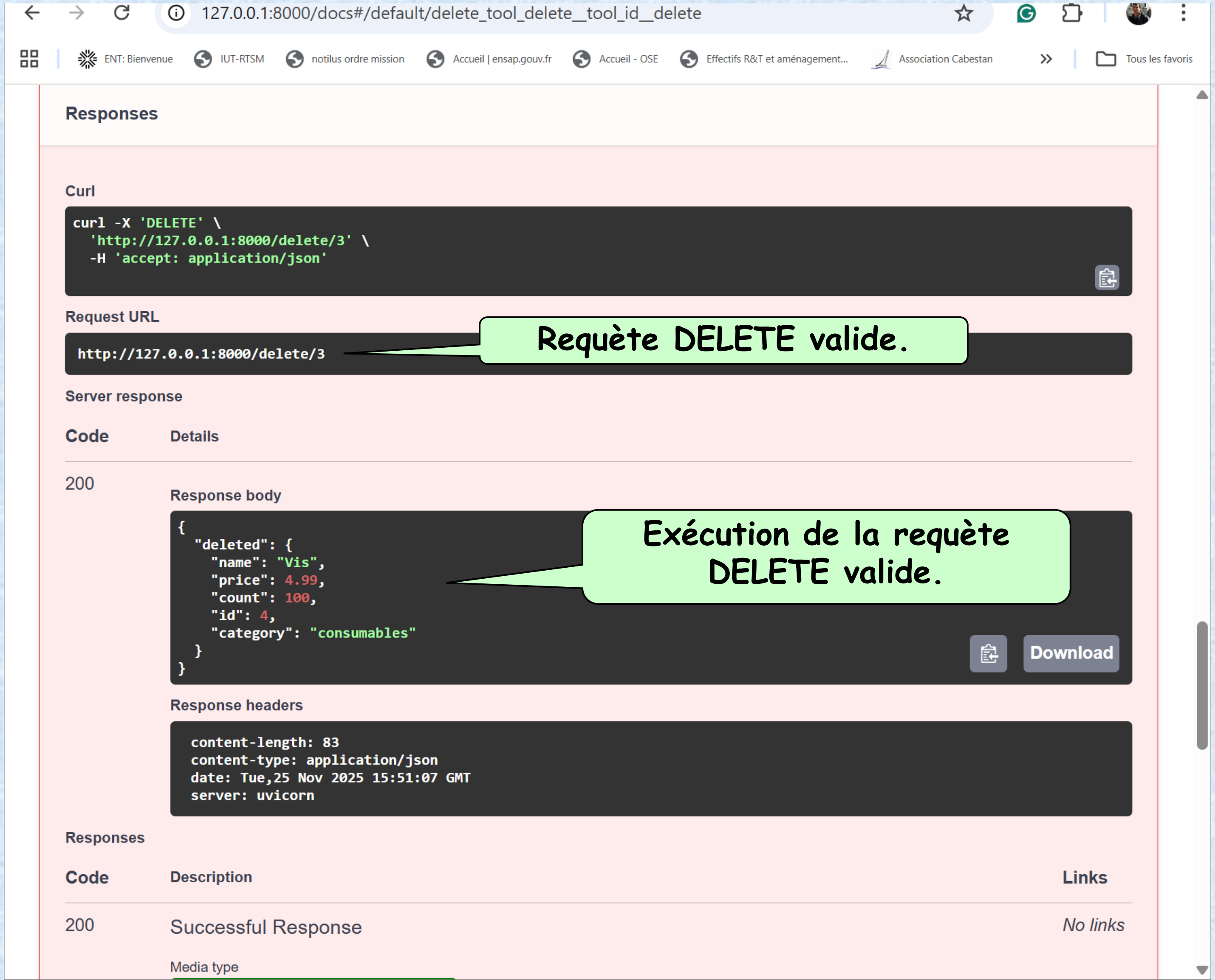
- **Requête DELETE**

☞ **Créer un autre programme *main6.py* pour récupérer des requêtes**

http://127.0.0.1:8000/

main6.py

23/24



Responses

Curl

```
curl -X 'DELETE' \
  'http://127.0.0.1:8000/delete/3' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/delete/3
```

Server response

Code Details

200

Response body

```
{
  "deleted": {
    "name": "Vis",
    "price": 4.99,
    "count": 100,
    "id": 4,
    "category": "consumables"
  }
}
```

Response headers

```
content-length: 83
content-type: application/json
date: Tue,25 Nov 2025 15:51:07 GMT
server: uvicorn
```

Responses

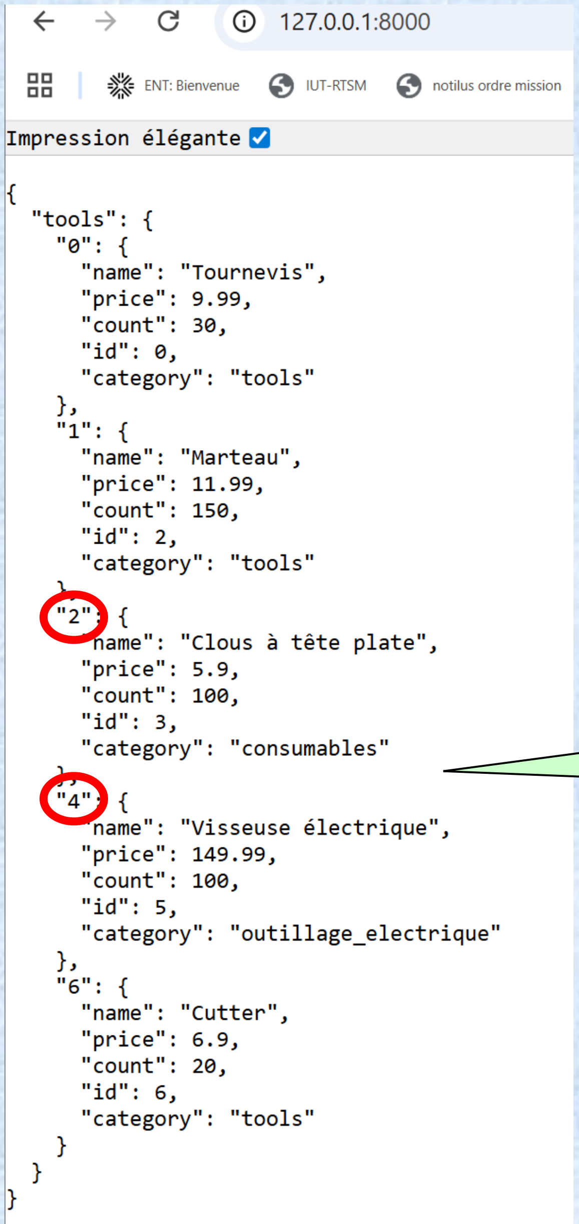
Code	Description	Links
200	Successful Response	No links

Media type

DELETE

- **Requête DELETE**

☞ **Créer un autre programme *main6.py* pour récupérer des requêtes**



127.0.0.1:8000

ENT: Bienvenue IUT-RTSM notilus ordre mission

Impression élégante ✓

```
{
  "tools": {
    "0": {
      "name": "Tournevis",
      "price": 9.99,
      "count": 30,
      "id": 0,
      "category": "tools"
    },
    "1": {
      "name": "Marteau",
      "price": 11.99,
      "count": 150,
      "id": 2,
      "category": "tools"
    },
    "2": {
      "name": "Clous à tête plate",
      "price": 5.9,
      "count": 100,
      "id": 3,
      "category": "consumables"
    },
    "4": {
      "name": "Visseuse électrique",
      "price": 149.99,
      "count": 100,
      "id": 5,
      "category": "outillage_electrique"
    },
    "6": {
      "name": "Cutter",
      "price": 6.9,
      "count": 20,
      "id": 6,
      "category": "tools"
    }
  }
}
```

http://127.0.0.1:8000/

main6.py

24/24

Suppression du Tool d'id 3 par la requète DELETE.

GET

- Requète GET

☞ Créer un autre programme *main7.py* pour récupérer des requêtes

main7.py

1/5

Définition de méthode associée à la requête DELETE

```
@app.delete("/delete/{tool_id}")
def delete_tool(tool_id: int) -> dict[str, Tool]:
    if tool_id not in tools:
        raise HTTPException(
            status_code=404, detail=f"Tool with {tool_id=} does not exist."
        )

    tool = tools.pop(tool_id)
    return {"deleted": tool}
```