

# Note de stage 2022

## ⌘UH Group⌘

### Semaine n° 1

## ⌘Objectifs

---

**Contexte** : UH Lawyers<sup>1</sup> est une application créée pour les avocats, un Add-In Word vient la compléter en proposant l'insertion facile de paragraphes pré-écrits dans un document Word ainsi que des « champs de fusion » qui permettront à l'avocat de positionner des sortes de « placeholder » qui seront automatiquement remplis par l'application UH Lawyers grâce aux fonctionnalités de PHPWord.

1. Reprendre le projet sur l'Add-In Word.
2. Se renseigner sur PHPWord<sup>2</sup> et tester<sup>3</sup>.
3. Réaliser les fonctionnalités de l'Add-in avec des données provenant d'une API.
4. Se renseigner sur comment convertir un fichier Word en PDF avec PHP.

---

<sup>1</sup> [Showcase LinkedIn de UH Lawyers](#)

<sup>2</sup> [Documentation générale sur PHPWord](#)

<sup>3</sup> [Documentation sur l'installation de PHPWord](#)

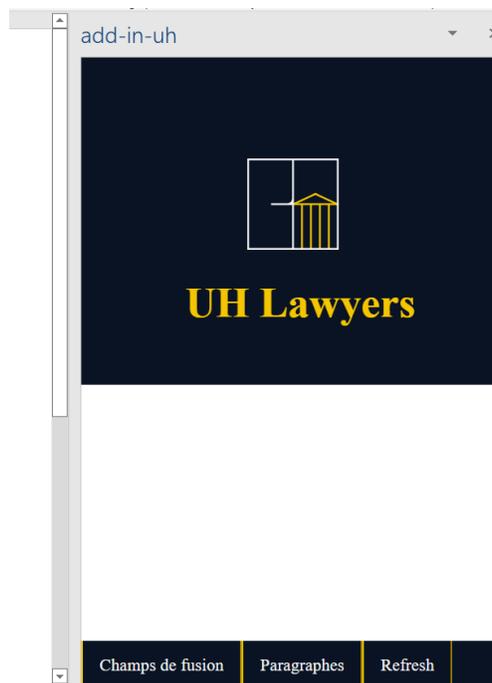
## Réalisations

---

Pour cette reprise de projet, j'ai commencé par me réapproprier le projet que j'avais fait précédemment. J'ai donc de nouveau consulté le tutoriel de Microsoft<sup>4</sup>, les fichiers du projet et mes rapports de l'année précédente. Pour repartir sur de bonnes bases j'ai décidé de faire un nouveau projet composé de ce que j'avais déjà (de plus les anciens projets possèdent des erreurs à cause de mes différents tests). Avec l'expérience de l'année dernière j'ai rencontré beaucoup moins de difficulté et d'erreurs lors de la mise en place du projet, et lorsque l'Add-In plantait je savais pourquoi et cela m'a permis de me corriger plus facilement.

J'ai aussi dû faire quelques révisions sur le JavaScript : 'while', 'for', 'foreach', 'if', 'array', 'innerHTML', 'onclick'.

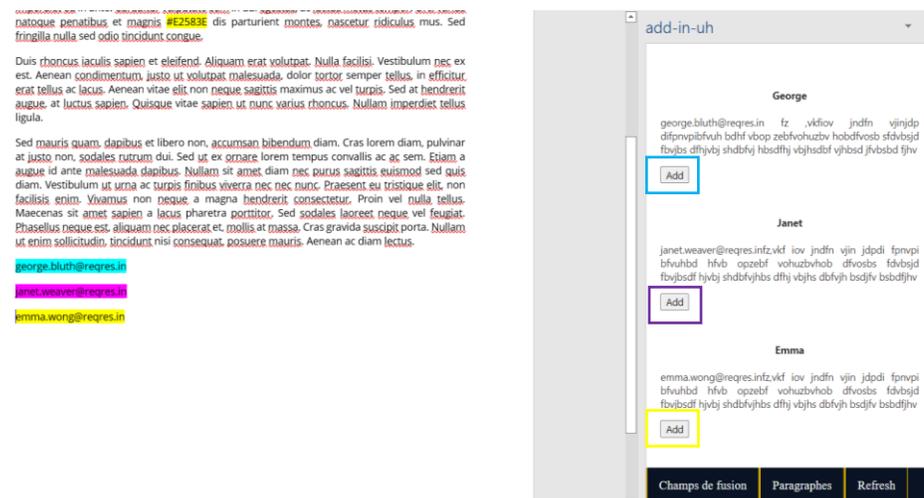
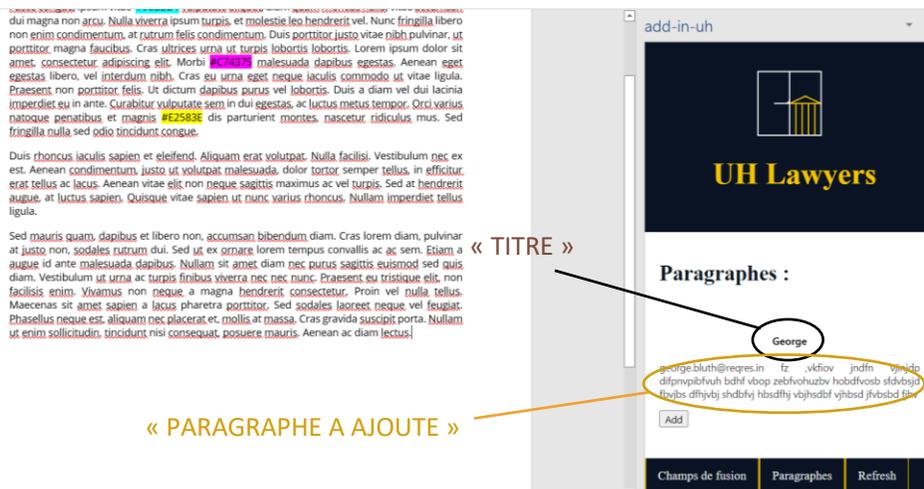
Ainsi sur ce nouveau projet j'ai commencé par faire la mise en forme de l'application. Faire en sorte que le bandeau soit personnalisé pour l'application UH Lawyers ainsi qu'une barre de navigation permettant de faire apparaître les différents onglets de fonctionnalité :



---

<sup>4</sup> [Tutoriel Microsoft Add-In Word](#)

Ensuite j'ai implémenté les fonctionnalités que m'a demandé mon maître de stage. J'ai commencé par la fonctionnalité « Paragraphes », j'ai eu pas mal de difficultés car avec les différentes sources que j'ai consultées me rendait assez confuse sur la syntaxe à utiliser. Notamment parce que j'étais partie sur la syntaxe avec les « Littéraux de gabarits »<sup>5</sup> pour l'interprétation des données récupérées par Axios<sup>6</sup> alors que cela n'était pas nécessaire et restrictif.



Lorsque l'on clique sur le bouton « Add » le paragraphe concerné s'ajoute automatiquement à la fin du document (si besoin ce paramètre peut être changé dans le futur).

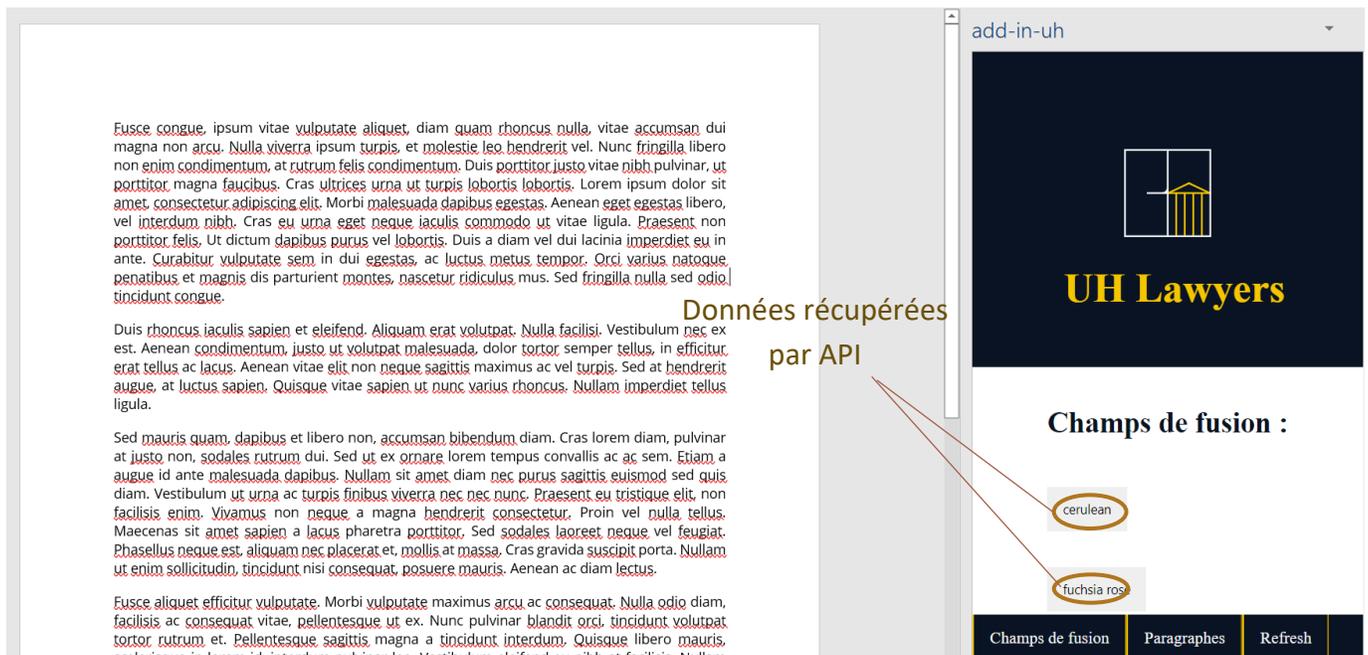
Le paragraphe sur la droite ne correspond pas au paragraphe sur le document car c'est seulement les données de l'API que j'ajoute au document, le texte que j'ai rajouté était un moyen pour moi d'améliorer le visuel de l'application.

<sup>5</sup> [Documentation MDN Littéraux de gabarits](#)

<sup>6</sup> [GitHub Axios](#) & [Documentation Axios](#)

A l'origine j'avais créé une API de test avec des paragraphes en base de données, mais malheureusement en voulant l'essayer je me suis rappelé que l'Add-In Word n'accepte pas les APIs avec un protocole HTTP, seulement celles avec un protocole HTTPS. Après quelques recherches il semble possible de changer le protocole de mon API en HTTPS grâce à l'outil « mckert »<sup>7</sup>. J'ai cependant préféré utiliser une API de reqres.in que j'avais déjà utilisé pour pouvoir gagner du temps et progresser sur la partie fonctionnelle de l'Add-In (Je n'ai malheureusement pas réussi à trouver une API en HTTPS contenant plusieurs données qui collait mieux au contexte).

De la même manière, je me suis occupé de la fonctionnalité « Champs de fusion ». Au début ce que je devais faire dans cette fonctionnalité était très flou pour moi, mon maître de stage utilisait les termes de « fusion », « champs de fusion », « qui permettra de fusionner le document une fois que l'utilisateur l'importera dans l'application » ... Ce qui m'a mené à différentes pistes avec l'existence de champs de fusion natifs à Word et différentes fonctions de fusion (« merge ») proposées par PHPWord. Finalement après clarification et quelques recherches il fallait utiliser les « Templates<sup>8</sup> » de PHPWord qui permettent d'ajouter des sortes de « Placeholder » au document qui seront remplis automatiquement par l'application UH Lawyers (et ici de l'application de test que j'ai créée).



<sup>7</sup> [GitHub mckert](#)

<sup>8</sup> [Documentation Templates PHPWord](#)

En condition réelle on aurait `{cerulean}`, ici c'est parce que la donnée provient d'une API quelconque

Fusce congue, ipsum vitae `#98B2D1` vulputate aliquet, diam quam rhoncus nulla, vitae accumsan dui magna non arcu. Nulla viverra ipsum turpis, et molestie leo hendrerit vel. Nunc fringilla libero non enim condimentum, at rutrum felis condimentum. Duis porttitor justo vitae nibh pulvinar, ut porttitor magna faucibus. Cras ultrices urna ut turpis lobortis lobortis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi `#C74375` malesuada dapibus egestas. Aenean eget egestas libero, vel interdum nibh. Cras eu urna eget neque iaculis commodo ut vitae ligula. Praesent non porttitor felis. Ut dictum dapibus purus vel lobortis. Duis a diam vel dui lacinia imperdiet eu in ante. Curabitur vulputate sem in dui egestas, ac luctus metus tempor. Orci varius natoque penatibus et magnis `#E2583E` dis parturient montes, nascetur ridiculus mus. Sed fringilla nulla sed odio tincidunt congue.

Duis rhoncus iaculis sapien et eleifend. Aliquam erat volutpat. Nulla facilisi. Vestibulum nec est. Aenean condimentum, justo ut volutpat malesuada, dolor tortor semper tellus, in efficitur erat tellus ac lacus. Aenean vitae elit non neque sagittis maximus ac vel turpis. Sed at hendrerit augue, at luctus sapien. Quisque vitae sapien ut nunc varius rhoncus. Nullam imperdiet tellus ligula.

Sed mauris quam, dapibus et libero non, accumsan bibendum diam. Cras lorem diam, pulvinar at justo non, sodales rutrum dui. Sed ut ex ornare lorem tempus convallis ac ac sem. Etiam a augue id ante malesuada dapibus. Nullam sit amet diam nec purus sagittis euismod sed quis diam. Vestibulum ut urna ac turpis finibus viverra nec nec nunc. Praesent eu tristique elit, non facilisis enim. Vivamus non neque a magna hendrerit consectetur, Proin vel nulla tellus. Maecenas sit amet sapien a lacus pharetra porttitor. Sed sodales laoreet neque vel feugiat. Phasellus neque est, aliquam nec placerat et, mollis at massa. Cras gravida suscipit porta. Nullam ut enim sollicitudin, tincidunt nisi consequat, posuere mauris. Aenean ac diam lectus.

Fusce aliquet efficitur vulputate. Morbi vulputate maximus arcu ac consequat. Nulla odio diam, facilisis ac consequat vitae, pellentesque ut ex. Nunc pulvinar blandit orci, tincidunt volutpat tortor rutrum et. Pellentesque sagittis magna a tincidunt interdum. Quisque libero mauris, scelerisque in lorem id, interdum pulvinar leo. Vestibulum eleifend eu nibh at facilisis. Nullam

add-in-uh

Champs de fusion :

cerulean

fuchsia rose

true red

aqua sky

tigerlily

blue turquoise

Champs de fusion Paragraphe Refresh

Quant au bouton « Refresh » il sert à rafraîchir les données en provenance de l'API.

Pour ces fonctionnalités j'ai travaillé avec l'API proposé par reqres.in, la « users »<sup>9</sup> pour les paragraphes et la « unknown »<sup>10</sup> pour les champs de fusion afin de varier un peu les données :

```
reqres.in/api/users
{"page":1,"per_page":6,"total":12,"total_pages":2,"data":
[{"id":1,"email":"george.bluth@reqres.in","first_name":"George","last_name":"Bluth","avatar":"https://reqres.in/img/faces/1-image.jpg"},
{"id":2,"email":"janet.weaver@reqres.in","first_name":"Janet","last_name":"Weaver","avatar":"https://reqres.in/img/faces/2-image.jpg"},
{"id":3,"email":"emma.wong@reqres.in","first_name":"Emma","last_name":"Wong","avatar":"https://reqres.in/img/faces/3-image.jpg"},
{"id":4,"email":"eve.holt@reqres.in","first_name":"Eve","last_name":"Holt","avatar":"https://reqres.in/img/faces/4-image.jpg"},
{"id":5,"email":"charles.morris@reqres.in","first_name":"Charles","last_name":"Morris","avatar":"https://reqres.in/img/faces/5-image.jpg"},
{"id":6,"email":"tracey.ramos@reqres.in","first_name":"Tracey","last_name":"Ramos","avatar":"https://reqres.in/img/faces/6-image.jpg"}],
"support":{"url":"https://reqres.in/#support-heading","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}}
```

```
reqres.in/api/unknown
{"page":1,"per_page":6,"total":12,"total_pages":2,"data":
[{"id":1,"name":"cerulean","year":2000,"color":"#98B2D1","pantone_value":"15-4020"},
{"id":2,"name":"fuchsia rose","year":2001,"color":"#C74375","pantone_value":"17-2031"},
{"id":3,"name":"true red","year":2002,"color":"#BF1932","pantone_value":"19-1664"},{"id":4,"name":"aqua sky","year":2003,"color":"#7BC4C4","pantone_value":"14-4811"},
{"id":5,"name":"tigerlily","year":2004,"color":"#E2583E","pantone_value":"17-1456"},{"id":6,"name":"blue turquoise","year":2005,"color":"#53B0AE","pantone_value":"15-5217"}],
"support":{"url":"https://reqres.in/#support-heading","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}}
```

<sup>9</sup> Lien API users

<sup>10</sup> Lien API Unknown



• JS

```

OPEN EDITORS
taskpane.ht... 3
manifest.xml
webpack.config.js
taskpane.css src/...
commands.html...
taskpane.js src/...
ADD-IN-...
.vscode
extensions.json
launch.json
settings.json
tasks.json
assets
icon-16.png
icon-32.png
icon-64.png
icon-80.png
icon-128.png
logo-filled.png
logo.svg
node_modules
src
commands
commands.html
commands.js
taskpane
taskpane.css
taskpane... 3
taskpane.js
.eslintrc.json
.gitignore
babel.config.json
manifest.xml
package-lock.json
package.json
tsconfig.json
webpack.config.js

src > taskpane > taskpane.js > showOrHideCF
You, 9 hours ago | 1 author (You)
1  /*
2  * Copyright (c) Microsoft Corporation. All rights reserved. Licensed under the MIT license.
3  * See LICENSE in the project root for license information.
4  */
5
6  // Data
7  var data;
8  var dataTest;
9  var dataChamps;
10
11  /* global document, Office, Word */
12
13  Office.onReady((info) => {
14    if (info.host === Office.HostType.Word) {
15      // Determine if the user's version of Office supports all the Office.js APIs that are used in the tutorial.
16      if (!Office.context.requirements.isSetSupported('WordApi', '1.3')) {
17        console.log('Sorry. The tutorial add-in uses Word.js APIs that are not available in your version of Office.');

```

```

OPEN EDITORS
taskpane.ht... 3
manifest.xml
webpack.config.js
taskpane.css src/...
commands.html...
taskpane.js src/...
ADD-IN-...
.vscode
extensions.json
launch.json
settings.json
tasks.json
assets
icon-16.png
icon-32.png
icon-64.png
icon-80.png
icon-128.png
logo-filled.png
logo.svg
node_modules
src
commands
commands.html
commands.js
taskpane
taskpane.css
taskpane... 3
taskpane.js
.eslintrc.json
.gitignore
babel.config.json
manifest.xml
package-lock.json
package.json
tsconfig.json
webpack.config.js

src > taskpane > taskpane.js > showOrHideCF
42  // Tutoriel & Tests
43
44  function insertParagraph() {
45    Word.run(function (context) {
46
47      var docBody = context.document.body;
48      docBody.insertParagraph("Office has several versions, including Office 2016, Microsoft 365 subscription, and Office on the web.", "Start");
49
50      return context.sync();
51    })
52    .catch(function (error) {
53      console.log("Error: " + error);
54      if (error instanceof OfficeExtension.Error) {
55        console.log("Debug info: " + JSON.stringify(error.debugInfo));
56      }
57    });
58  }
59
60  function insertParagraphAPI() {
61    Word.run(function (context) {
62      if(!data) {
63        fetchData();
64        var docBody = context.document.body;
65        docBody.insertParagraph(
66          `${data.data.data[0].email}`, "Start"
67        );
68      }
69      else {
70        var docBody = context.document.body;
71        docBody.insertParagraph(
72          `${data.data.data[0].email}`, "Start"
73        );
74      }
75
76      return context.sync();
77    })
78    .catch(function (error) {
79      console.log("Error: " + error);
80      if (error instanceof OfficeExtension.Error) {
81        console.log("Debug info: " + JSON.stringify(error.debugInfo));
82      }
83    });
84  }

```

```

src > taskpane > taskpane.js > showOrHideCF
85 function insertParagraphApiTest() {
86     Word.run(function (context) {
87         if(!dataTest) {
88             fetchDataApiTest();
89             var docBody = context.document.body;
90             docBody.insertParagraph(
91                 `${dataTest[0].title}`, "Start"
92             );
93         }
94         else {
95             var docBody = context.document.body;
96             docBody.insertParagraph(
97                 `${dataTest[0].title}`, "Start"
98             );
99         }
100     }
101     return context.sync();
102 }
103 .catch(function (error) {
104     console.log("Error: " + error);
105     if (error instanceof OfficeExtension.Error) {
106         console.log("Debug info: " + JSON.stringify(error.debugInfo));
107     }
108 });
109 }
110
111 function insertChamp() {
112     fetchDataApiTestChamps();
113     Word.run(function (context) {
114
115         var doc = context.document;
116         var originalRange = doc.getSelection();
117         originalRange.insertText( dataChamps[2].name , "End");
118
119         originalRange.load("text");
120         return context.sync();
121     })
122 .catch(function (error) {
123     console.log("Error: " + error);
124     if (error instanceof OfficeExtension.Error) {
125         console.log("Debug info: " + JSON.stringify(error.debugInfo));
126     }
127 });
128 }
129
130
131 // Fonction application finale
132

```

```

src > taskpane > taskpane.js > showOrHideCF
130
131 // Fonction application finale
132
133 function showOrHideCF() {
134     fetchDataApiTestChamps();
135     var contenu = document.getElementById('champs');
136
137     var name = "" + dataChamps[0].name + "";
138     contenu.innerHTML = '<button class="ms-Button" onclick="addChamp('+ name + ')">' + dataChamps[0].name + '</button><br><br>';
139     // contenu.innerHTML = '<button class="champ-button" onclick="insertChamp()">' + dataChamps[0].name + '</button>';
140     for (let i = 1; i < dataChamps.length; i++) {
141         name = "" + dataChamps[i].name + "";
142         contenu.innerHTML += '<button class="ms-Button" onclick="addChamp('+ name + ')">' + dataChamps[i].name + '</button><br><br>';
143         // contenu.innerHTML += '<button class="champ-button" onclick="insertChamp()">' + dataChamps[i].name + '</button>';
144     }
145
146     var champ = document.getElementById('champs-fusion');
147     var para = document.getElementById('paragraphes');
148     if(champ.style.display == "none") {
149         champ.style.display = "block";
150         para.style.display = "none";
151     }
152     else {
153         champ.style.display = "none";
154     }
155 }
156

```

```

OPEN EXTENSIONS  src > taskpane > taskpane.js > showOrHideP
taskpane.html... 157 function showOrHideP() {
manifest.xml     158   fetchDataApiTest();
webpack.config.js 159   var contenu = document.getElementById('paraphs');
taskpane.css src... 160
commands.html... 161   //var nb = parseInt(`${dataTest.data.data}`.length);
taskpane.js src... 162   //var nb = Object.keys(`${dataTest.data.data}`.length);
ADD-IN-UH       163   var title = dataTest[0].first_name;
.vscode         164   var description = dataTest[0].email + " fz ,vkfiv jndfn yjindf dlfmnpvlrfvbn bnf vob zefvnuvzv hodfvob stvbcvj fvbjv dnfjvj chdvjv hvdvjv vjhdvj vjhdvj vjhdvj fjhvj";
extensions.json  165   contenu.innerHTML += `<div class="text-ajout"><div class="description">+ description </div> <button id="${ dataTest[0].id + " class="add-button" onclick="addParagraph(0)">Add/button</div>`;
launch.json     166   for (let i = 1; i < dataTest.length; i++) {
tasks.json      167     var title = dataTest[i].first_name;
settings.json   168     var description = dataTest[i].email + " fz,vkf ivj jndfn yjin jndfj fmvjv hfvubd hfvjv optvbf vovvchvob dfvobv fvdvbjv fvdvbjv hvjvj shbvfvchvjv dfvjv vobvjv dfvjvj bsdfvv bsdfvvjv";
tasks.json      169     contenu.innerHTML += `<div class="text-ajout"><div class="description">+ description </div> <button id="${ dataTest[i].id + " class="add-button" onclick="addParagraph(+ 1 +)">Add/button</div>`;
assets          170   }
icon-16.png     171   }
icon-32.png     172   var champ = document.getElementById('champs-fusion');
icon-64.png     173   var para = document.getElementById('paraphes');
icon-80.png     174   if(para.style.display == "none") {
icon-128.png    175     para.style.display = "block";
logo-filled.png 176     champ.style.display = "none";
logo.png        177   }
node_modules    178   else {
src             179     para.style.display = "none";
commands        180   }
commands.html   181   }
commands.js     182   function addParagraph(ind){
taskpane        183   fetchDataApiTest();
taskpane        184   Word.run(function (context) {
taskpane        185     Word.run(function (context) {
taskpane        186       var docBody = context.document.body;
taskpane        187       docBody.InsertParagraph(dataTest[ind].email,"end");
taskpane        188     });
taskpane        189     return context.sync();
.eslintrc.json  190   });
.gitignore      191   });
babel.config.js 192   .catch(function (error) {
manifest.xml    193     console.log("Error: " + error);
package-lock.json 194     if (error instanceof OfficeExtension.Error) {
package.json    195       console.log("Debug Info: " + JSON.stringify(error.debugInfo));
tsconfig.json  196     });
webpack.config.js 197   });
webpack.config.js 198   });

```

```

OPEN E...  src > taskpane > taskpane.js > showOrHideP
taskpane.html... 3
manifest.xml     200
webpack.config.js 201
taskpane.css src... 202
commands.html... 203
taskpane.js src... 204
ADD-IN-UH       205
.vscode         206
extensions.json  207
launch.json     208
settings.json   209
tasks.json      210
assets          211
icon-16.png     212
icon-32.png     213
icon-64.png     214
icon-80.png     215
icon-128.png    216
logo-filled.png 217
logo.png        218
node_modules    219
src             220
commands        221
commands.html   222
commands.js     223
taskpane        224
taskpane        225
taskpane        226
taskpane        227
taskpane        228
taskpane        229
taskpane        230
taskpane        231
taskpane        232
.eslintrc.json  233
.gitignore      234
babel.config.js 235
manifest.xml    236
package-lock.json 237
package.json    238
tsconfig.json  239
webpack.config.js 240
webpack.config.js 241
webpack.config.js 242
webpack.config.js 243
webpack.config.js 244
webpack.config.js 245
webpack.config.js 246
webpack.config.js 247
webpack.config.js 248
webpack.config.js 249
webpack.config.js 250
webpack.config.js 251

200 function addChamp(nom) {
201   fetchDataApiTestChamps();
202   Word.run(function (context) {
203     let i = 0;
204     while( dataChamps[i].name != nom){
205       i++;
206     }
207     var doc = context.document;
208     var originalRange = doc.getSelection();
209     originalRange.insertText( " " + dataChamps[i].color , "End");
210     originalRange.load("text");
211
212     return context.sync();
213   })
214   .catch(function (error) {
215     console.log("Error: " + error);
216     if (error instanceof OfficeExtension.Error) {
217       console.log("Debug info: " + JSON.stringify(error.debugInfo));
218     }
219   });
220 }
221
222 function fetchData() {
223   axios.get('https://reqres.in/api/users')
224     .then(resp => {
225       // handle success
226       showOutput(resp);
227     })
228     .catch(error => {
229       // handle error
230       console.error(error);
231     })
232   }
233
234 function showOutput(resp) {
235   data = resp;
236   // data = `${resp.data.data[0].email}`;
237 }
238
239 function fetchDataApiTest() {
240   axios.get('https://reqres.in/api/users')
241     .then(resp => {
242       // handle success
243       dataTest = resp.data.data;
244     })
245     .catch(error => {
246       // handle error
247       console.error(error);
248     })
249   }
250 }
251 }

```

```

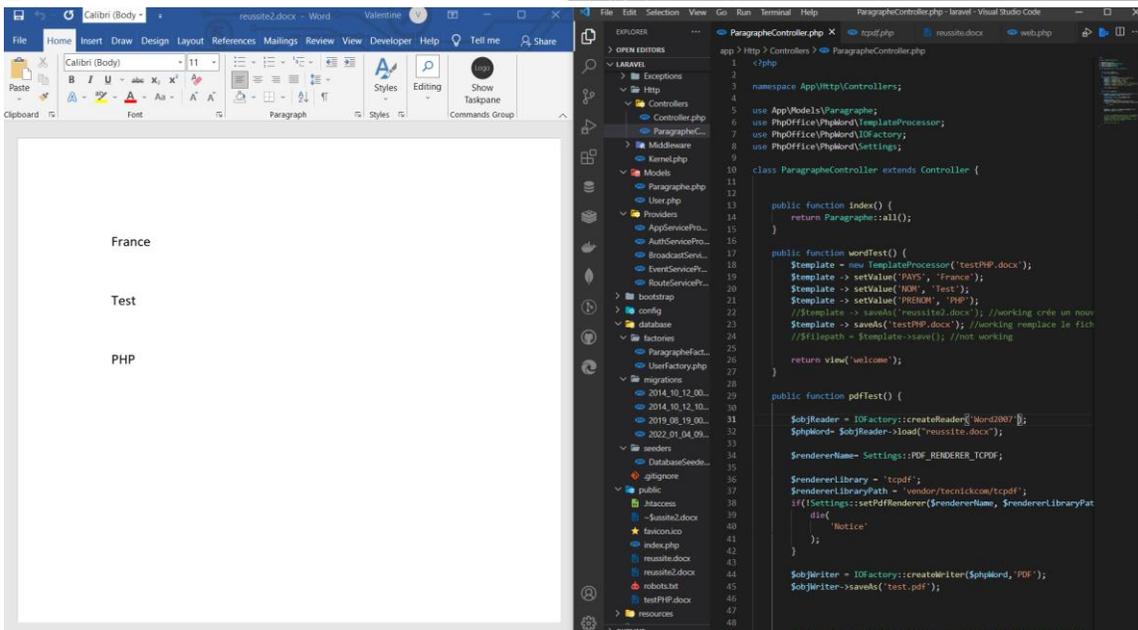
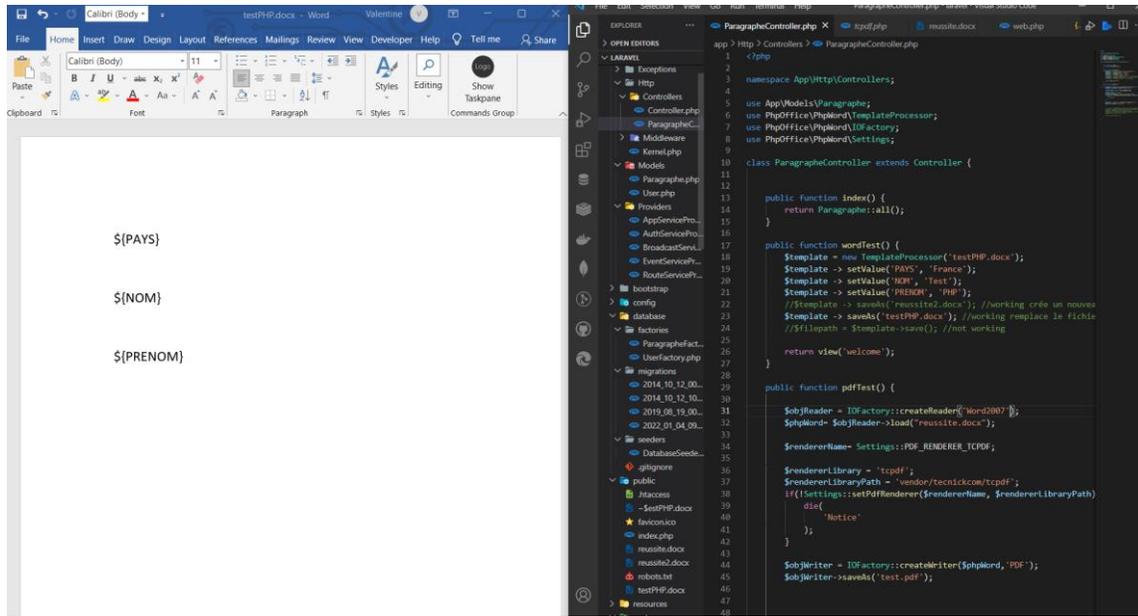
src > taskpane > taskpane.js > ...
252
253 function fetchDataApiTestChamps() {
254     axios.get('https://reqres.in/api/unknown')
255         .then(resp => {
256             // handle success
257             dataChamps = resp.data.data;
258         })
259         .catch(error => {
260             // handle error
261             console.error(error);
262         })
263     }
264 }
265
266 function refresh() {
267     fetchData();
268 }
    
```

- CSS

```

src > taskpane > taskpane.css > .add-button
107 h1 {
108     font-size: 54px;
109     padding-left: 3.76%;
110 }
111
112 h3 {
113     color : #091323;
114     font-size: 30px;
115     font-family: 'Sen';
116     font-weight: 700;
117     line-height: normal;
118 }
119 #app-bosy {
120     padding-bottom: 20px;
121 }
122 .app-navigation {
123     background-color: #091323;
124     position: fixed;
125     bottom: 0;
126     width: 100%;
127 }
128
129 #paragraphes {
130     width: 100%;
131     display: block;
132 }
133
134 .navbar {
135     width: 100%;
136     justify-content: space-between;
137     font-family: 'Sen';
138     background-color: #091323;
139     position: fixed;
140     bottom: 0;
141     width: 100%;
142 }
143
144 .nav-link {
145     position: relative;
146     margin-right: 32px;
147 }
148
149 .nav-links {
150     list-style-type: none;
151 }
152
153
154
155 .nav-item {
156     float: left;
157     display: block;
158     text-align : center;
159     background-color: #091323;
160     padding : 14px 16px;
161     border-color: #f4c500;
162     color: #f2f4f4;
163     text-decoration: none;
164     font-family: 'Sen';
165     font-size: 17px;
166     border-top-style: none;
167     border-bottom-style: none;
168 }
169
170 #paragraphs, #champs {
171     display: block;
172 }
173
174 .div-contenu {
175     float: center;
176     text-align : center;
177     padding-bottom: 100px;
178 }
179
180 .ms-Button {
181     font-family: 'Sen';
182     border-style: none;
183     padding : 14px 16px;
184     display: block;
185 }
186
187 .description {
188     text-align: justify;
189     width: 100%;
190 }
191
192 .add-button {
193     display: block;
194 }
195
196 h4 {
197     display : block;
198 }
199
200 .text-ajout {
201     display: block;
202     padding-bottom: 25px;
    }
    
```

En plus de l'Add-In j'ai donc effectué un test avec PHPWord pour les champs de fusion avec un projet Laravel de test :



```

Route::get('/', function () {
    return view('welcome');
});
Route::get('/test', [ParagrapheController::class, 'wordTest']);
Route::get('/testPDF', [ParagrapheController::class, 'pdfTest']);
    
```

Actuellement je suis donc sur l'objectif numéro 4, je suis en train de chercher comment convertir les documents Word créés par l'Add-In en PDF une fois importés dans l'application UH Lawyers afin de proposer les deux formats. Pour le moment, cela semble possible via PHPWord plus TCPDF<sup>11</sup> ou DOMPDF<sup>12</sup>, mais d'après les différents témoignages que j'ai lus cela ne préserverait pas la mise en forme ni les images. Je suis donc toujours sur mes recherches et je consulte les différentes solutions proposées, et je commence à faire des petits tests sur mon projet de test Laravel (peu concluant pour le moment).

Avec les difficultés que j'ai rencontrées et mon inexpérience j'ai consulté pas mal de liens et de blogs avec les mêmes problèmes que moi. J'ai aussi découvert pas mal d'outils qui pourraient me servir dans le futur. Donc j'ai documenté une partie des liens que j'ai consultés : [Document Drive](#).

---

<sup>11</sup> [Site web TCPDF](#) & [GitHub TCPDF](#)

<sup>12</sup> [GitHub TCPDF](#)