INSTITUT SUPERIEUR D'INFORMATIQUE

ET DES TECHNIQUES DE COMMUNICATION – HAMMAM SOUSSE

المعهد العالي للإعلامية وتقنيات الاتصال بحمام سوسة

Projet de la nuit d'info

Thème : Développement

Savez-vous convertir un PDF en HTML ?

Saurez-vous relever le défi ? A vous de jouer !

Réalisé par :

Unbroken_ISITCom

Proposé par :

Année universitaire : 2021/2022

Institut Supérieur d'Informatique et des Techniques de Communication Hammam Sousse

ISITCOM

Tél/Fax : +216 73 37 15 71 / +216 73 36 44 11

Remerciements

Au terme de ce projet, nos vifs remerciements sont dédiés à tous ceux qui ont contribué directement ou indirectement à l'élaboration de ce projet. Nous tenons à remercier aussi tous nos professeurs pour leurs conseils et leurs remarques constructives tout au long la réalisation de ce projet. Nous n'oublions pas de souligner que ce travail n'aurait jamais pu voir le jour sans le savoir-faire acquis dans notre honorable institut : l'Institut Supérieur d'Informatique et des Techniques de Communication de Hammam Sousse.
C'est donc avec une immense fierté que nous adressons nos profonds remerciements à l'intégralité des membres de notre institut.

Présentation du projet

Introduction :

Les développeurs en réalisant leurs projets surtout les débutants veulent toujours faire des tâches mais avec peu de connaissances ils les ressentent difficiles et impossibles. Et cette difficulté les bloque dans la réalisation de leurs applications web par exemple. Du coup ils ont recours parfois à des API (Interface de Programmation d'Application) déjà créées qui leur permettent de réaliser cette fonction. Par les fonctionnalités cherchées on trouve le fusionnement des articles en format PDF dans le code.

Problématique :

Cette fonctionnalité utilisé dans la réalisation de différents types de projet nous mène à nous poser la question comment réaliser une API qui convertit un document PDF en HTML.

1. Contexte du projet :

Dans cette partie, la résolution de notre problématique se met en avant par le contexte le défi de la nuit d'info 2022 proposée par la société LISIO lors de la nuit d'info 2022. C'est une entreprise qui propose une solution web d'inclusion et d'écologie numérique créée par Eric Gayraud. C'est un projet qui vise à concevoir et développer une API qui prend l'url d'un PDF en paramètre et retourne la structure HTML de ce fichier contenant tous ses composants tels que ses textes, images, liens, tableaux, … .Cette conversion doit se faire sans avoir aucune perte de l'état initial des composants.

2. Description du projet :

Notre défi qui est la réalisation d'une API qui permet à son consommateur de pour convertir un document PDF qui sera passé en paramètre en un fichier HTML pour pouvoir le fusionner au sein de notre code facilement. Et cette création s'est faite en plusieurs parties et en ayant recours à différentes bibliothèques et fonctionnalités. Tout d'abord on a utilisé la bibliothèque « PdfToHtml » qui est une commande du système d'exploitation Linux. Cette commande permet de convertir un PDF donné en une page HTML. Après, on a utilisé la fonction « exec » qui est une fonction Node qu'on a utilisé et qui permet à Node d'exécuter des commandes sur le système d'exploitation Linux.

Notre API prend comme point d'entrée cette Url : « localhost:5000/api/tohtmlloca » qui permet de consommer le service. Il ne faut pas oublier qu'il faut installer la bibliothèque « PdfToHtml ». On trouve après comme paramètre (« user_id »: « Url du PDF », "filename": "nom du fichier").

Dans la figure suivante nous présentons la description des commandes.

```
Install  popular utils (HTMLTOPDF) to linux CMD

 you see the basic command that you will be able to convert your PDF file to HTML. Now open a terminal in the directory where you have saved or stored your PDF document.

 pdftohtml -s docName filename.html

 then we use exec
 => (const { exec } = require('child_process');)

 to make cmd with code nodeJS

 so we create api with expressJS that get URL PDF from FRONT (Postman)
 then exec the cmd above so

 the steps is API POST

 1 ) cd upload
 2 ) mkdir " + filename + (create folder with file name under upload)
 3 ) cd " + filename (point under filename folder)
 4 ) wget -c " + user_id + " -O " + filename + ".pdf" (upload pdf file and rename with O)
 5 ) pdftohtml -s " + filename + ".pdf output.html"  (run pdftohtml cmd )
 6 ) return path to html file

the steps to GET html file

1 ) GET get('/gethtml/:filename/:file'
2 ) filename (the folder name)
3 ) file the name of each files that navigator needed to show files
```

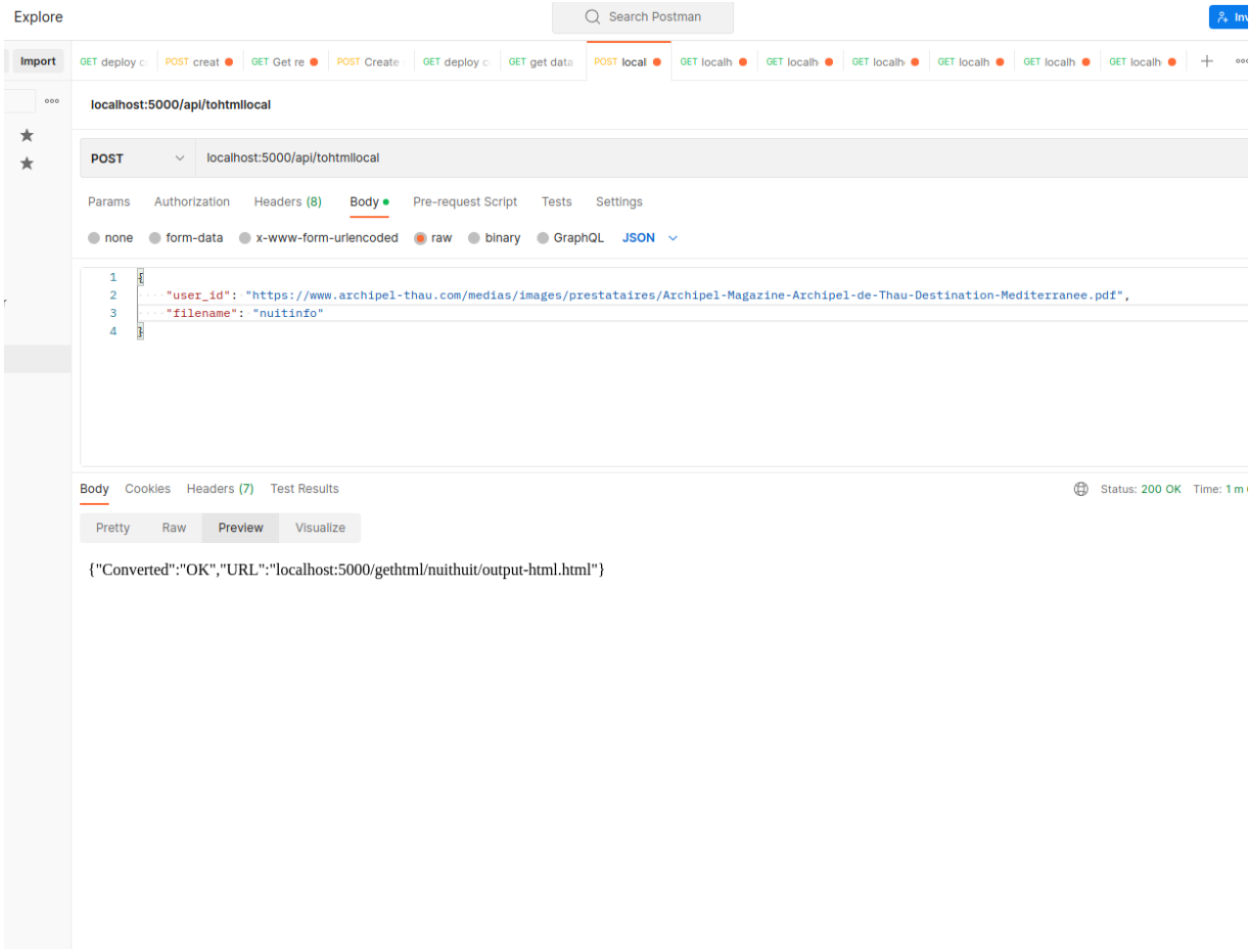
Dans la figure suivante nous présentons le code source de notre API.

index.js

index.js > app.post('/api/tohtmllocal') callback > exec() callback

```
const path = require('path');
const { exec } = require('child_process');

app.use(express.json());
app.use(express.static('public'));
app.use(express.static('upload'));


app.get('/gethtml/:filename/:file', function(req, res){

    console.log(req.params.filename);

    res.sendFile(path.join(__dirname + "/upload/"+req.params.filename+"/" + req.params.file));

});

app.post('/api/tohtmllocal', function(req, res) {

    const user_id = req.body.user_id;
    const filename = req.body.filename;



    const cmdDownload = "cd upload && mkdir " + filename + " && cd " + filename + " && wget -c " + user_id + " -O " + filename + ".pdf";
    const cmdConver = " && pdftohtml -s " + filename + ".pdf output.html"
    const cmdFinal = cmdDownload + cmdConver;
    console.log(cmdFinal);

exec(cmdFinal , (err, stdout, stderr) => {
  if (err) {
    // node couldn't execute the command
    return;
  }


  res.send({"Converted":"OK","URL": "localhost:5000/gethtml/"+filename+"/output-html.html"});


  // the *entire* stdout and stderr (buffered)
  console.log();(`stdout: ${stdout}`);
  console.log();(`stderr: ${stderr}`);
});


});

```

Dans la figure suivante nous présentons la consommation de notre API.



les URL obtenus sont :

localhost:5000/gethtml/PDF1/output-html.html

localhost:5000/gethtml/PDF2/output-html.html

localhost:5000/gethtml/PDF3/output-html.html

localhost:5000/gethtml/PDF4/output-html.html

On peut tester manuellement les résultats dans le dossier « upload » si on clique sur output-html.html .

run code avec > node index.js

Et utiliser le postman avec l'url de collection suivante
https://www.getpostman.com/collections/82b1c82193046a634921

N.B : Voici le lien suivant pour télécharger la bibliothèque dans linux (ubuntu).

https://updf.com/convert-pdf/convert-pdf-to-html/

3. Les perspectives :

Bien évidemment notre API est apte à être améliorer si on avait plus de temps. Son amélioration consiste à rendre le résultat de la conversion plus vaste. C'est-à-dire convertir un PDF en une image, un fichier Power Point et d'autres.

Conclusion :

En guise de conclusion, notre défi a consisté à la réalisation d'une API consommable qui permet de convertir un document PDF en fichier HTML. Et cette API sera très utile en la consommant pour les développeurs et leur fera gagner beaucoup de temps.