

Rapport de stage

04/07/2021

BTS SIO 1E ANNÉE

Raphaël DUMAS

Santé Académie SAS

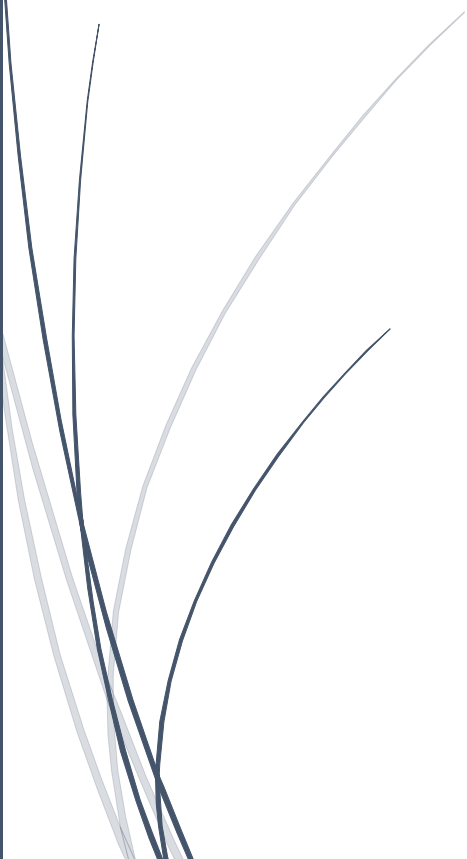


Table des matières

1.	Introduction.....	2
2.	Contexte	2
a.	L'entreprise.....	2
b.	Les Projets & Outils	3
3.	Le déroulé.....	3
4.	Conclusion	4
5.	Annexes	5
a	Annexes du premier projet	5
b	Annexes du second projet.....	6

1. Introduction

Je suis Raphaël DUMAS, étudiant en 1^e année de BTS SIO, admis en option SLAM, à l'Institution des Chartreux à Lyon. Ce cursus nous demande au cours de ses 2 années de formation de réaliser un stage de minimum 4 semaines sur chaque année. Durant cette première année, mon stage s'est déroulé dans l'entreprise Santé Académie SAS, durant 5 semaines du 31 mai au 2 juillet au poste de stagiaire développeur Backend. Les stages en entreprises sont très importants, quel que soit le domaine, car il permet en premier lieu d'acquérir de l'expérience d'entreprise, mais également des compétences plus techniques.

2. Contexte

a. L'entreprise

L'entreprise Santé Académie SAS est une société d'e-learning pour professionnels de santé, proposant des formations par vidéo sur leur site et application, le tout s'adressant aux médecins, infirmiers et pharmaciens. Leur programme s'étoffe de jour en jour. Elle compte aujourd'hui 19 employés, dont 3 stagiaires, et 4 en freelance.

Le service informatique est donc composé de 5 personnes, dont un stagiaire (moi), et une en freelance. Santé Académie voit le jour en janvier 2020. L'entreprise est issue d'une précédente entreprise, Airteach qui était elle-même issue d'une autre startup, Pangone.

Les activités de ces 2 précédentes entreprises sont désormais stoppées mais leurs technologies (Apps Web et Mobile) ont servi de base au lancement de Santé Académie. Santé Académie réunit donc l'équipe, l'expérience et la Tech des 2 précédentes entreprises en une seule dédiée uniquement au secteur de la Santé. Début 2020, le lancement se fait uniquement en ciblant les IDEL (Infirmier diplômé ayant exercé à l'hôpital, ou Infirmière Libérale) avec l'appui de plusieurs organismes de formations partenaires afin de disposer rapidement de contenus de qualité à faire valoir sur la plateforme.

Parmi ces formations on y trouve :

- L'ANDPC (ou DPC) : Agence Nationale du Développement Professionnel Continu
- Enzym : espace de formation pour infirmier libéraux (ou IDEL)
- MEDEO : entreprise de dispositifs médicaux
- Socoform : autre agence de formation continue pour professionnels de santé

L'idée directrice est donc de tester le plus vite possible l'appétence des professionnels de Santé pour la formation digitale. Ainsi, l'informatique est primordiale pour l'entreprise, car tout est basé sur leurs plateformes développées, passant par le site proposant leurs formations, base de leur activité.

b. Les Projets & Outils

Le premier projet devait se réaliser en PHP, langage majeure de cette partie de l'entreprise, avec le module PHP Excel, afin de répondre à un besoin exprimé par une collègue, concernant un export de données des sessions appartenant à un certain organisme. Certains exports existaient déjà, mais il fallait cette fois ci respecter un certain Template fourni (voir annexes) selon le type d'export (DPC, MEDEO, FIFPL, voir annexes). Il y avait donc une base existante, suffisante pour ne pas être perdu dès le début.

Les outils à ma disposition étaient :

- Un poste de travail Mac OS
- Une VM Debian 10 installée sur VMWare Fusion pour pouvoir réaliser et tester tous les ajouts sur un serveur local avant de l'envoyer en production,
- L'IDE PHPStorm, le plus adapté au projet,
- Le projet complet de l'entreprise (dans cette branche) avec une architecture Symfony
- Le repository GitHub de l'entreprise
- Des plateformes exclusives à l'entreprise comme le site Asana (gestion de projets et tâches) ou encore le groupe Slack (réseau social adapté à l'entreprise)

Le second projet était un script à réaliser se séparant en étapes :

- A partir d'un fichier CSV contenant 2 colonnes de données, un code d'une formation et son ID, stocker des informations dans un fichier JSON.
- Pour chaque ID récupéré, lire un URL dynamique (avec chaque ID), avec un cookie permettant de se connecter, puis récupérer le corps de la page HTML du site.
- Pour chaque corps HTML lu (sur le site de l'ANDPC), retourner les données récupérées (date, id et numéro de session) à partir d'un script
- Stocker chaque donnée dans l'ordre demandé dans un objet, puis insérer le tout dans un fichier JSON
- Adapter les fichiers en TypeScript et créer les interfaces, pour typer toutes les variables, fonctions et valeurs retournées, afin de ne pas créer de conflits.
- Afin d'utiliser le script, il faut rentrer une commande du type :
« node -i [nom du fichier lu en entrée] -o [nom du fichier écrit en sortie] -c [cookie de session pour accéder au site pour lire et récupérer toutes les données] »

Des exemples de code sont disponibles en annexe

3. Le déroulé

La première journée peut se résumer par une découverte de l'environnement, de l'entreprise dans sa globalité, des employés, des locaux, mais également des outils et des ressources utilisées.

La première semaine est donc constituée de prise de connaissance en profondeur du projet et d'une réalisation type, avec les requêtes SQL avec le Framework Symfony, ainsi que la réalisation en brut du projet.

La deuxième semaine à elle été consacrée à l'optimisation du programme, à la mise en forme, ainsi qu'à des réunions afin de faire le point sur l'avancée, et pour savoir si des modifications devaient être apportées. Le projet s'est déroulé sans accroc, malgré la principale contrainte : celle d'utiliser le Framework Symfony et la librairie PHP Excel dans des versions anciennes afin d'être prises en charge par le serveur, lui aussi assez ancien. Cela a donc ralenti la recherche et la lecture des documentations

respectives, certains aspects n'étant pas encore implémentés, ou bien n'existant plus dans les versions actuelles.

Le premier projet a ensuite été décliné en plusieurs versions, devant réaliser une tâche plus ou moins similaire pour différents partenaires : la première était pour l'ANDPC (Agence Nationale du Développement Professionnel Continu), la 2e pour l'organisme FIF-PL, et le troisième pour la start-up MEDEO, une entreprise de téléconsultation, devant alors revenir sur une version ou l'autre en fonction des demandes des organismes.

Le second projet était différent, puisque la demande était complètement différente. Elle provenait une nouvelle fois de l'organisme DPC, mais dans un cadre différent de lecture et récupération de données via un script. Le choix du langage était libre, j'ai donc choisi le NodeJS (puis TypeScript pour peaufiner) ainsi que Visual Studio Code pour IDE, étant le plus fourni pour ce langage. Les modules nécessaires étaient sync-request, pour réaliser la requête synchronisée et non en asynchrone, permettant d'attendre la fin d'exécution avant d'écrire le fichier JSON, cheerio, pour émuler JQuery avec des modules de CommonJS, fs pour permettre la lecture et l'écriture de fichiers, ainsi que commander pour récupérer les arguments de la ligne de commande. La commande compterait 3 arguments : un fichier d'entrée (pour l'instant en CSV, généré à partir d'une requête SQL), un de sortie (pour l'instant en JSON) et le cookie.

4. Conclusion

Sur les 2 projets que j'ai eues à réaliser durant ces 5 semaines de stage, celui utilisant le module PHPEXCEL est totalement terminé, pour les 3 exports : DPC, FIF-PL & MEDEO. Quant au second projet, celui en NodeJS/TypeScript, il ne peut être considéré à 100% terminé, car bien que le programme soit tout à fait fonctionnel, il m'aura manqué du temps pour la dernière étape : celle de tout afficher sur une page web.

D'un point de vue personnel, j'ai remarqué au fil des jours une évolution, aussi bien dans le domaine technique que professionnel, bien que l'adaptation lors des premiers jours dans l'environnement m'ai déstabilisé, n'osant pas toujours poser les questions.

En définitive, ce stage m'aura été plus que bénéfique sur tous les aspects. Tout d'abord du côté professionnalisant, l'un des objectifs majeurs du stage, ayant beaucoup appris sur le fonctionnement d'une entreprise, en son sein, et à l'extérieur. Également, devoir gérer personnellement des projets m'aura permis une meilleure gestion des responsabilités et des tâches assignées, ainsi qu'une meilleure autonomie. Tous ces projets auront évidemment élargi mes connaissances techniques, ayant alors avant seulement des connaissances relativement théoriques sur les outils et langages utilisés (PHP, GitHub notamment).







5. Annexes

a Annexes du premier projet

Voici un des Template fournis par un organisme externe, et dont j'ai du respecter la mise en forme donnée pour répondre au mieux à la demande (les données personnelles ont été masquées) :

Nom de l'organisme : Enzym Formation						
Identifiant : 6986						
N° Action / Programme : 69862000004		N° de session : 3		Volume horaire déclaré : 420 minutes / 7 heures		
Description des activités hors connexion : volume horaire déclaré :						
Non-applicable						
Description des activités connectées : adresse url : volume horaire déclaré : 420 minutes / 7 heures						
0. Bienvenue 1. Rappel des textes de loi 2. Les règles de la prescription 3. Pansements dans la NGAP 4. La législation, le dossier de soin en pratique et le droit à l'image 5. Elimination et transport des DASRI 6. Rappel anatomique et principes de cicatrisation cutanée 7. Cicatrisation dirigée et impasses de cicatrisation 8. Bactériologie 9. Déterision : cadre réglementaire et principes de base 10. La déterision mécanique en pratique 11. Larvothérapie, fibrine et découpe de nécrose 12. La douleur 13. Principes, tableau d'évaluation et mise en pratique de la méthodologie 14. Cadre, pansement idéal et classification 15. Pansements passifs 16. Pansement interactifs 17. Pansements actifs 18. Récapitulatif 19. Les ulcères 20. Les plaies du pied diabétique 21. Les escarres						
NOM	Prénom	N°RPPS ou Adeli	A réalisé les activités	Jour de la 1ère	Jour de la dernière	Total du temps
			oui	10/4/2020	30/4/2020	2563
			oui	2/4/2020	30/4/2020	649
			oui	6/4/2020	30/4/2020	571
			oui	2/4/2020	25/4/2020	1150
			oui	18/4/2020	28/4/2020	460
			oui	1/4/2020	30/4/2020	438
			oui	3/4/2020	29/4/2020	496
			oui	15/4/2020	30/4/2020	492
			oui	4/4/2020	20/4/2020	1490
			oui	1/4/2020	7/4/2020	433
			oui	2/4/2020	24/4/2020	564
			oui	1/4/2020	29/4/2020	1632
			oui	5/4/2020	28/4/2020	433
Je soussigné(e) Brigitte BAX agissant en ma qualité de co-gérante de l'organisme ENZYM atteste que les personnes dont les noms figurent ci-dessus ont suivi des séquences non présentielle de l'action ou du programme n° 69862000004, session n°1.						
Je tiens à la disposition de l'Agence nationale du DPC les documents attestant de ces participations.						
à	Caluire et Cuire		Cachet de l'organisme :			
le	16/3/2020					

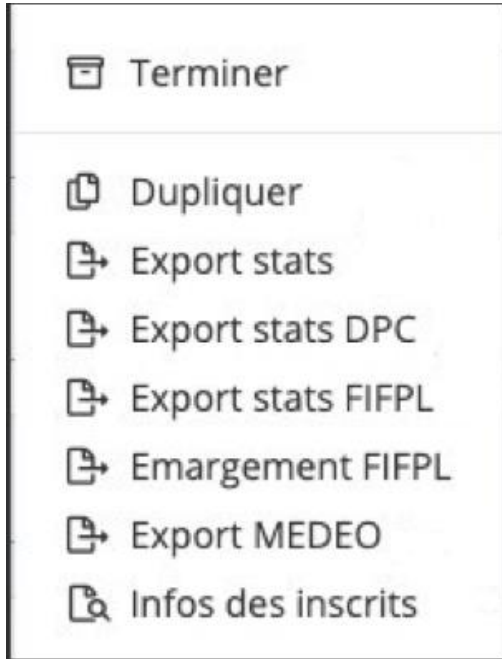
Afin de pouvoir utiliser les outils réalisés lors de mon stage, il faut d'abord se rendre sur la page interne de l'entreprise, réalisée via Airteach. Pour le personnel, le mode édition est disponible, et après, sur la page listant les parcours/sessions, il est possible de choisir la formation de notre choix :

Titre	Etapes	Nombre d'utilisateurs	Nombre de sessions	Options
Formation Oncologie	1	0/17	0 / 14	 
Formation NGAP	3	0/2	14 / 24	 
Formation Plaie et Cicatrisation	3	0/16	19 / 96	 

Après avoir choisi, ces informations à propos de la session apparaissent, et dans les options, un menu déroulant listant les différentes options et exports possibles :

ID	Titre	Code	Date de début	Date de fin	Statut	Séances publiques	Tags	Procédure début	Procédure complétion	Attestation de fin	Options
#112	NGAP-DPC-15-05-2021	NGAP-DPC-15-05-2021	15/05/2021 09:00	15/05/2021 22:55	EN COURS	NGAP				Pas de certif	...

Et lorsque l'on clique sur les options, tous les exports existant apparaissent, incluant ceux sur lesquels j'ai travaillé, qui sont totalement fonctionnels :



b Annexes du second projet

Le fichier csv lu par le programme ressemble à ceci : 2 colonnes un pour le code de la session (abréviation du thème) ainsi que l'identifiant qui lui est associé

```
code,id  
PLAI,"96135"  
ONCO,"107680"  
PRADO,"100669"  
DIABETE,"96417"  
ALZ,"108474"  
DSI,"99427"  
DOULSM,"96445"  
BURNOUT,"107550"  
PALLIA,"107306"  
PICC,"99428"  
ADDIC,"99431"  
TABAC,"108456"  
HYPNO,"109743"  
APNE,"99434"  
SEXUA,"96886"  
PRADOIC,"110380"
```

```

"use strict";

/**
 * Author : Raphaël DUMAS
 * Date : 01/07/21
 * Version : 1.4
 */

export const __esModule = true;

import parserHTML from "./parserHTML.js";
import getterHTML from "./getterHTML.js";
import writerJSON from "./writerJSON.js";
import readerCSV from "./readerCSV.js"

import commander from "commander"
//imports needed : others files and modules

var program = new commander.Command();

//used to initialize the parameters that will be loaded from the command line
program
    .option('-i, --input <file>', 'input file')
    .option('-o, --output <file>', 'output file')
    .option('-c, --cookie <arg>', 'cookie to connect');
program.parse(process.argv);

var inputFilename = '';
var outputFilename = '';
var cookie = '';
var json = [{}];

//load the command line args
var options = program.opts();
if (options.input) {
    inputFilename = options.input;
}
if (options.output) {
    outputFilename = options.output;
}
if (options.cookie) {
    cookie = options.cookie;
}
var listActions = readerCSV.read(inputFilename);

//get all data for all codes

for (var i = 0; i < listActions.length; i++) {
    var htmlAction = getterHTML.getByActionId(listActions[i].id, cookie);

```



```

var listSessionsForAction = parserHTML.parse(listActions[i].code,
htmlAction);
if (listSessionsForAction.sessions.length === 0) {
    console.log(listSessionsForAction.code + ' : aucune session n\'a été
trouvée pour ce code');
}
json = [{ code: listSessionsForAction.code, sessions:
listSessionsForAction.sessions }];
}
writerJSON.write(outputFilename, json);

```

Ci-dessus se trouve le fichier index.js, central à tous les autres, utilisant toutes les fonctions créées dans les différents fichiers pour écrire le fichier JSON. On y trouve donc les imports des autres fichiers, la création d'option pour lire les arguments de la commande. Le programme ne lira donc qu'une seule fois la page, et pour toutes les informations récupérées, exécutera toutes les actions de la boucle : lire le corps HTML, récupérer les données (dans un objet), puis à partir de toutes les informations, écrire le fichier JSON. Ce fichier sera de cette forme (avec plus de données).

```

{
  "code" : {
    "session" : [
      "number" : 0,
      "id" : 0,
      "date" : "01/07/2021"
    ]
  }
}

```

Ici on trouve le fichier parserHTML.js, qui a été codé à partir d'une fonction JQuery donnée par mon tuteur de stage. La librairie Cheerio est ici importée car la plupart des librairies utilisés par les différents fichiers sont issues du module esModule, or il est également nécessaire d'avoir JQuery pour la fonction de base, qui récupère tous les éléments nécessaires (numéro, id, date) pour construire le futur export. La librairie Cheerio permet donc d'avoir accès aux fonctions de JQuery même avec un autre module utilisé.

Les variables sont retournées puis exportées sous forme d'objet, afin de simplifier l'écriture du fichier JSON créé à la fin de l'exécution.

```
"use strict";
export const __esModule = true;
import cheerio from 'cheerio'
/**
 *
 * @param code : code reference of a formation, linked with all parameters and
data
 * @param html
 * @returns object
 */
var parse = function (code, html) {
  var $ = cheerio.load(html);
  var listSessions = [];
  var i = 0;
  $('.sessionNum').each(function () {
    var _a;
    var session = { number: 0, id: '', date: '' };
    var $tr = $(this).closest('tr');
    console.log($tr);
    session.number = (parseInt($(this).text().replace('Session ', '')));
    session.id = (_a = (($tr.find('.fa-eye').closest('a').attr('href') ||
'').split('/').pop())) !== null && _a !== void 0 ? _a : '';
    session.date = ($tr.find('td ul li').text().replace(/(.*)\s > .*/,
'$1'));
    listSessions.push(session);
    i++;
  });
  return { code: code, sessions: listSessions };
};
export default { parse: parse };
```

Enfin, ici se trouve le fichier `writerJSON.js`, qui comme son nom l'indique contient la fonction qui permettra d'écrire le fichier JSON, à partir d'une variable nommée « `json` », contenant, sous la forme d'objets, toutes les données récoltées au préalable, et grâce à la méthode `stringify`, permet de créer le fichier.

```
"use strict";

export const __esModule = true;
import fs from 'fs'

/**
 *
 * @param filename : name of the file containing the data
 * @param json : JSON object containing the data
 */

var write = function (filename, json) {
  fs.writeFile(filename, JSON.stringify(json), function (err) {
    if (err) {
      throw err;
    }
    console.log('Written !');
  });
};
export default { write: write };
```