NOTE DE STAGE 2021

CHEZ **COFLINE**

SOMMAIRE

GLOSSAIRE

- **DOM: D**ocument **O**bject **M**odel
- **VSC: V**isual **S**tudio **C**ode
- **NPM: N**ode **P**ackage **M**anager
- **JSON: J**ava**S**cript **O**bject **N**otation
- **API: A**pplication **P**rogramming **I**nterface
- **HTTP: H**yper**T**ext **T**ransfer **P**rotocol
- **XML: Ex**tensible **M**arkup **L**anguage

LUNDI 14 JUIN 2021

MISSIONS :

Pause sur les activités *ReactJS* en rapport avec le projet. L'initiation a *ReactJS* étant assez avancé, deux groupes ont été formés, un groupe pour produire un clone de Google Docs et un pour produire un clone de Google Drive.

1. Faire des recherches pour cloner Google Docs. Faire de la ***conception*** : **Quelle technologie ? Pourquoi ? Comment ?**

REALISATIONS :

1.
    a. Recherches

Notre maître de stage nous avait vaguement évoqué **Socket.IO** et **MongoDB**, je suis donc partie dans cette direction pour mes recherches. J'ai fini par trouver un tutoriel assez complet sur la chaîne YouTube **Web Dev Simplified**[1].

Résumé des recherches sur les technologies employées :

Quill[2] s'initialise avec un élément du **DOM** (un **objet Quill**) qui contiendra alors l'éditeur de texte. Il est surtout reconnu pour sa flexibilité et son extensibilité (personnalisation). On peut l'installer sur ***VSC*** grâce à ***NPM*** (dépendance) et l'ajouter à notre projet. Ce qui nous intéresse particulièrement avec cette dépendance ce sont les **Deltas**, compatible **JSON**, ils représentent les changements apportés à un document et permettent d'organiser/diviser les documents Quill (le texte, la mise en forme…) sans ambiguïté et sans complexifier la structure HTML. On enregistrera les Deltas (les changements donc) au lieu d'enregistrer tout le document à chaque fois. Quill est open source et gratuit, les alternatives telles que **Froala Editor** et **TinyMCE** sont payants. (Les Deltas sont une librairie à part entière de Quill donc possiblement utilisable avec d'autre éditeur de texte mais plus compliqué).

Socket.IO[3] est une librairie qui permet une **communication bidirectionnelle en temps réel** et basé sur des **évènements** entre le serveur et l'application. On obtient aussi un **objet socket** qui étend la classe **EventEmitter** de Node.js.

UUID[4] nous permet de générer automatiquement un identifiant unique ***RFC***, qui nous permettra d'identifier chaque document et de les isoler (**Socket.IO**) afin que les données d'un utilisateur ne puisse pas être envoyées à un autre.

MongoDB[5] est une base de données « document » non relationnelle (**NoSQL**), elle conserve les données dans des documents similaire **JSON.** On peut donc utiliser du langage **JQuery** qui nous permettra de faire énormément d'opération de manière plus naturelle et plus simple qu'avec le SQL.

---

[1] https://www.youtube.com/watch?v=iRaelG7v0OU

[2] https://quilljs.com/docs/quickstart/

[3] https://socket.io/docs/v4

[4] https://www.npmjs.com/package/uuid

[5] https://www.mongodb.com/

b. Application (avec tutoriel)

```
EXPLORER
GOOGLE-DOCS-CLONE
  client
    node_modules
    public
    src
      App.js
      index.js
      styles.css
      TextEditor.js
    .gitignore
    package-lock.json
    package.json
    README.md
  server
    node_modules
    Document.js
    package-lock.json
    package.json
    server.js
  package-lock.json
```

App.js | index.js | TextEditor.js | styles.css | server.js | Document.js

client > src > TextEditor.js > TextEditor > useEffect() callback > handler

```
67          }
68      }, [socket,quill])
69
70      useEffect(() => {
71          if(socket == null || quill == null) return
72
73          const handler = (delta, oldDelta, source)=> {
74              if (source !== "user") return
75              //cela nous permet de s'assurer que c'est bien l'utilisateur qui a fait ces changements
76              socket.emit("send-changes", delta)
77          }
78          quill.on("text-change", handler)
79          return () => {
80              quill.off("text-change",handler)
81          }
82      }, [socket,quill])
83
84      const wrapperRef = useCallback(wrapper => {
85          if (wrapper == null) return
86
87          wrapper.innerHTML = ""
88          const editor = document.createElement("div")
89          wrapper.append(editor)
90          const q = new Quill(editor, {
91              theme: "snow",
92              modules: { toolbar: TOOLBAR_OPTIONS},
93          })
94          q.disable()
95          q.setText('Loading ...')
96          setQuill(q)
97
98      }, [])
99
100     return <div className="container" ref={wrapperRef}></div>
101
102 }
103
```

```
EXPLORER
GOOGLE-DO...
  client
    node_modules
    public
    src
      App.js
      index.js
      styles.css
      TextEditor.js
    .gitignore
    package-lock.json
    package.json
```

App.js | index.js | TextEditor.js | styles.css

client > src > index.js

```
1   import React from 'react';
2   import ReactDOM from 'react-dom';
3   import App from './App';
4   import "./styles.css"
5
6   ReactDOM.render(
7     <React.StrictMode>
8       <App />
9     </React.StrictMode>,
10    document.getElementById('root')
11  );
12
13
```

EXPLORER

GOOGLE-DOCS-CLONE
- client
  - node_modules
  - public
  - src
    - App.js
    - index.js
    - styles.css
    - TextEditor.js
  - .gitignore
  - package-lock.json
  - package.json
  - README.md
- server
  - node_modules
  - Document.js
  - package-lock.json
  - package.json
  - server.js
- package-lock.json

App.js | index.js | TextEditor.js | styles.css | server.js | Doc

client > src > TextEditor.js > TextEditor

```javascript
import { useCallback, useEffect, useState } from 'react'
import Quill from "quill"
import "quill/dist/quill.snow.css"
import {io} from "socket.io-client"
import { useParams } from "react-router-dom"

const SAVE_INTERVAL_MS = 2000
const TOOLBAR_OPTIONS = [
    [{header: [1,2,3,4,5,6, false]}],
    [{font: []}],
    [{list: "ordered"},{list: "bullet"}],
    ["bold","italic","underline"],
    [{color:[]},{background: []}],
    [{script:"sub"},{script: "super"}],
    [{align: []}],
    ["image","blockquote","code-block"],
    ["clean"],
]

export default function TextEditor() {
    const { id: documentId } = useParams()
    const [socket, setSocket] = useState()
    const [quill, setQuill] = useState()

    useEffect(() => {
        const s = io("http://localhost:3001")
        setSocket(s)

        return () => {
            s.disconnect()
        }
    }, [])

    useEffect(() => {
        if (socket == null || quill == null) return

        socket.once("load-document", document => {
            quill.setContents(document)
            quill.enable()
        })

        socket.emit("get-document", documentId)

    }, [socket,quill,documentId])
```

EXPLORER

GOOGLE-DOCS-CLONE
- client
  - node_modules
  - public
  - src
    - App.js
    - index.js
    - styles.css
    - TextEditor.js
  - .gitignore
  - package-lock.json
  - package.json
  - README.md
- server
  - node_modules
  - Document.js
  - package-lock.json
  - package.json
  - server.js
- package-lock.json

App.js | index.js | TextEditor.js | styles.css | server.js | Document.js

client > src > TextEditor.js > TextEditor

```
45
46  useEffect(() => {
47      if (socket == null || quill == null) return
48
49      const interval = setInterval(() => {
50        socket.emit("save-document", quill.getContents())
51      }, SAVE_INTERVAL_MS)
52
53      return () => {
54        clearInterval(interval)
55      }
56    }, [socket, quill])
57
58  useEffect(() => {
59      if(socket == null || quill == null) return
60
61      const handler = delta => {
62          quill.updateContents(delta)
63      }
64      socket.on("receive-changes", handler)
65      return () => {
66          socket.off("receive-changes",handler)
67      }
68  }, [socket,quill])
69
70  useEffect(() => {
71      if(socket == null || quill == null) return
72
73      const handler = (delta, oldDelta, source)=> {
74          if (source !== "user") return
75          //cela nous permet de s'assurer que c'est bien l'utilisateur qui a fait ces changements
76          socket.emit("send-changes", delta)
77      }
78      quill.on("text-change", handler)
79      return () => {
80          quill.off("text-change",handler)
81      }
82  }, [socket,quill])
83
```

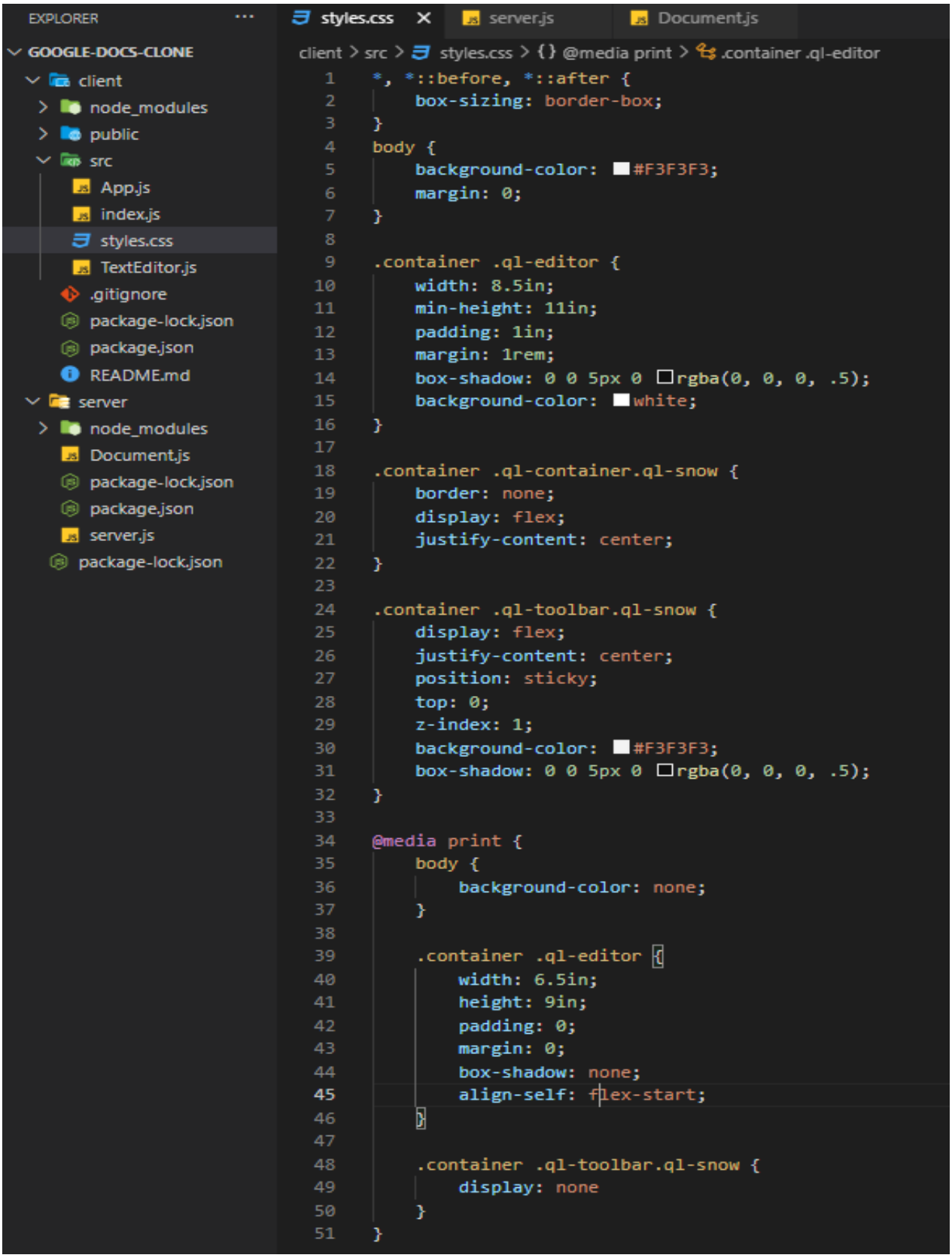
EXPLORER

GOOGLE-DOCS-CLONE
- client
  - node_modules
  - public
  - src
    - App.js
    - index.js
    - styles.css
    - TextEditor.js
  - .gitignore
  - package-lock.json
  - package.json
  - README.md
- server
  - node_modules
  - Document.js
  - package-lock.json
  - package.json
  - server.js
- package-lock.json

App.js | index.js | TextEditor.js | styles.css | server.js

client > src > TextEditor.js > ...

```
83
84      const wrapperRef = useCallback(wrapper => {
85          if (wrapper == null) return
86
87          wrapper.innerHTML = ""
88          const editor = document.createElement("div")
89          wrapper.append(editor)
90          const q = new Quill(editor, {
91              theme: "snow",
92              modules: { toolbar: TOOLBAR_OPTIONS},
93          })
94          q.disable()
95          q.setText('Loading ...')
96          setQuill(q)
97
98      }, [])
99
100     return <div className="container" ref={wrapperRef}></div>
101
102 }
103
```


EXPLORER
styles.css
server.js
Document.js
GOOGLE-DOCS-CLONE
client
node_modules
public
src
App.js
index.js
styles.css
TextEditor.js
.gitignore
package-lock.json
package.json
README.md
server
node_modules
Document.js
package-lock.json
package.json
server.js
package-lock.json
client > src > styles.css > {} @media print > .container .ql-editor
*, *::before, *::after {
box-sizing: border-box;
}
body {
background-color: #F3F3F3;
margin: 0;
}
.container .ql-editor {
width: 8.5in;
min-height: 11in;
padding: 1in;
margin: 1rem;
box-shadow: 0 0 5px 0 rgba(0, 0, 0, .5);
background-color: white;
}
.container .ql-container.ql-snow {
border: none;
display: flex;
justify-content: center;
}
.container .ql-toolbar.ql-snow {
display: flex;
justify-content: center;
position: sticky;
top: 0;
z-index: 1;
background-color: #F3F3F3;
box-shadow: 0 0 5px 0 rgba(0, 0, 0, .5);
}
@media print {
body {
background-color: none;
}
.container .ql-editor {
width: 6.5in;
height: 9in;
padding: 0;
margin: 0;
box-shadow: none;
align-self: flex-start;
}
.container .ql-toolbar.ql-snow {
display: none
}
}

EXPLORER

GOOGLE-DOCS-CLONE
- client
  - node_modules
  - public
  - src
    - App.js
    - index.js
    - styles.css
    - TextEditor.js
  - .gitignore
  - package-lock.json
  - package.json
  - README.md
- server
  - node_modules
  - Document.js
  - package-lock.json
  - package.json
  - server.js
- package-lock.json

styles.css | server.js | Document.js

server > server.js > io > cors

```
1  const mongoose = require("mongoose")
2  const Document = require("./Document")
3
4  mongoose.connect("mongodb://localhost/google-docs-clone", {
5    useNewUrlParser: true,
6    useUnifiedTopology: true,
7    useFindAndModify: false,
8    useCreateIndex: true,
9  })
10
11 const io = require("socket.io")(3001, {
12   cors: {
13     origin: "http://localhost:3000",
14     methods: ["GET", "POST"],
15   },
16 })
17
18 const defaultValue = ""
19
20 io.on("connection", socket => {
21   socket.on("get-document", async documentId => {
22     const document = await findOrCreateDocument(documentId)
23     socket.join(documentId)
24     socket.emit("load-document", document.data)
25
26     socket.on("send-changes", delta => {
27       socket.broadcast.to(documentId).emit("receive-changes", delta)
28     })
29
30     socket.on("save-document", async data => {
31       await Document.findByIdAndUpdate(documentId, { data })
32     })
33   })
34 })
35
36 async function findOrCreateDocument(id) {
37   if (id == null) return
38
39   const document = await Document.findById(id)
40   if (document) return document
41   return await Document.create({ _id: id, data: defaultValue })
42 }
43
```

EXPLORER

GOOGLE-DO...
- client
  - node_modules
  - public
  - src
    - App.js
    - index.js
    - styles.css
    - TextEditor.js
  - .gitignore

styles.css | server.js | Document.js

server > Document.js > <unknown>

```
1  const { Schema, model } = require("mongoose")
2
3  const Document = new Schema({
4    _id: String,
5    data: Object,
6  })
7
8  module.exports = model("Document", Document)
```

EXPLORER

GOOGLE-DO...

- client
  - node_modules
  - public
  - src
    - App.js
    - index.js
    - styles.css
    - TextEditor.js
  - .gitignore
  - package-lock.json
  - package.json
  - README.md
- server
  - node_modules
  - Document.js
  - package-lock.json
  - package.json
  - server.js
- package-lock.json

styles.css | server.js | Document.js | package.json

server > package.json > {} devDependencies > socket.io

```
1  {
2    "name": "server",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
       Debug
6    "scripts": {
7      "devStart": "nodemon server.js",
8      "test": "echo \"Error: no test specified\" && exit 1"
9    },
10   "keywords": [],
11   "author": "",
12   "license": "ISC",
13   "dependencies": {
14     "install": "^0.13.0",
15     "mongoose": "^5.12.13",
16     "socket.io": "^4.1.2"
17   },
18   "devDependencies": {
19     "nodemon": "^2.0.7",
20     "mongoose": "^5.12.13",
21     "socket.io": "^4.1.2"
22   }
23 }
24
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS D:\Laragon\laragon\www\google-docs-clone> cd client
PS D:\Laragon\laragon\www\google-docs-clone\client> npm run start
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Compiled successfully!

You can now view google-docs-clone in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.4.0.254:3000

Note that the development build is not optimized.
To create a production build, use npm run build.
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS D:\Laragon\laragon\www\google-docs-clone> cd server
PS D:\Laragon\laragon\www\google-docs-clone\server> npm run devStart
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

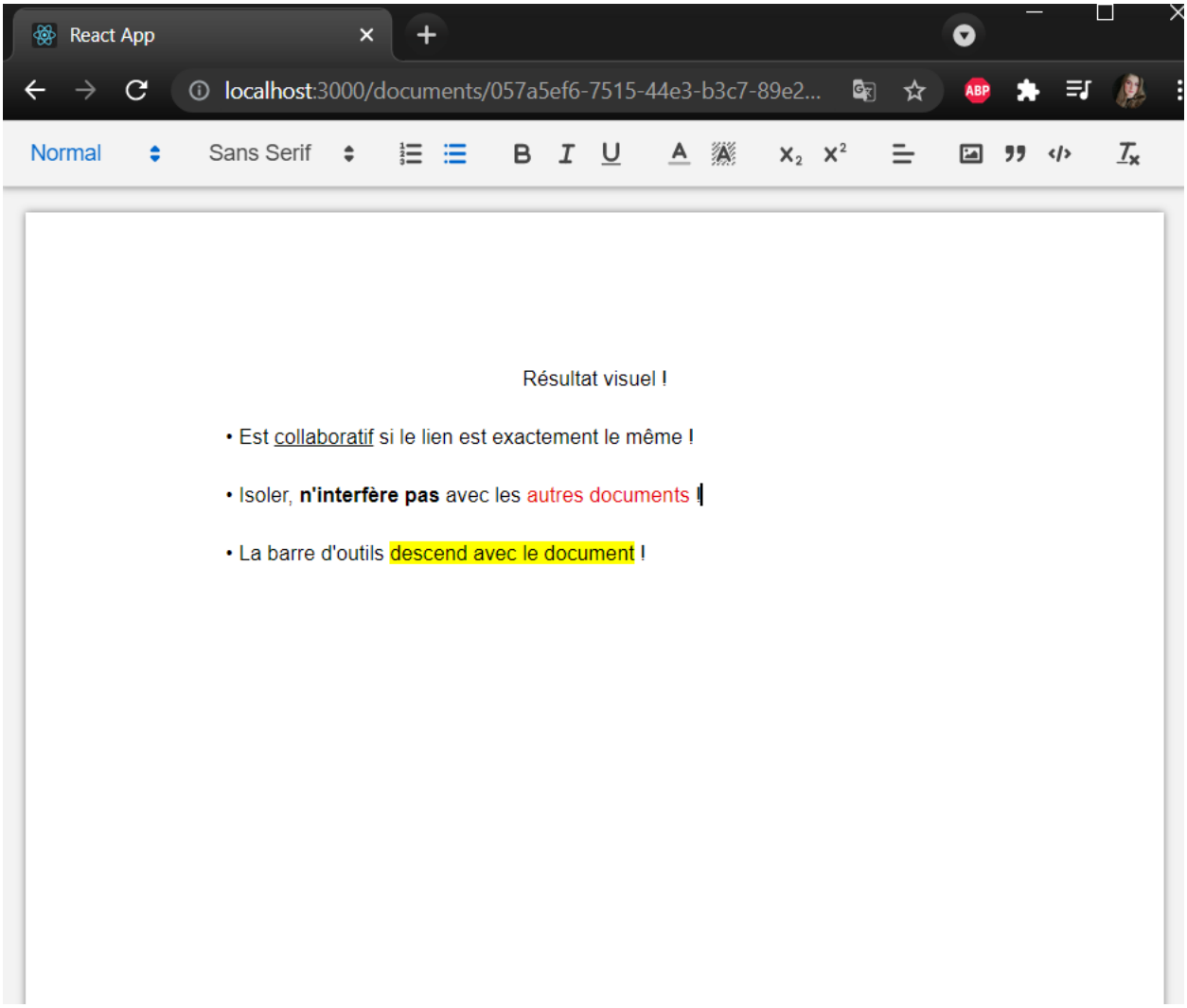
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS D:\Laragon\laragon\www\google-docs-clone> cd server
PS D:\Laragon\laragon\www\google-docs-clone\server> npm run devStart

> server@1.0.0 devStart D:\Laragon\laragon\www\google-docs-clone\server
> nodemon server.js

[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
```


React App
localhost:3000/documents/057a5ef6-7515-44e3-b3c7-89e2...
Normal
Sans Serif
Résultat visuel !
• Est collaboratif si le lien est exactement le même !
• Isoler, n'interfère pas avec les autres documents !
• La barre d'outils descend avec le document !

MARDI 15 & MERCREDI 16 JUIN 2021

MISSIONS :

1. Chercher comment faire un **Add-In Word** pour pouvoir le lier par la suite à une api Laravel. Aboutir à la création de cette Add-In.

REALISATIONS :

1.
   a. Recherche par rapport à la création d'un Add-In Word

Au début il y'a eu une confusion avec notre maître de stage car il avait employé le terme d'Add-On. Mais, après plusieurs recherches, on a pu trouver la documentation Microsoft et son tutoriel pour créer un Add-In. J'ai aussi trouvé le même tutoriel sur GitHub qui personnellement était plus simple à suivre pour moi.

Liens parcourus :

https://docs.microsoft.com/fr-fr/javascript/api/overview?view=word-js-preview
https://docs.microsoft.com/fr-fr/javascript/api/word?view=word-js-preview
https://docs.microsoft.com/fr-fr/javascript/api/word/word.application?view=word-js-preview
https://docs.microsoft.com/en-us/javascript/api/office?view=excel-js-preview (common)
https://docs.microsoft.com/en-us/office/dev/add-ins/develop/understanding-the-javascript-api-for-office
https://www.youtube.com/watch?v=D8zIHNtMSmE
https://www.youtube.com/watch?v=T3Px88x_PsA

Liens retenus :

https://github.com/OfficeDev/office-js-docs-pr/blob/master/docs/tutorials/word-tutorial.md
https://docs.microsoft.com/en-us/office/dev/add-ins/tutorials/word-tutorial

   b. Tutoriel et application des recherches :

J'ai commencé par installer **Yeoman Generator** grâce à **NPM** afin de pouvoir générer le projet de l'Add-In Word selon les critères de Microsoft Office.

```
C:\Users\vscha>npm install -g yo generator-office
```

```
C:\Users\vscha>yo office
```

```
     _-----_     ╭──────────────────────╮
    |       |    │  Welcome to the Office │
    |--(o)--|    │  Add-in generator, by  │
   `---------´   │ @OfficeDev! Let's create │
    ( _´U`_ )    │   a project together!  │
    /___A___\   /╰──────────────────────╯
     |  ~  |
   __'.___.'__
 ´   `  |° ´ Y `

? Office Add-in CLI tools collect anonymized usage data which is sent to Microsoft to help improve our product. Please r
ead our privacy notice at https://aka.ms/OfficeAddInCLIPrivacy.  To disable data collection, choose Exit and run "npx of
fice-addin-usage-data off".

 Continue
? Choose a project type: Office Add-in Task Pane project
? Choose a script type: JavaScript
? What do you want to name your add-in? My Office Add-in
? Which Office client application would you like to support? Word
```

Puis j'ai suivi étape par étape le tutoriel GitHub et j'ai abouti à ceci :

EXPLORER

MY OFFICE ADD-IN
- .vscode
- assets
- node_modules
- src
  - commands
  - taskpane
    - taskpane.css
    - taskpane.html
    - taskpane.js
  - .eslintrc.json
  - babel.config.json
  - base64Image.js
  - manifest.xml
  - package-lock.json
  - package.json
  - tsconfig.json
  - webpack.config.js

taskpane.html | taskpane.js | taskpane.css | package.json

src > taskpane > taskpane.html > html > body.ms-font-m.ms-welcome.ms-Fabric > main#app-body.ms-welcome__main > br

```html
<!-- Copyright (c) Microsoft Corporation. All rights reserved. Licensed under the MIT License. -->
<!-- This file shows how to design a first-run page that provides a welcome screen to the user about the features of the add-in. -->

<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=Edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Contoso Task Pane Add-in</title>

    <!-- Office JavaScript API -->
    <script type="text/javascript" src="https://appsforoffice.microsoft.com/lib/1.1/hosted/office.js"></script>

    <!-- For more information on Fluent UI, visit https://developer.microsoft.com/fluentui#/. -->
    <link rel="stylesheet" href="https://static2.sharepointonline.com/files/fabric/office-ui-fabric-core/9.6.1/css/fabric.min.css"/>

    <!-- Template styles -->
    <link href="taskpane.css" rel="stylesheet" type="text/css" />
</head>

<body class="ms-font-m ms-welcome ms-Fabric">
    <header class="ms-welcome__header ms-bgColor-neutralLighter">
        <img width="90" height="90" src="../../assets/logo-filled.png" alt="Contoso" title="Contoso" />
        <h1 class="ms-font-su">Welcome</h1>
    </header>
    <section id="sideload-msg" class="ms-welcome__main">
        <h2 class="ms-font-xl">Please sideload your add-in to see app body.</h2>
    </section>
    <main id="app-body" class="ms-welcome__main" style="display: none;">
        <button class="ms-Button" id="insert-paragraph">Insert Paragraph</button><br/><br/>
        <button class="ms-Button" id="apply-style">Apply Style</button><br/><br/>
        <button class="ms-Button" id="apply-custom-style">Apply Custom Style</button><br/><br/>
        <button class="ms-Button" id="change-font">Change Font</button><br/><br/>
        <button class="ms-Button" id="insert-text-into-range">Insert Abbreviation</button><br/><br/>
        <button class="ms-Button" id="insert-text-outside-range">Add Version Info</button><br/><br/>
        <button class="ms-Button" id="replace-text">Change Quantity Term</button><br/><br/>
        <button class="ms-Button" id="insert-image">Insert Image</button><br/><br/>
        <button class="ms-Button" id="insert-html">Insert HTML</button><br/><br/>
        <button class="ms-Button" id="insert-table">Insert Table</button><br/><br/>
        <button class="ms-Button" id="create-content-control">Create Content Control</button><br/><br/>
        <button class="ms-Button" id="replace-content-in-control">Rename Service</button><br/><br/>
    </main>
</body>

</html>
```

EXPLORER

MY OFFICE ADD-IN
- .vscode
- assets
- node_modules
- src
  - commands
  - taskpane
    - taskpane.css
    - taskpane.html
    - taskpane.js
  - .eslintrc.json
  - babel.config.json
  - base64Image.js
  - manifest.xml
  - package-lock.json
  - package.json
  - tsconfig.json
  - webpack.config.js

taskpane.html | taskpane.js | taskpane.css | package.json

src > taskpane > taskpane.js > replaceContentInControl

```javascript
/*
 * Copyright (c) Microsoft Corporation. All rights reserved. Licensed under the MIT license.
 * See LICENSE in the project root for license information.
 */

// images references in the manifest
import "../../assets/icon-16.png";
import "../../assets/icon-32.png";
import "../../assets/icon-80.png";
import { base64Image } from "../../base64Image";


/* global document, Office, Word */

Office.onReady((info) => {
    if (info.host === Office.HostType.Word) {
    if (!Office.context.requirements.isSetSupported('WordApi', '1.3')) {
      console.log('Sorry. The tutorial add-in uses Word.js APIs that are not available in your version of Office.');
    }//determine si notre version de word supporte word.js
    document.getElementById("insert-paragraph").onclick = insertParagraph;
    //event handlers et autre pour initialisation logique
    document.getElementById("apply-style").onclick = applyStyle;
    document.getElementById("sideload-msg").style.display = "none";
    document.getElementById("app-body").style.display = "flex";
    document.getElementById("apply-custom-style").onclick = applyCustomStyle;
    document.getElementById("change-font").onclick = changeFont;
    document.getElementById("insert-text-into-range").onclick = insertTextIntoRange;
    document.getElementById("insert-text-outside-range").onclick = insertTextBeforeRange;
    document.getElementById("replace-text").onclick = replaceText;
    document.getElementById("insert-image").onclick = insertImage;
    document.getElementById("insert-html").onclick = insertHTML;
    document.getElementById("insert-table").onclick = insertTable;
    document.getElementById("create-content-control").onclick = createContentControl;
    document.getElementById("replace-content-in-control").onclick = replaceContentInControl;

  }
});
```

EXPLORER

MY OFFICE ADD-IN
- .vscode
- assets
- node_modules
- src
  - commands
  - taskpane
    - taskpane.css
    - taskpane.html
    - taskpane.js
  - .eslintrc.json
  - babel.config.json
  - base64Image.js
  - manifest.xml
  - package-lock.json
  - package.json
  - tsconfig.json
  - webpack.config.js

taskpane.html | taskpane.js | taskpane.css | package.json

src > taskpane > taskpane.js > replaceContentInControl

```javascript

function insertParagraph() {
  Word.run(function (context) {

      var docBody = context.document.body;
      docBody.insertParagraph(
        "Office has many versions, including Office 2016, Office 2019, Office 365 Click-to-Run (C2R), Office on the web.",
        "Start"
      );

      return context.sync();
  })
  .catch(function (error) {
      console.log("Error: " + error);
      if (error instanceof OfficeExtension.Error) {
          console.log("Debug info: " + JSON.stringify(error.debugInfo));
      }
  });
}

function applyStyle() {
  Word.run(function (context) {

      var firstParagraph = context.document.body.paragraphs.getFirst();
      firstParagraph.styleBuiltIn = Word.Style.intenseReference;

      return context.sync();
  })
  .catch(function (error) {
      console.log("Error: " + error);
      if (error instanceof OfficeExtension.Error) {
          console.log("Debug info: " + JSON.stringify(error.debugInfo));
      }
  });
}

```

EXPLORER

MY OFFICE ADD-IN
- .vscode
- assets
- node_modules
- src
  - commands
  - taskpane
    - taskpane.css
    - taskpane.html
    - taskpane.js
  - .eslintrc.json
  - babel.config.json
  - base64Image.js
  - manifest.xml
  - package-lock.json
  - package.json
  - tsconfig.json
  - webpack.config.js

taskpane.html | taskpane.js | taskpane.css | package.json

src > taskpane > taskpane.js > replaceContentInControl

```javascript
function applyCustomStyle() {
  Word.run(function (context) {

      var lastParagraph = context.document.body.paragraphs.getLast();
      lastParagraph.style = "MyCustomStyle";

      return context.sync();
  })
  .catch(function (error) {
      console.log("Error: " + error);
      if (error instanceof OfficeExtension.Error) {
          console.log("Debug info: " + JSON.stringify(error.debugInfo));
      }
  });
}

function changeFont() {
  Word.run(function (context) {

      var secondParagraph = context.document.body.paragraphs.getFirst().getNext();
      secondParagraph.font.set({
        name: "Courier New",
        bold: true,
        size: 18
      });

      return context.sync();
  })
  .catch(function (error) {
      console.log("Error: " + error);
      if (error instanceof OfficeExtension.Error) {
          console.log("Debug info: " + JSON.stringify(error.debugInfo));
      }
  });
}
```

EXPLORER

MY OFFICE ADD-IN
- .vscode
- assets
- node_modules
- src
  - commands
  - taskpane
    - taskpane.css
    - taskpane.html
    - taskpane.js
  - .eslintrc.json
  - babel.config.json
  - base64Image.js
  - manifest.xml
  - package-lock.json
  - package.json
  - tsconfig.json
  - webpack.config.js

taskpane.html | taskpane.js | taskpane.css | package.json

src > taskpane > taskpane.js > applyCustomStyle

```
109
110   function insertTextIntoRange() {
111     Word.run(function (context) {
112
113         var doc = context.document;
114         var originalRange = doc.getSelection();
115         originalRange.insertText(" (C2R)", "End");
116
117         originalRange.load("text");
118         return context.sync()
119           .then(function() {
120               doc.body.insertParagraph("Current text of original range: " + originalRange.text, "End");
121           })
122           .then(context.sync);
123     })
124     .catch(function (error) {
125         console.log("Error: " + error);
126         if (error instanceof OfficeExtension.Error) {
127             console.log("Debug info: " + JSON.stringify(error.debugInfo));
128         }
129     });
130   }
131
132   function insertTextBeforeRange() {
133     Word.run(function (context) {
134
135         var doc = context.document;
136         var originalRange = doc.getSelection();
137         originalRange.insertText("Office 2019, ", "Before");
138         originalRange.load("text");
139         return context.sync()
140           .then(function() {
141             doc.body.insertParagraph("Current text of original range: " + originalRange.text, "End");
142           })
143           .then(context.sync);
144     })
145     .catch(function (error) {
146         console.log("Error: " + error);
147         if (error instanceof OfficeExtension.Error) {
148             console.log("Debug info: " + JSON.stringify(error.debugInfo));
149         }
150     });
151   }
```

EXPLORER

MY OFFICE ADD-IN
- .vscode
- assets
- node_modules
- src
  - commands
  - taskpane
    - taskpane.css
    - taskpane.html
    - taskpane.js
  - .eslintrc.json
  - babel.config.json
  - base64Image.js
  - manifest.xml
  - package-lock.json
  - package.json
  - tsconfig.json
  - webpack.config.js

taskpane.html | taskpane.js | taskpane.css | package.json

src > taskpane > taskpane.js > applyCustomStyle

```
152
153   function replaceText() {
154     Word.run(function (context) {
155
156         var doc = context.document;
157         var originalRange = doc.getSelection();
158         originalRange.insertText("many", "Replace");
159
160         return context.sync();
161     })
162     .catch(function (error) {
163         console.log("Error: " + error);
164         if (error instanceof OfficeExtension.Error) {
165             console.log("Debug info: " + JSON.stringify(error.debugInfo));
166         }
167     });
168   }
169
170   function insertImage() {
171     Word.run(function (context) {
172
173         context.document.body.insertInlinePictureFromBase64(base64Image, "End");
174
175         return context.sync();
176     })
177     .catch(function (error) {
178         console.log("Error: " + error);
179         if (error instanceof OfficeExtension.Error) {
180             console.log("Debug info: " + JSON.stringify(error.debugInfo));
181         }
182     });
183   }
184
185   function insertHTML() {
186     Word.run(function (context) {
187
188         var blankParagraph = context.document.body.paragraphs.getLast().insertParagraph("", "After");
189         blankParagraph.insertHtml('<p style="font-family: verdana;">Inserted HTML.</p><p>Another paragraph</p>', "End");
190
191         return context.sync();
192     })
193     .catch(function (error) {
194         console.log("Error: " + error);
195         if (error instanceof OfficeExtension.Error) {
196             console.log("Debug info: " + JSON.stringify(error.debugInfo));
197         }
198     });
199   }
```

EXPLORER

MY OFFICE ADD-IN
- .vscode
- assets
- node_modules
- src
  - commands
  - taskpane
    - taskpane.css
    - taskpane.html
    - taskpane.js
  - .eslintrc.json
  - babel.config.json
  - base64Image.js
  - manifest.xml
  - package-lock.json
  - package.json
  - tsconfig.json
  - webpack.config.js

taskpane.html | taskpane.js | taskpane.css | package.json

src > taskpane > taskpane.js > insertImage > Word.run() callback

```
201  function insertTable() {
202    Word.run(function (context) {
203
204        var secondParagraph = context.document.body.paragraphs.getFirst().getNext();
205
206        var tableData = [
207          ["Name", "ID", "Birth City"],
208          ["Bob", "434", "Chicago"],
209          ["Sue", "719", "Havana"],
210        ];
211  secondParagraph.insertTable(3, 3, "After", tableData);
212
213
214        return context.sync();
215    })
216    .catch(function (error) {
217        console.log("Error: " + error);
218        if (error instanceof OfficeExtension.Error) {
219            console.log("Debug info: " + JSON.stringify(error.debugInfo));
220        }
221    });
222  }
223
224  function createContentControl() {
225    Word.run(function (context) {
226
227        var serviceNameRange = context.document.getSelection();
228        var serviceNameContentControl = serviceNameRange.insertContentControl();
229        serviceNameContentControl.title = "Service Name";
230        serviceNameContentControl.tag = "serviceName";
231        serviceNameContentControl.appearance = "Tags";
232        serviceNameContentControl.color = "blue";
233
234        return context.sync();
235    })
236    .catch(function (error) {
237        console.log("Error: " + error);
238        if (error instanceof OfficeExtension.Error) {
239            console.log("Debug info: " + JSON.stringify(error.debugInfo));
240        }
241    });
242  }
243
```

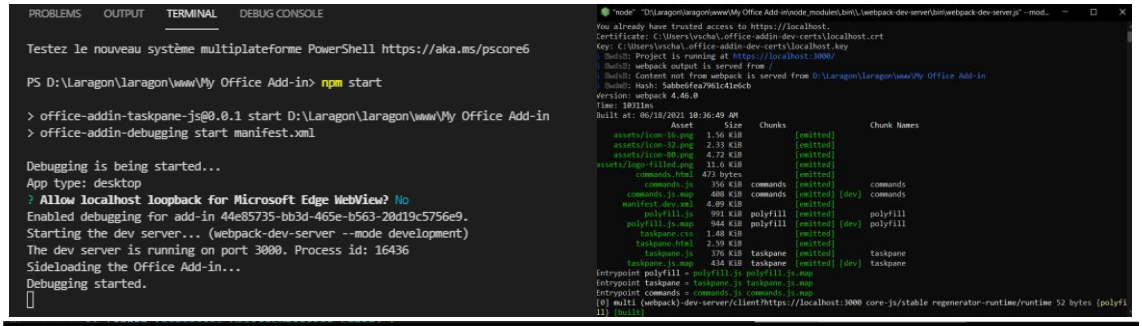
EXPLORER

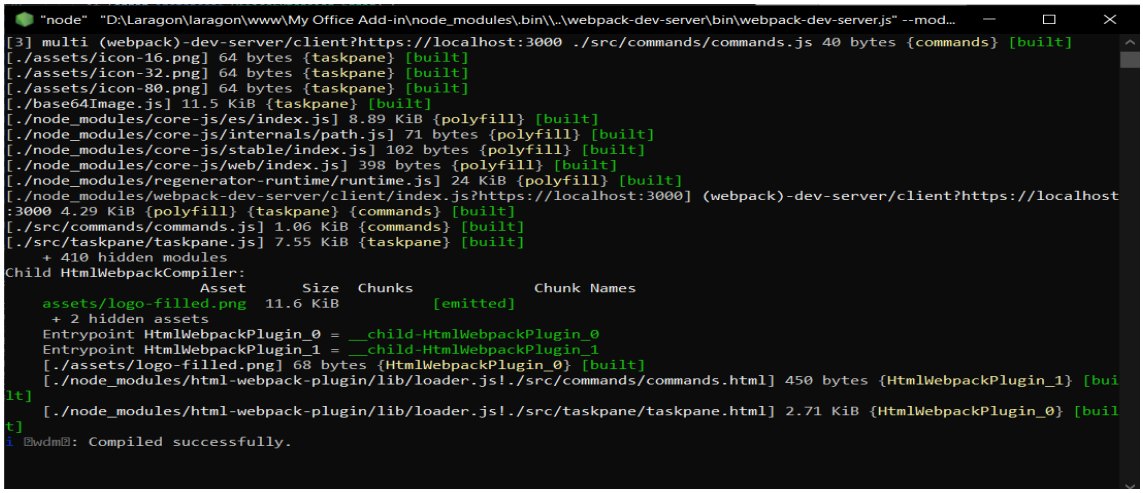
MY OFFICE ADD-IN
- .vscode
- assets
- node_modules
- src
  - commands
  - taskpane
    - taskpane.css
    - taskpane.html
    - taskpane.js
  - .eslintrc.json
  - babel.config.json
  - base64Image.js
  - manifest.xml
  - package-lock.json
  - package.json
  - tsconfig.json
  - webpack.config.js

taskpane.html | taskpane.js | taskpane.css | package.json

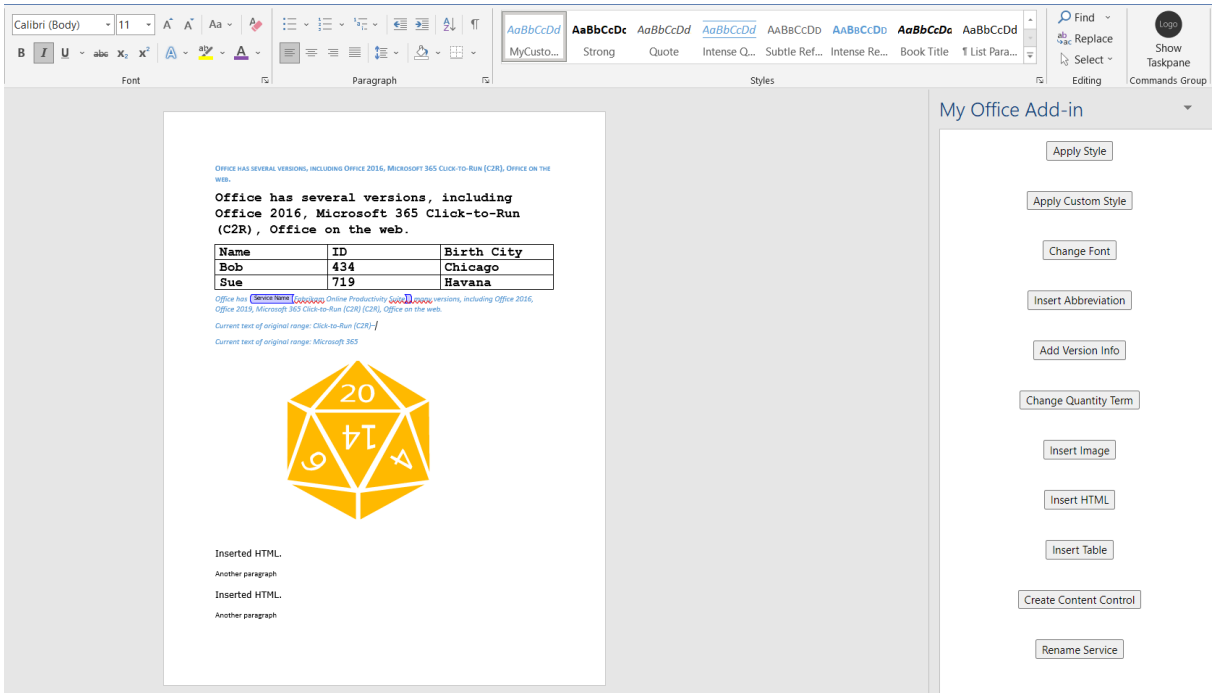
src > taskpane > taskpane.js > createContentControl > Word.run() callback

```
243
244  function replaceContentInControl() {
245    Word.run(function (context) {
246
247        var serviceNameContentControl = context.document.contentControls.getByTag("serviceName").getFirst();
248        serviceNameContentControl.insertText("Fabrikam Online Productivity Suite", "Replace");
249
250        return context.sync();
251    })
252    .catch(function (error) {
253        console.log("Error: " + error);
254        if (error instanceof OfficeExtension.Error) {
255            console.log("Debug info: " + JSON.stringify(error.debugInfo));
256        }
257    });
258  }
259
```





Résultat visuel dans Word :



JEUDI 17 JUIN 2021

MISSIONS :

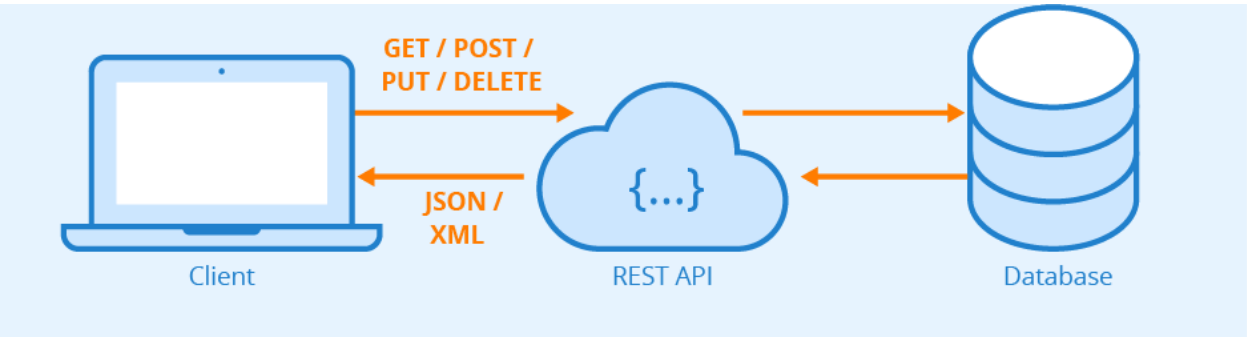
1. Chercher comment relier cet Add-In à notre projet.
2. Faire une **API** qui permet de mettre dans un document Word des paragraphes enregistrés en base de données.

REALISATIONS :

1. Recherches :

D'après mes recherches, la solution pour relier cet Add-In à notre projet serait de passer par une **API**. Une **API** permet de faire le lien entre une partie **Frontend (Client)** et la partie **Backend (Server).** La partie client peut choisir différents « Endpoint » proposés par l'**API** qui engendreront des actions et un retour de la partie serveur vers l'**API** qui redistribue l'information à la partie client.

Plus précisément on utilisera une **RESTful API** (**Re**presentational **S**tate **T**ransfer), cela représente en fait une forme d'architecture utilisée pour communiquer entre le client et le serveur. Ce type d'**API** utilise l'**HTTP** et les données qu'elle transmet sont en **JSON** (ou **XML** mais cela ne nous intéresse pas dans notre cas).



Source : seobility.net

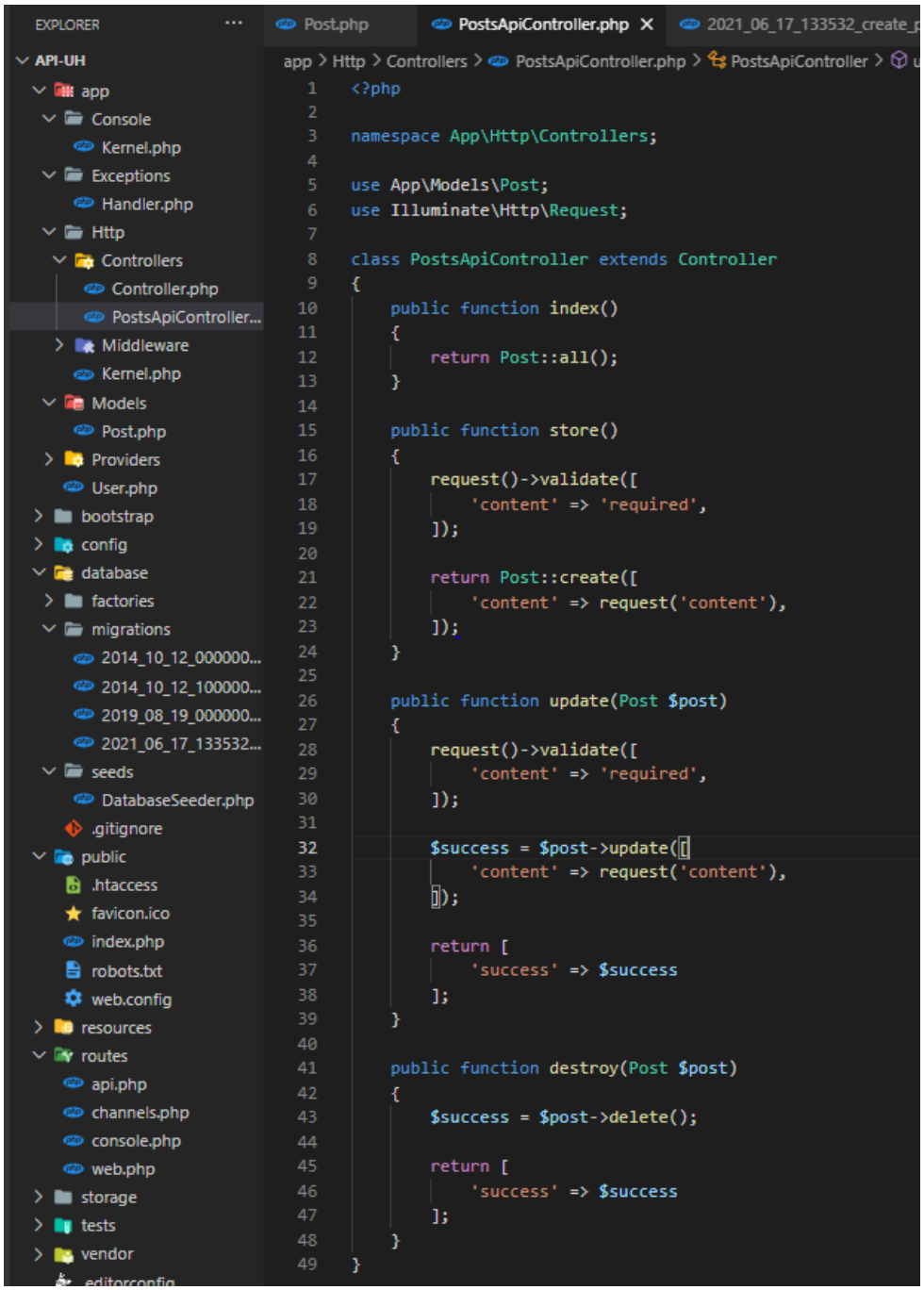
Liens retenus lors des recherches :

https://www.youtube.com/watch?v=xvqPEEpRBJ4
https://www.youtube.com/watch?v=WDha52dbLWM
https://laravel.com/docs/8.x/routing#view-routes
https://laravel.com/docs/8.x/container#container-events
https://laravel.sillo.org/cours-laravel-8-les-donnees-les-ressources-api/
https://www.seobility.net/en/wiki/REST_API

2. Application :

Après avoir effectué la première mission, j'ai demandé à mon maitre de stage la prochaine étape : « *Pour le moment j'aimerais que vous fassiez une API qui permet de mettre dans un document Word des paragraphes enregistrés en base de données.* »

J'ai donc commencé par créer un nouveau projet avec la structure qui me semblait adéquate (similaire à la première semaine de stage) :



```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Post extends Model
{
    use HasFactory;

    protected $guarded = [];
}
```

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreatePostsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('posts', function (Blueprint $table) {
            $table->id();
            $table->text('content');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('posts');
    }
}
```

Après avoir fait la structure et la base de données avec des paragraphes pré-enregistrés (**'content'**), j'ai cherché comment faire le lien entre cette API et l'Add-In que nous avons créé. Je sais maintenant qu'il faut passer par le fichier **'routes/api.php'**, mais je n'ai pas encore saisi comment réaliser ces routes correctement. Je continue donc mes recherches.

VENDREDI 18 JUIN 2021

MISSIONS :

1. Continuer sur les activités en cours.

REALISATIONS :

1.

En Cours …