
Auteur: CAMARA Laby Damaro

Email: ldamaro98@gmail.com

github: <https://github.com/camara94>

Création API REST en Python

Dans ce tuto, nous allons réaliser une API RESTFULL en qui va exploiter notre algorithme de synthèse automatique que nous avons réalisé dans les tuto précédents.

Installer Python

Si vous avez pas python, pouvez l'installer à travers ce [lien](#).
Je vous recommande la version 3.x.x surtout.

Sous Windows, cliquez sur l'icône du menu Démarrer et tapez `cmd` dans la zone de recherche, puis appuyez sur **Entrée**.

Une fois votre ligne de commande ouverte, saisissez ces commandes :

```
python --version  
pip --version
```

Si la sortie de ces commandes inclut un numéro de version, Python est installé et disponible à partir de la ligne de commande et vous pouvez passer à l'étape suivante.

Installer flask

```
pip install flask --user
```

Créer un projet

Sous Windows, vous pouvez créer le dossier `api` avec ces commandes dans votre environnement de ligne de commande `cmd` :

```
md api
cd api
```

Installer flask-cors

```
pip install -U flask-cors
```

Ensuite, ouvrez un éditeur de texte (tel que VSCODE ou Notepad++), créer un fichier `api.py` et entrez le code suivant :

Importer les packages suivants

```
import flask
from flask import request, jsonify
from synthese import *
from flask_cors import CORS, cross_origin
```

Créer un endpoint

Cet endpoint, va me permettre de prendre l'URL d'un article retourner l'article et son résumé.

```
@app.route('/api', methods=['GET'])
@cross_origin()
def api_url():
    # Check if an URL was provided as part of the
    URL.
    # If URL is provided, assign it to a variable.
    # If no URL is provided, display an error in
    the browser.
    if 'url' in request.args:
        url = str(request.args['url'])
        return synthese_automatique(url)
    else:
        return "Error: No id field provided. Please
        specify an url."
```

Exécuter

```
python api.py
```

Demo

```
* Serving Flask app "api" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with windowsapi reloader
2021-10-16 02:58:44.781388: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cuda
rt64_110.dll'; dLError: cuda64_110.dll not found
2021-10-16 02:58:44.781995: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have
a GPU set up on your machine.
* Debugger is active!
* Debugger PIN: 234-411-055
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Appelle de notre à travers l'objet XMLHttpRequest

```
<script>
    var bouton =
document.getElementById('btn');
    bouton.addEventListener( 'click', (e)
=> {
        var param =
document.getElementById('url').value;
        getContent(param);
    } );
    const getContent = (param) => {
        var url =
"http://127.0.0.1:5000/api?url=" + param;
        var oReq = new XMLHttpRequest();
        var div =
document.getElementById('load');
        var image =
document.createElement('img')
        oReq.addEventListener('progress',
(e) => {
            image.setAttribute('src',
'loading-buffering.gif');
            div.appendChild(image);
        })
        oReq.addEventListener("load", () =>
{
            var reponse =
document.getElementById('reponse');
            var article =
```

```

document.getElementById('article');
        reponse.innerHTML="";
        article.innerHTML=""
        if (oReq.status != 200) {
// analyze HTTP status of the response
            alert(`Error
${oReq.status}: ${oReq.statusText}`); // e.g. 404:
Not Found

            } else { // show the result
                //alert(`Done, got
${oReq.responseText} bytes`);
                var rep =
JSON.parse(oReq.responseText);
                setTimeout(()=>{
                    div.removeChild(image)
                }, 2000);
                var titre =
document.createElement('h3');
                titre.style.margin =
'0px auto';
                var titreText =
document.createTextNode('Le résumé de l\'article');
                titre.appendChild(titreText);
                var para =
document.createElement('p');
                var parText =
document.createTextNode(rep.resume);
                para.appendChild(parText);
                var titre2 =
document.createElement('h3');
                titre2.style.margin =
'0px auto';
                var titreText2 =
document.createTextNode('L\'article original');
                titre2.appendChild(titreText2);
                var para2 =
document.createElement('p');
                var parText2 =
document.createTextNode(rep.article);
                para2.appendChild(parText2);
                reponse.appendChild(titre);
                reponse.appendChild(para);

```

```
article.appendChild(titre2);  
article.appendChild(para2);  
    }  
    }  
    );  
    oReq.open("GET", url);  
    oReq.send()  
}  
</script>
```