****

**2020 / 2021**

NOM  Jehenne

PRÉNOM  Arthur

Stage scientifique et technique I2 (S7)

Rapport de stage / emploi

ENTREPRISE : *University of Malta*

DATES : *Du « 01/09/2021 » au « 26/09/2021 »*

SUJET MISSIONS : *JavaScript / node.js*

TUTEUR ENTREPRISE : *Monsieur Mark Vella,*

*Professeur en informatique*



*Je soussigné, Arthur Jehenne, étudiant à l'ESEO, atteste avoir pris connaissance du contenu du Règlement intérieur de l'École et de l'engagement de « non-plagiat ». Je déclare m'y conformer dans le cadre de la rédaction de ce document. Je déclare sur l'honneur que le contenu du présent mémoire est original et reflète mon travail personnel. J'atteste que les citations sont correctement signalées par des guillemets et que les sources de tous les emprunts ponctuels à d'autres auteur(e)s, textuels ou non textuels, sont indiquées. Le non-respect de cet engagement m'exposerait à des sanctions dont j'ai bien pris connaissance.*

*Fait à Malte le 2 octobre 2021*

REMERCIEMENTS

Tout d’abord, je tiens à remercier mon école, l’ESEO pour m’avoir permis de réaliser un stage à l’étranger au sein de mon parcours universitaire. Je souhaite aussi remercier l’Université de Malte pour m'avoir accueilli au sein de son établissement, et tout particulièrement mon maître de stage, le docteur Mark Vella. Ce dernier a su m’épauler, me soutenir et m’apporter les connaissances nécessaires à la réalisation de mon stage. De nature disponible, toujours à l'écoute, il a su me consacrer du temps.

Par ailleurs, j’aimerai remercier Madame Saliba Katia, qui appartient au département management des étudiants de l’Université de Malte. Elle a su répondre toujours très vite à mes différentes questions pour l’organisation du stage et accompagner au mieux dans mes démarches administratives relatives à mon départ à l’étranger.

Enfin je voudrais remercier Guillaume Cheruel. Ami et colocataire il a su m’accompagner dans ma démarche universitaire par le biais d’une bonne humeur journalière et d’une motivation constante tout au long de la semaine.

SOMMAIRE

Synthèse : page 5.

Introduction, page 6.

Première partie : L'environnement et le contexte du stage, page 7 à .

* Histoire de Malte
* La création de l’université de Malte
* L’université de Malte de nos jours
* Le contexte du stage

Deuxième partie : Les activités propres et détaillées pendant mon stage, page 11 à 26.

* Le déroulement de mon stage

Troisième partie : Rapport personnel, page 27 à 30

Conclusion page 31 à 33.

SYNTHÈSE

Sujet :

Création d’un explorateur de fichiers en node.js

Domaine :

Informatique

Université :

Université de Malte

Date du stage :

01/09/2021 au 27/09/2021 (13 semaines)

Relations :

Docteur Mark Vella (maître de stage)

Planning résumé :

* Mois de septembre :
* Découverte des environnements
* Apprentissage du langage JavaScript
* Réalisation de différents exercices en JavaScript
* Mois d’octobre :
* Apprentissage et compréhension de la relation serveur/ client avec Node.JS
* Réalisation de différents exercices en Node.JS
* Réalisation de petits projets comme parking.js.
* Réalisation de mon projet terminal : explorateur de fichiers
* Mois de novembre :

- Début de l’apprentissage de Vue.js

- Finitions de l’explorateur de fichier

Contrainte et difficultés rencontrés :

* Barrière de la langue, technique au départ mais la compréhension s'est améliorée au fur et à mesure du temps.
* La compréhension de la relation serveur/client
* Le système asynchrone et de synchrone de node.js

Introduction :

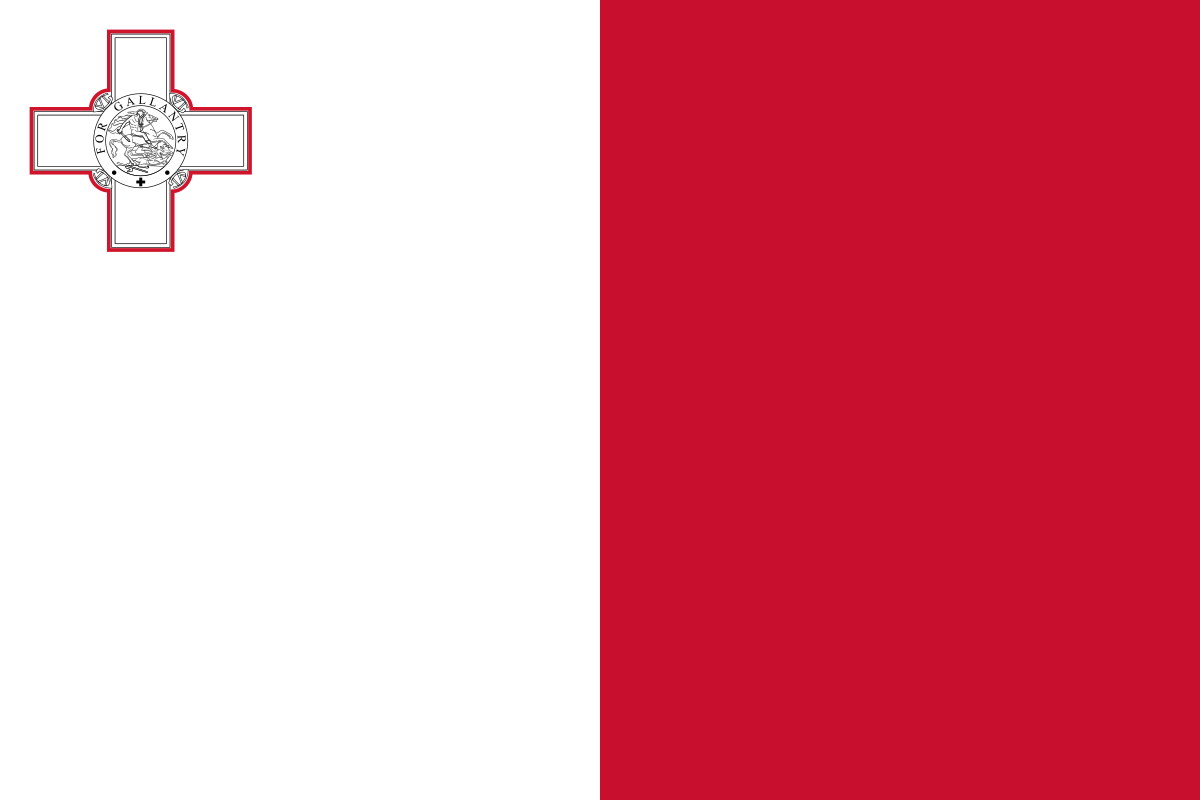
Tout d’abord, j’étais désireux de faire mon stage à l’étranger. Le but étant d’améliorer mon anglais, découvrir de nouvelles personnes et un rythme de vie différent notamment au niveau du système scolaire étranger. Par nature, j’aime découvrir de nouveaux horizons notamment en réalisant des voyages. Afin de m’inscrire dans cette volonté, j’ai recherché activement des destinations via les offres de stages de l’école, les expériences des anciens élèves ayant réalisé ma formation au sein de l’école, mais aussi via internet. Je n’avais pas de préférence au sujet du lieu de destination ou du lieu de réalisation du stage. J’étais prêt à m'adapter tant au rythme professionnel d’une entreprise, qu’au rythme scolaire d’une université.

Néanmoins, j’étais attaché à l’idée de réaliser un stage dans le domaine de l’informatique. Je suis moins attaché à la découverte et l'approfondissement du domaine de l'électronique. Dans un premier temps, mes recherches se sont tournées vers les pays du nord à savoir l’Angleterre, l’Irlande et le Pays de Galles. Ne trouvant pas un stage correspondant à la fois à mes envies mais aussi à mes besoins, mes recherches se sont portées vers l’international.

Malheureusement, j’ai dû essuyer quelques refus avant de parvenir à trouver mon stage. La période de la pandémie n’a pas été de tout repos. En février, l’Université de Malte a répondu positivement à ma demande de stage. Ainsi, après avoir eu un échange en visioconférence fin-février avec Monsieur Mark Vella, d’une trentaine de minutes, ce dernier m’a convaincu de venir vivre une expérience universitaire au sein de l’université de Malte. Les missions proposées et l’objectif du stage à atteindre m'ont convaincu dans l’idée de réaliser celui-ci. Le stage envisagé alliait l’informatique et l’apprentissage approfondi de la langue anglaise, deux notions qui m'intéressent tout particulièrement.

PREMIÈRE PARTIE : L’ENVIRONNEMENT ET LE CONTEXTE DU STAGE

Tout d’abord, la première partie de ce rapport sera consacrée à l'environnement et le contexte de mon stage. J’ai réalisé mon stage au sein de l’université de Malte. Je suis arrivé en août à Malte, soit quelques jours avant le début de mon stage, j’ai pu découvrir de nombreuses facettes de cette île qui m’était jusqu'alors inconnue (I). L’université de Malte au sein de laquelle j’ai réalisé mon stage n’est pas une université récente (II). Elle est une des universités situées sur le campus le plus étendu de l'île (III). L’université de Malte est le fruit d’une longue construction. Mon stage a duré trois mois. Au cours de ce laps de temps j’ai réalisé un projet que je m’attarderai à développer dans une dernière sous-partie consacrée au contexte de mon stage (IV).



1. HISTOIRE DE MALTE

La construction de l'île de Malte commence à l'Antiquité, lorsque des groupes de pêcheurs venant de Sicile s'y installent afin de mener une activité économique par l’exploitation d’une richesse qui leur est rare : la pêche. C’est la création du premier peuple maltais. La position stratégique de Malte n’a pas échappé à d’autres colonies qui se sont empressés de s’y implanter. En 1798, Napoléon Bonaparte tente de conquérir et d’envahir l'île en vain. Le peuple maltais dans un but de conservation de son île, décida d’appeler le peuple anglais afin que ces derniers les aident à repousser l’envahisseur. Deux années plus tard, Napoléon Bonaparte cède et retire ses troupes de l'île. Les Britanniques décident de rester sur l'île. Malte devient une colonie Britannique. Le 21 septembre 1964, l’île de Malte obtient son indépendance. Enfin, en 2004 Malte intègre l’Union européenne. Aujourd’hui Malte possède 525 000 habitants pour une superficie de 316 km². La capitale actuelle est La Valette, remplaçant l’ancienne capitale Mdina. La Valette recense 5730 habitants en 2016.



1. LA CRÉATION DE L'UNIVERSITÉ DE MALTE

L’université de Malte est le fruit d’une longue construction. Cette dernière a été créée il y a environ quatre cents ans. L’université de Malte est un des pôles de nombreux échanges universitaires internationaux.

Les origines de cette dernière remontent à la fondation du Collegium Melitense. En 1675, est mis en place au sein de l'université de Malte un enseignement nommé enseignement médical. Ce dernier a été et est encore considéré comme les prémisses de notre faculté de médecine française. Napoléon Bonaparte ayant tenté de coloniser l'île de Malte (v.supra) a décidé de mettre un terme à l'enseignement universitaire maltais une nouvelle fois en vain car en 1800, l’université de Malte est rétablie. La colonisation britannique s’est aussi ressentie au sein de l’université de Malte qui n’a cessé de mettre en œuvre une nouvelle réglementation de l’enseignement afin de coller au mieux au modèle anglais.

Aujourd’hui, l’Université de Malte est une des plus hautes institutions d’enseignements de l'île. Elle dispense des enseignements dans tout un tas de domaines tel que le domaine de l’informatique. Elle est ouverte à tous les étudiants ayant les qualifications requises, elle ne les soumet pas au passage d’un concours.

1. L'UNIVERSITÉ DE MALTE DE NOS JOURS

J’ai réalisé mon stage sur le site de l’Université de Malte (Università ta’ Malta en maltais), plus précisément au sein de la faculté ICT (Information and Communications Technology) . Cette dernière se localise sur le campus principal nommé Msida (v.confra) lui-même composé de [quatorzes facultés](https://www.um.edu.mt/academicentities/faculties) , d’une quinzaine d'instituts, de [centres](https://www.um.edu.mt/academicentities/centres) interdisciplinaires , de trois [écoles](https://www.um.edu.mt/academicentities/schools)  dont une spécialisé dans les arts du spectacle et d’un [collège junior](https://www.um.edu.mt/academicentities/juniorcollege) (correspondant à un collège secondaire français). Le campus de Msida n’est pas unique. Ce dernier co-existe avec trois autres campus situés à [La Valette](https://www.um.edu.mt/campuses/valletta), à [Marsaxlokk](https://www.um.edu.mt/campuses/marsaxlokk) et à [Gozo](https://www.um.edu.mt/campuses/gozo) .



En 2013, le campus comprenait plus de 600 professeurs, environ 11 000 étudiants, dont 650 étudiants internationaux de 77 pays. La langue d'enseignement est l'anglais. Plus de 3500 étudiants obtiennent leur diplôme dans diverses disciplines chaque année.



L’université de Malte a une superficie totale de 250 207 m². Les plans de construction du campus de Msida ont commencé à la fin des années 1950 après que les locaux d'origine de l'université à La Valette aient été jugés trop petits et désuets. La faculté ICT est logée dans un bâtiment en acier et en verre qui a été construit entre 2009 et 2013 (numéro 35 entouré sur la photo).

La devise de l’Université de Malte est la suivante « Ut Fructificemus Deo » qui par traduction latine signifie “ Nous devrions porter du fruit à Dieu”. Cette dernière a été proposée en 1923. En 1968, le nouveau campus de Msida a été inauguré. En 2017, le logo a été modifié. Le logo de l’université de Malte met en avant l’identité de cette dernière.



1. LE CONTEXTE DU STAGE

J’ai effectué mon stage sur une période de trois mois. Mon stage a commencé le premier septembre et il s’est terminé le vingt six novembre. Mes horaires étaient les suivants, je me rendais à la faculté tous les jours. De manière hebdomadaire, j’étais présent au sein de l’université de huit heures à dix-sept heures.

Les enjeux et objectifs principaux de mon stage :

Les enjeux de mon stage sont nombreux. D’abord, j’ai pu découvrir le travail d’un chercheur au sein d’une université, notamment au regard de ses missions. L’enjeu était de taille. Par ailleurs, ce stage avait pour but d'approfondir et mettre en application mes connaissances découvertes depuis le début de mon parcours scolaire au sein de l’ESEO.

Les principaux enjeux de mon stages on été d’apprendre le JavaScript, de mettre en place une API, d’utiliser postman

Les difficultés liées à la réalisation de mon stage :

Mon rythme scolaire est resté inchangé. Mes horaires précédemment explicités sont similaires aux horaires réalisés au sein de mon cursus au sein de l’ESEO. Néanmoins, une première difficulté s’est présentée à mois quant à l’intégration au sein de l’université. En effet, les étudiants commençant leurs années scolaire à la fin du mois de septembre, l’université était vide. Par ailleurs, les cours en distance sont maintenues en raison de la pandémie mondiale, ce qui ne facilite pas le contact avec les autres étudiants.

DEUXIÈME PARTIE : MES ACTIVITÉS PROPRES ET DÉTAILLÉES PENDANT MON STAGE

Concernant la réalisation, à proprement parlé de mon stage. J’ai pu réaliser au cours de ces trois mois de stage de nombreuses missions, exercices, pour parvenir à la réalisation de mon projet terminal (I). Ces dernières seront exposées. Le déroulement de mon stage sera soumis à une auto-critique et une auto-évaluation (II).

1. LE DÉROULEMENT DE MON STAGE

* **Sujet**

Le sujet de mon stage a été pour l’essentiel de maîtriser les langages web : JavaScript et node.js en regardant des tutos et en réalisant des exercices consultés et corrigés hebdomadairement par mon maître de stage.

* **Le cahier des charges**

Le cahier des charges était bien rempli. En effet, j’ai dû réaliser une multitude de codes dans un temps imparti et court. De cette réalisation j’ai pu apprendre à utiliser et manier de nombreux langages nouveaux et différents, mais aussi différentes librairies.

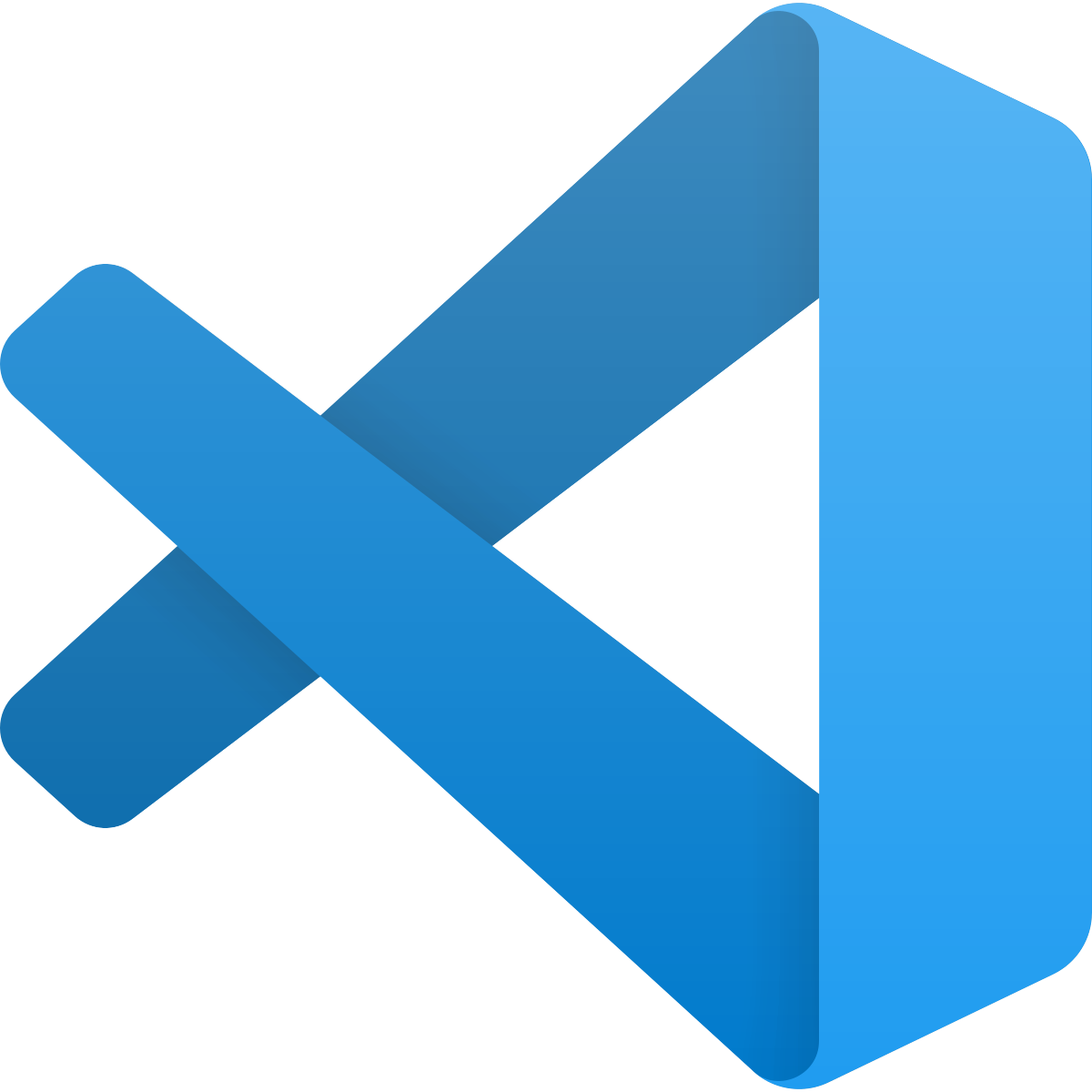
Mon premier mois était consacré à la découverte de l’environnement de mon stage. Je devais effectuer des petites tâches afin de m’imprégner des langages et des librairies. Ces dernières relevaient notamment de la pratique du JavaScript afin de réaliser mon projet terminal. Je devais donc suivre plusieurs tutoriels donnés par mon maître de stage sur les sites comme w3School ou tutorialsPoint. J’ai aussi eu l’occasion d’en trouver par mes propres moyens tel que JavaTpoint. Sur youtube, j’ai eu l'occasion de m’exercer sur des vidéos telles que PrimFX (version française) ou encore ProgramminwithMosh (version anglaise). Ces dernières m'ont permis d’intégrer l'utilisation de Javascript. Je devais rendre compte de manière hebdomadaire à mon maître de stage mon avancée dans l’intégration de ces différents tutoriels.

Je devais apprendre à écrire et envoyer des données sur une page créée par mes soins. De plus, je devais apprendre à maîtriser la création d'alertes, de switch et de cases. Je devais présenter à mon maître de stage et soumettre à sa correction la réalisation de ces différents exercices.

* **Les moyens**

Pour réaliser ce stage, j’ai essentiellement travaillé sur mon ordinateur. Je disposais à la fois du logiciel windows mais aussi de linux. J’effectuais mes travaux dans une salle d’informatique, plus précisément située dans le secteur sécurité et de recherche de l’université. Ainsi, j'étais en contact direct avec certains chercheurs et étudiants, de temps à autre, qui se rendaient dans cette salle afin de travailler. Néanmoins, ces contacts étaient succinct et peu fréquents. La majorité du temps, je travaillais seul, en autonomie. Par conséquent, j’avais un espace de travail optimal et très avantageux.

* **Les compétences acquises** :



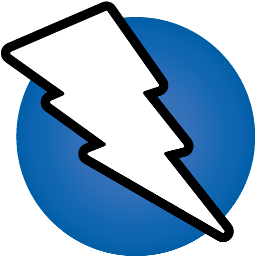
Visual Studio Code

Pour le codage, j’ai utilisé le logiciel Visual Studio code. Ce dernier m’avait été fortement conseillé par mon maître de stage. De plus, j'avais eu l’occasion par des recherches actives sur la question de me rendre compte que ce logiciel était préféré pour sa qualité et son efficacité. Par conséquent, l’utilisation de ce logiciel m’a paru tout à fait appropriée.

Visual Studio est un logiciel d’éditeur de code. Il permet d’éditer du code, mais aussi de tester, de lancer et de sauvegarder ce dernier dans un espace de stockage conséquent et utile en la matière. Outre cet espace, le logiciel est adaptable. Il contient une possibilité de nombreuses extensions téléchargeables au sein même du logiciel. Ce qui m’a permis de constater une nouvelle fois, la caractéristique d’effectivité du logiciel. L’outil est rapide et permet une utilisation optimale des données traitées et des langages utilisés.

Au-delà de son aspect théorique, l’outil a révélé une nouvelle fois son aspect pratique par une de ces caractéristiques : la fonctionnalité palette. Cette dernière est utile à bien des égards, en effet elle permet de lancer un code assorti de diverses fonctionnalités téléchargées antérieurement. La fonctionnalité palette m’a permis de visualiser du code HTML via le logiciel, de manière directe, et de lancer des serveurs web sans devoir au préalable les créer.

Une seconde caractéristique, particulièrement utile du logiciel s’illustre dans la fonctionnalité terminal. Le terminal intégré au sein du logiciel Visual Studio permet de lancer un code sans avoir besoin au préalable d’aller rechercher le dossier spécifique qui lui est associé à l’aide d’une commande. Autrement dit, la commande cd devient secondaire et n’a plus besoin d'être mise en œuvre.



Owasp Zap

Owasp zap (zed attack proxy) est un logiciel open-source qui m’a permis de réaliser au mieux mon projet et les divers exercices que mon maître de stage m’a confiés. Ce dernier me laissant une grande autonomie dans la gestion de mes tâches. A nouveau l’utilisation de cet outil m’a été recommandé par le docteur Mark Vella.

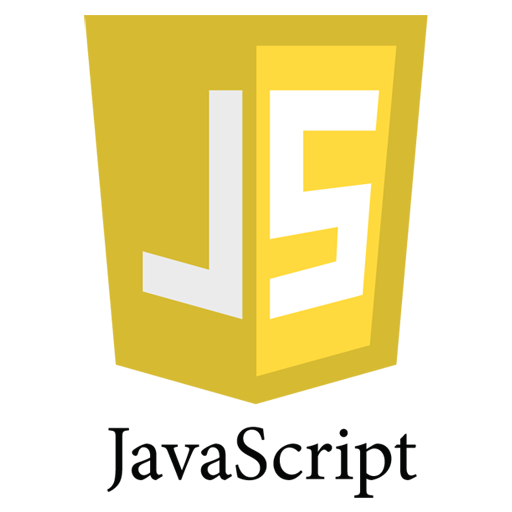
Le logiciel permet d’identifier et d’étudier les différentes requêtes arrivantes et sortantes des applications créées préalablement. Ces dernières peuvent être de tout genre. Autrement dit, l’outil permet d’analyser les trafics existant entre le serveur et le client. J’ai pu découvrir le fonctionnement de ce logiciel après avoir réalisé un test. Pour ce faire, j’ai décidé d'attaquer mon application à l’aide du proxy, ce dernier se définissant comme un composant logiciel informatique permettant de surveiller et faciliter les échanges entre deux zones. Ce qui m’a permis de conclure à l’efficacité et au rôle nécessaire du logiciel dans la réalisation de mon projet et dans la résolution de mes différents exercices.

Enfin, le logiciel Owasp Zap a montré son efficacité par le biais de son second versant. L’outil permet d’analyser la sécurité de diverses pages web. Pour ce faire, une nouvelle attaque est nécessaire.



Postman

Postman est un logiciel pour tester l’API rest. C’est un client HTTP qui permet de créer et de tester les requêtes HTTP. A l’aide de ce logiciel on peut créer des requêtes POST et GET.



JavaScript :

Le JavaScript peut se définir comme un outil informatique permettant par le biais d’un langage de programmation d’appréhender des mécanismes diverses et complexes relatifs à une page web. Aujourd’hui, ce langage est connu par une majorité de personnes s’intéressant fortement ou de loi aux outils informatiques. Anciennement appelé LiveScript, le langage a changé de dénomination. Le langage Javascript a été développé par Netscape, une entreprise d’informatique américaine créée en 1994. Il est principalement utilisé pour la conception de sites web, mais aussi pour la réalisation de diverses applications. Des millions de pages web et d'applications utilisent JavaScript.

Ce langage de programmation permet de créer et mettre en œuvre des mécanismes informatiques complexes. JavaScript permet la mise à jour de pages web de manière dynamique. Le langage est dit “léger” en raison du peu d'interactions avec le serveur. Le Javascript peut être écrit directement dans un code HTML ou dans un code unique et propre à ce dernier. Les scripts sont fournis et exécutés en texte brut. Ils n'ont pas besoin de préparation ou de compilation spéciale pour fonctionner.

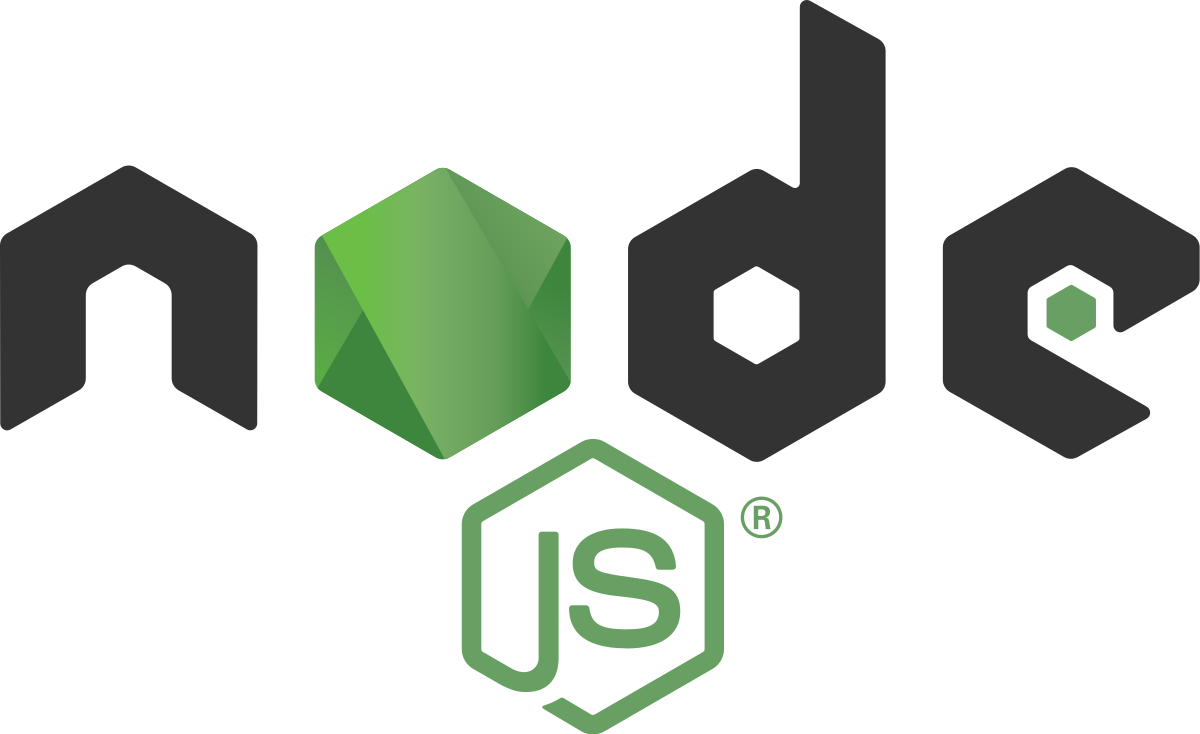
Le principal intérêt de Javascript c’est sa capacité à fournir une pluralité d’exécution. Autrement dit, il peut à la fois être exécuté du côté du navigateur, mais aussi bien du côté du serveur

Leurs implémentations permettent au script côté client d'interagir avec l'utilisateur et de créer des pages dynamiques.

Ce langage est similaire sans la syntaxe au Java mais aussi au C++ ainsi différentes fonctions fonctionnent de la même manière.

Le Javascript ne fournit pas d'accès de bas niveau à la mémoire ou au processeur, car il a été initialement créé pour les navigateurs qui n'en ont pas besoin.

Les capacités de JavaScript dépendent grandement de l'environnement dans lequel il s'exécute. Par exemple, [Node.js](https://wikipedia.org/wiki/Node.js) prend en charge les fonctions qui permettent à JavaScript de lire/écrire des fichiers arbitraires, d'effectuer des requêtes réseau, etc.



NODE.js

Node.js est le principale “langage”  dont je me suis approprié durant ce stage. C’est un environnement qui permet d’exécuter du JavaScript côté serveur. Il permet de séparer la partie script à la partie qui demande l’accès au fichier. Donc dans un même intervalle de temps on va pouvoir gérer beaucoup plus de concurrence par rapport à un script bloquant.

Node.js utilise le moteur JavaScript v8 qui permet de compiler le code javascript en code machine optimisé. Cela permet à node.js d’être très performant.

Ce dernier permet de créer des serveurs web mais aussi une multitude de besoins.

Node.js utilise un modèle asynchrone (un système non bloquant) qui permet de prendre en charge une multitude de requêtes en un minimum de temps. Tandis que la plupart des autres serveurs web utilisent des modèles dit synchrones (bloquants).

Une image contenant texte, piscine à balles, graphiques vectoriels

Description générée automatiquement

JSON

Json (JavaScript object Notation) est un langage dit léger. Il permet un échange de données entre le serveur et l’utilisateur. Il est utilisable par de nombreux types de langages(javascript, php). De plus sa compréhension et sa lecture est très facile.

JSON est un format texte totalement indépendant de tout langage. Son apprentissage est facile et intuitif.

Il est habituellement utilisé pour structurer et transmettre des données sur des sites web (par exemple, envoyer des données depuis un serveur vers un client afin de les afficher sur une page web ou vice versa)

Vue.Js

Vue.js est un framework, technologie en javascript qui permet de créer et de simplifier des interfaces dynamiques en JavaScript à l’aide de HTML. Il est donc utilisé pour la construction d'interfaces utilisateurs. C’est une technologie très récente car elle est apparue dans les années 2014.

Vue.Js a de nombreux points positifs mais étant donné que c' est un framework nous pouvons l’utiliser dans un projet déjà existant. On peut aussi créer des applications autour de Vue.js.

* **Le plan d’organisation prévisionnel**

Semaine 1 :

* Découverte et installations des environnements
* Premier code en JavaScript

Semaine 2 :

* Création d’un jeu (trouver le bon nombre)
* Création d’une page personnalisée avec différentes langues puis montre la localisation
* JavaScript orienté objet
* Création de classe
* Premier code en JSON

Semaine 3 :

* Phénomène Asynchrone / Synchrone en JavaScript
* Notions Promise /CallBack
* Apprentissage de l’Ajax

Semaine 4:

* Premier pas en Node.js (création de serveur, utilisation des modules…)
* Création d'événements en Node.js (module events)
* Utilisation de buffer

Semaine 5:

* Utilisation des méthode get /post
* Utilisation de JSON
* Création du répertoire de donnée

Semaine 6 :

* Utilisation de child\_Process
* Utilisation de Middleware ( se connecter en tant qu’administrateur et ou client)
* Utilisation des streams

Semaine 7 :

* Utilisation de express\_fileupload,multer...
* Code en JSON
* Utilisation OwaspZap
* Début de réalisation de l‘explorateur de fichier

Semaine 8 :

* Utilisation Postman
* Réalisation de l’explorateur de fichier

Semaine 9 :

* Découverte de Vue.js
* Découverte de Axios
* **Missions du stage**

1. Maîtrise de Javascript (côté navigateur web)
2. L’API : Application Programming Interface

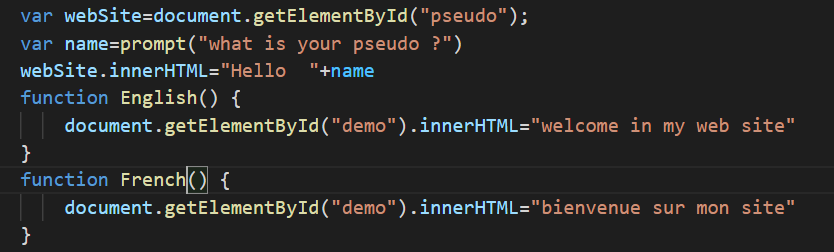
Une API(Application Programming Interface) est une interface qui permet de faciliter la création d’une page web. Elle fournit une liste d'opérations. Ainsi il permet la réalisation d’opérations complexes en toute simplicité car l’utilisateur n’est pas obligé de comprendre sont fonctionnement. Il suffit juste de choisir son API et de savoir si elle est utile ou non pour l’application. Elle permet donc un gain de temps car elle utilise régulièrement d'autres logiciels. Différents types d’API existent comme l’API reste orienté node.js dont je parlerai ultérieurement.

En JavaScript les API sont séparés en 2:

-Premièrement les API intégrés aux navigateur web que l’on utilise immédiatement comme l’API DOM( manipuler le HTML d’une page), l’API localisation ( permet l’accès à la localisation de l’utilisateur)

- Deuxièmement, les API externes. C'est-à- dire des API qui utilisent d'autres logiciels. Tel que Google Maps, API Youtube…

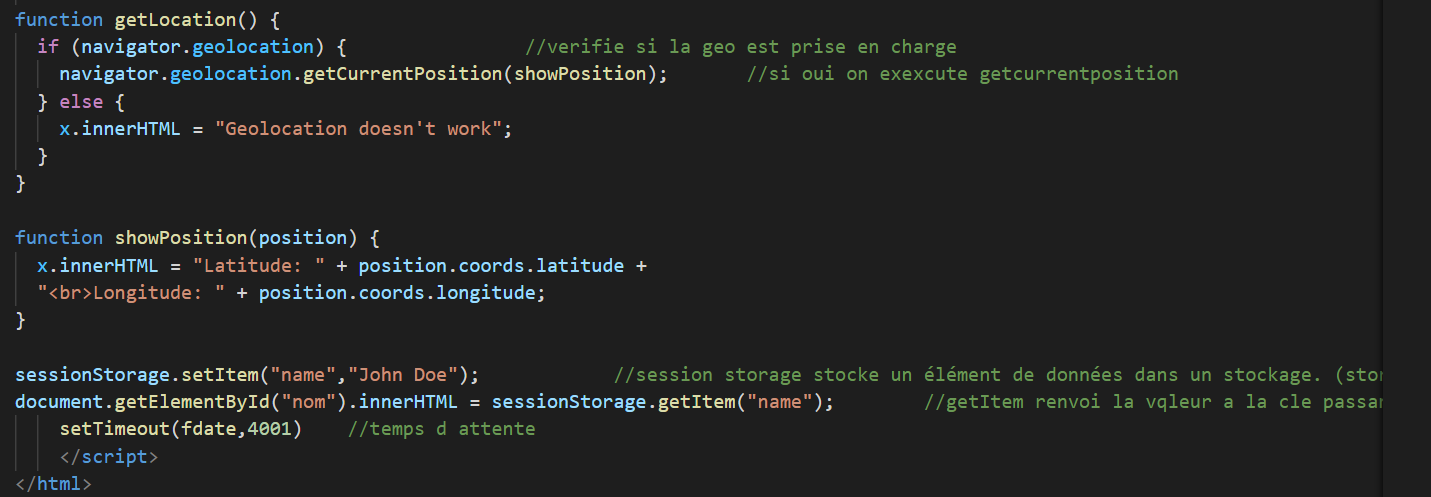
Ma première utilisation d’une API à été le DOM. L’API DOM(Document Object Modele) est utilisé pour interagir avec le navigateur. Il permet ainsi de créer des pages dynamiques en fonction des requêtes et des réponses envoyées par l’utilisateur. Cette interface de programmation a pour but d’interagir avec une page web, modifier cette page et d'interagir en JavaScript. Document est la principale méthode utlisé en JavaScript.Il permet d’envoyer un objet Element représentant l’id.



Voici un point de départ de mon code. Ici document.getElementById permet de récupérer une balise HTML créée précédemment. Je peux ainsi récupérer la valeur dans webSite puis à l’aide de innerHTML je peux afficher. Ici on affiche le nom rentrée dans la commande prompt permettant d’afficher une question pouvant être répondu directement dans une alerte(on y la valeur dans la variable name).

Ici dans cette partie du code on peut remarquer 2 fonctions. Ces deux fonctions sont appelées précédemment en HTML dans un type button. Ce bouton possède une méthode onclick ou est appelé les différentes fonctions( English; French). Ainsi lorsque l’on appuie sur un des boutons la langue change.

D'autres API tel que BOM(Browser Object Model) appartenant aux API externes ont été utilisés dans ce code tel que la localisation.

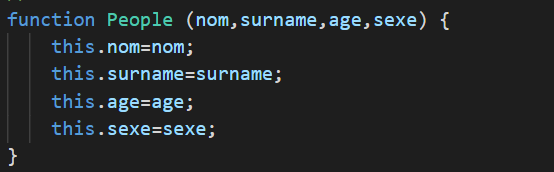


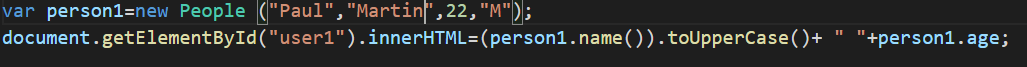
L’API localisation permet de renvoyer la latitude et la longitude de l’utilisateur.

navigator.geolocaliation permet de savoir si la localisation est possible sur le navigateur. L’utilisateur doit ainsi valider ou non sa permission d’utiliser sa position. Si la réponse est oui alors on peut montrer la localisation sur la page web.

JavaScript orienté objet :

Par la suite, j’ai réalisé des programmes en JavaScript orienté objet. Ce type de JavaScript ressemble étroitement aux langages Java, n’est pas sa fonctionnalité première. Après avoir réalisé de nombreux code en Java au cours de cette année de I1, le JavaScript orienté objet m’a paru facile. Ces 2 types de langages possèdent des constructeurs qui permettent de définir des objets et leurs propriétés. Leurs écritures et modes de fonctionnement se ressemblent beaucoup.





Par exemple, ici, je crée un constructeur nommé People. Je peux créer une nouvelle personne via la commande: new People. Ensuite j’affiche cette personne en appelant toujours mon id créé en HTML.

Mais des différences existent entre Java et JavaScript orienté objet.

Comme dit précédemment au fur et à mesure de mon avancement dans le JavaScript j’ai eu l’occasion de réaliser pluralité de codes. J’ai eu l’occasion de créer une calculette, qui me donnait le pgcd, la valeur la plus grande de mon calcul…

Maintenant que le langage JavaScript ai été assimilé, je me suis intéressé au langage Json

1. Le client serveur

Le langage Json est un langage qui permet l’échange de données entre le serveur et l’utilisateur.

Pour débuter il m’a fallu bien comprendre les notions de serveur et de client.

* La relation serveur-client

La relation client-serveur est très importante notamment dû à l’utilisation du JavaScript et de ses framework en Node.Js. En effet, lorsque l’on code en Node.js, un certain nombre de requêtes entre le serveur et le client existent. Dans les principaux codes que j’ai pu coder, la partie client était visible grâce à mon code HTML qui lui était en relation avec mon code Node.js.

La partie client est la partie que voit l’utilisateur lorsqu'il lance une page Web, c'est-à-dire l’interface. La partie serveur est la partie back-end.Contrairement au JavaScript, Node.js s’occupe de la partie Back-end.

Pour résumer:

-Le front end est la partie vue par le client. C’est avec lui que l'on peut interagir depuis votre navigateur( coder principalement grâce HTML,CSS, JavaScript)

-Back-end est la partie qui s’occupe de faire tourner le site internet. Il est invisible à l’utilisateur mais c'est ce qui donne vie au site web. Le Back-end est composé essentiellement d’un serveur pour l'hébergement web, d’une application qui tourne sur le serveur et aussi régulièrement d’une base de données ( coder principalement grâce Node.js, PHP, python, SQL…)

Un serveur informatique a pour utilité de fournir des services autour des données.

Il permet donc une multitude tâches que ce soit de fournir un accès à un site web, stocker des données en base de données ou encore faire tourner des applications.

Tous ces services sont accessibles via des requêtes des clients. L’utilisateur envoie une requête au serveur, le serveur réceptionne sa requête et envoie une réponse à l'utilisateur, donc la page web demandée. Cependant l’utilisateur peut aussi être un serveur, car un serveur peut demander une requête à un autre serveur.

* Le JSON

Le format de données Json (JavaScript object notation) peut s’écrire à la fois dans un fichier.json mais aussi dans une partie JavaScript d’un code. Le JSON se compose essentiellement de “texte”.

C’est à dire que ce format ne comporte pas:

* De fonction
* De boucle
* De Variables...

Le Json possède de nombreux avantage :

* il est facilement compréhensible pour les machines mais aussi pour les humains
* Il échange des données complexe entre les machines
* Prise en charge par de nombreux types de langages
* Syntaxe simple et légères

On dit que le JSON est un format simple car il possède seulement 2 types possibles d'objets .

Le JSON peut être soit un tableau ou soit un objet. Voici comment il fonctionne :

JSON

{

Valeur

String : Valeur

}

[

Object

Tableau

]

,

Lorsque l’on veut avoir plusieurs valeur on les sépare avec l’aide de virgule.

Le JSON possède différent type de donné qui peuvent être:

* Des chaînes de caractères
* Des entier ou décimaux
* La valeur null
* True ou False

Voici un code JSON intégré à une page web HTML:



Deux chaînes JSON ont été créées. La première qui est une simple chaîne JSON, et la deuxième qui est un tableau JSON.

Dans la première partie du code ma constante txt possède une chaîne JavaScript contenant du JSON. Elle possède donc comme entrée une accolade ainsi qu'en sortie. De plus on remarque que l’on peut appeler seulement une partie du JSON en nommant la variable nommée name dans le prénom. Dans cet exemple, je parse la chaîne JSON pour que cela devienne un objet JavaScript et ensuite pour pouvoir l’afficher via document.getEelementById.

La sortie nous donnera : Arthur, 21 en chaîne de caractère.

Pour utiliser ce format, il faut soit parser le JSON ou alors le stringifier.

La méthode parse : Elle permet de transformer une chaîne JSON en objet JavaScript

La méthode stringify : Elle permet de transformer un objet JavaScript en une chaîne JSON.

Dans la deuxième partie du code on remarque que l’on peut appeler une valeur d’un tableau grâce au crochet.

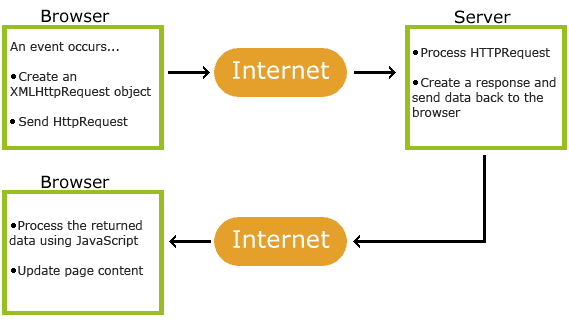
Préalablement parser ma réponse donne : myArr[3] donnera Malte

* L’AJAX

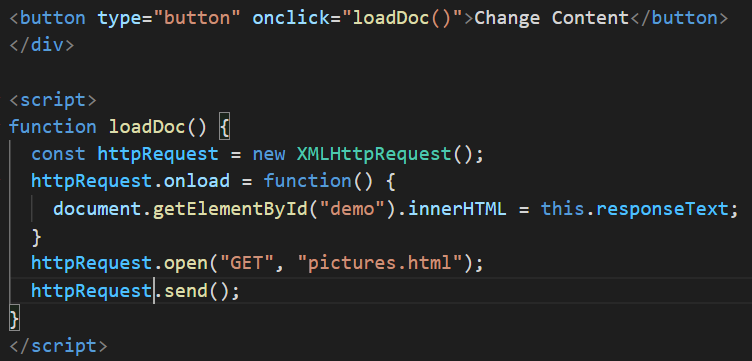
Comme dit précédemment le JavaScript est un langage permettant de créer des pages web dynamiques.

Ajax (Asynchronous JavaScript XML)

C’est une technique qui permet d’appeler une page en javascript pour pouvoir récupérer les informations et les afficher. Cette technique est utilisée pour envoyer et recevoir des données en arrière-plan sans recharger une page.



Voici comment fonctionne Ajax. L’utilisateur effectue des actions sur la page web. JavaScript récupère ces actions puis démarre une requête XMLHttpRequest( objet permettant d'interagir avec le serveur) vers le serveur. Après que le serveur ait récupéré ses requêtes, il crée une réponse et envoie les données au navigateur. Enfin le DOM récupère ces données et les traite de façon dynamique.



Comme dit précédemment, ajax s’utilise avec l’objet XMLHttpRequest. Ici on associe cet objet à une constante httpRequest. Lors de l'appui sur le bouton une méthode get est utilisée et mon fichier nommé pictures.html apparaît sans delay ni rechargement de page.

Néanmoins ajax ne peut pas être lancé seulement avec une page HTML. Pour lancer un code AJAX, un serveur utilisant http doit être créé.

Serveur

Un serveur web permet de stocker et publier des pages web. Il utilise le protocole Http(Hypertext Transfer Protocol) qui détermine comment la page est transmise du client au serveur. Http permet donc de créer une communication du client au serveur.

Pour ma part dans un premier temps, ce serveur a été créé à l’aide de VisualStudio code. J’ai téléchargé l’extension “live server” qui m’a permis de ne pas avoir à créer celui-ci.

Cette extension me créait un serveur au port 5500.



Protocole http

Nom du domaine : PORT

Chemin vers mon fichier

2. Le Node JS

Node.js est un environnement d'exécution javascript synchrone conçu pour créer des applications réseaux évolutives. Il se lance à l'aide du terminal grace a la commande node nomDuFichier

On se demande alors la différence entre asynchrone et synchrone.

1. Le synchrone et l'asynchrone

* synchrone

Un code Synchrone s'exécute de haut en bas, chaque événement commence quand l'événement précédent est terminé quel que soit la durée du processus. C'est-à-dire que chaque ligne est exécutée séquentiellement jusqu'à la dernière ligne de code.

Le javascript est un langage synchrone bloquant. Il ne peut donc faire qu'une seule chose à la fois. Néanmoins on peut intégrer des fonctions asynchrones dans le JavaScript.

* La notion de asynchrone

Un système est asynchrone est un système qui commence au début du code. Lorsqu’il atteint une partie asynchrone, il exécute cette fonction en même temps que le reste du code.

Cela permet d'exécuter plus d'opérations quel que soit le temps pris par un événement.

Le node.js est un système asynchrone non bloquant. De nombreuses tâches sont gérées en arrière-plan. Cela permet une rapidité d'exécution

En JavaScript nous pouvons créer de nombreuses fonctionnalités asynchrones. Cela est possible car notre serveur tourne grâce à node.js. Dans un serveur nous avons un moteur JavaScript pour Google permettant de lire du JavaScript. Cette interprèteur est codée en C++ comme Node.js. Grâce à eux on peut créer du code asynchrone.

* CallBack/ Promesses

Un callback en JavaScript est une fonction prenant pour argument une autre fonction. Un callback est donc une fonction que l'on fournit à une fonction asynchrone.Cette dernière est exécutée lorsque le résultat de la fonction asynchrone est prête.Un callback permet un meilleur contrôle sur le moment d'exécuter une fonction.

Les CallBacks sont utiles pour les fonctions courtes. Si des fonctions sont définies précédemment et doivent êtres appelé en callback il est préférable d’utiliser une promesse

Voici un exemple de fonction callback. Ici on remarque bien que l’on a une fonction callback en second paramètre de ma fonction. Ici le callback permet de retourner de nouveaux tableaux. Elle permet donc une infinité de possibilités pour réaliser une multiplication sur les valeurs du tableau.

Les fonctions callback s'utilisent et sont déjà intégrées dans de nombreuses fonctions JavaScript. Telle que le setTimeout, setInterval ...

Une autre méthode existe pour réaliser un code asynchrone en Java, elle se nomme promesse.

Une promesse est utilisée pour gérer le résultat asynchrone d'une opération.

En utilisant Promises, nous pouvons reporter l'exécution du bloc de code jusqu'à ce que la requête asynchrone soit terminée. De cette façon, d'autres opérations peuvent se poursuivre sans interruption

En conclusion une promesse est un objet retourné dans lequel on attache des callback. Cela est d'autant plus utile car nous n'avons pas à passer de callback a une fonction

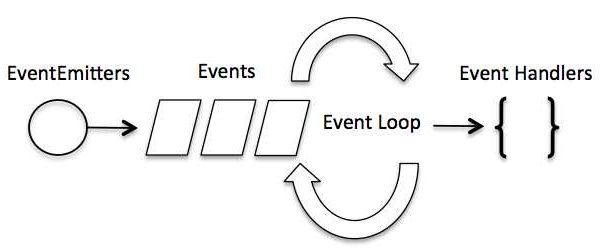
* Evénements

Node.js est un environnement single thread avec des boucles d'événements. Il permet de gérer un seul processus à la fois. Mais elle est utilisée pour mettre de côté des opérations longues et fastidieuses. L'événement exécute des processus définis pour chaque demande dans l’ordre.

Dans node.js tous les clients s'exécutent au sein du même thread.

Le modèle de traitement de Node JS est principalement basé sur le modèle d'événement JavaScript. Grâce à cela, Node.js peut gérer davantage de demandes concurrentes de clients. La boucle d'événements est au cœur du modèle de traitement Node.js

Etant donné que Node.js est asynchrone mais aussi monothread, node.js utilise des appels de fonction asynchrone pour avoir toujours une concurrence.



Voici comment fonctionne un événement. On envoie en premier une requête d'événement correspondant à la première tâche. Ensuite chaque requête est exécutée et est gardée en mémoire. La boucle d'événement se termine quand il n’y a plus de tâches. Cette boucle permet donc d’utiliser des callback et des promesses

3. Les modules Node.js

Comme dit précédemment node.js est un environnement. Il est composé de nombreux modules. Certains modules comme le module express ou encore http sont les plus connus. Mais d'autres peuvent être télécharger à l'aide de npm.

Npm est un gestionnaire de paquets qui comporte des milliers de modules. Chaque module doit etre télécharger sur le terminal via la commande: “npm install nomDuModule”

Npm est installé directement lors du téléchargement de node.js.

Dans cette partie je vais essentiellement parler dans un premier temps des modules les plus utilisé et les plus connu dans node.js

J’ai complété le contexte du stage. Mais tu dois plus détaillers

1. Modules http et fs

Un serveur web est créé à l’aide de node.js. Dans un premier temps, pour créer ce serveur, le module HTTP est la meilleure solution. Ce module est directement intégré à node.js qui permet donc de transférer des donnés au serveur via le protocole http.

Un module est ajoutable à un code via la ligne de code suivante : “ require “nom du module”



Ici après avoir ajouté notre module http, on crée le serveur via la requête http.createServer(). Celui-ci permet de créer un objet http serveur qui pourra ensuite écouter différent port. Dans ce code, mon serveur écoute au port 8081.

response.writeHead à pour but que à chaque appel de la fonction, donc à chaque requête entrante sur le serveur et envoi donc un état. Voici les principaux status http:

Quelques statuts HTTP :

- 200 : Requête OK

- 403 : Accès interdit

- 404 : Page (ou fichier) introuvable

- 500 : Erreur interne du serveur

Il faut réaliser la méthode writeHead avant d’envoyer la réponse au serveur (res.end).

Le module fs (file system) permet d'interagir et d’accéder aux fichiers. Ce module à l'instar de http module est directement installé avec node js. Beaucoup plus de fonctionnalités sont possibles avec ce module. Notamment l’ouverture d’un fichier avec sa lecture, la copie, la suppression, la création d’un fichier…

Dans le code précédent fs.readFile, comme son nom l’indique permet de lire un fichier (ici index.html)

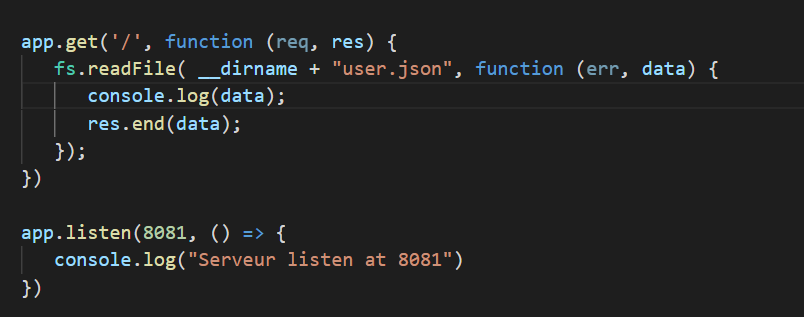
1. Module express

Le module express est le framework le plus populaire et le plus utilisé. Démarré en 2010 express n’est pas initialement installé avec node.js. Il est nécessaire de rentrer la commande : npm install express

Express est donc un framework permettant de développer facilement une application web. De nombreuses fonctionnalités sont disponibles via express. Il permet notamment de mettre en place des middlewares, permet de créer des requêtes poste et get, permet de créer un serveur web...

Express est le module que j’ai le plus utilisé au cours de stage.

Voici un des premier code utilisé pour me familiariser



Ce code crée une requête get. Celui-ci permet à l’aide du module fs d’afficher un document (ici un document json). De plus, il permet d’instancier un serveur web au port que l’on désire.

Ici ce code s’affichera donc dans l’url <http://127.0.0.1:8081/> et on pourra voir nos différents user du fichier user.json.

Méthode get et Post :

Get :

Généralement la méthode get est utilisée pour obtenir des données (même si elle peut en envoyer).

Dans cette méthode les données sont visibles par tous dans l’url.

Get permet de garder en données certaines tâches puis de revenir plus tard en retrouvant la page telle que l’on a laissée.

Post:

La méthode post est utilisée pour transférer des données du client au serveur.

Dans cette méthode les données que nous envoyons en tant qu’utilisateur ne sont pas affichées dans l’url mais dans la requête http. Elle permet donc une meilleure confidentialité que la méthode get. Généralement utilisé dans les formulaires, pour transmettre des informations aux utilisateurs.

* **Objectif final**

Mon projet s’est déroulé en deux temps. D’abord j’ai réalisé de nombreux codes ce qui m’ont permis de découvrir le javascript puis le node js, et notamment leur manipulation. Tout au long de la réalisation de mon stage j’ai pu mettre en pratique ma théorie. J’ai pu réaliser un exercice de code afin d’apprendre à manier le node js, ainsi qu’implémenter une API, implémenter Potsman…

1. La confection d’un répertoire de données

Pour la réalisation de la première partie de mon projet, j’ai utilisé un code qui utilise à la fois la notion de json mais aussi de node.js. Mon maître de stage m’a recommandé l’utilisation de ces derniers. Au cours de mes premiers temps de stage, j’ai eu l’occasion de m’exercer sur divers tutoriels (v.supra), ce qui m’a permis de confectionner mon répertoire de données.

Objectif du répertoire de données : Après avoir entré le nom de l’utilisateur au sein du programme :

* soit le nom de l’utilisateur est inscrit au sein de mon fichier json   
  alors ses informations personnelles sont renvoyés à l’utilisateur
* soit le nom de l’utilisateur n’est pas inscrit au sein de mon fichier json   
  alors je l’inscris dans ce dernier. Le fichier json est alors mis à jour.

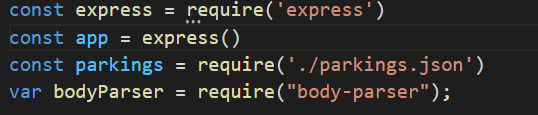
But du répertoire de données : Je devais donc rechercher un nom sur un formulaire html à partir de données json.

Réalisation du répertoire de données :

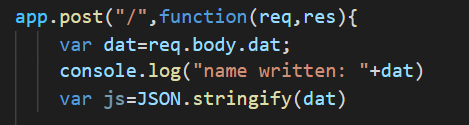
1. J’ai réalisé un code html. Puis j’ai introduit au sein de ce dernier une méthode form. Le code html permet à l’utilisateur d’inscrire son nom. Autrement dit, ce dernier peut faire une demande.



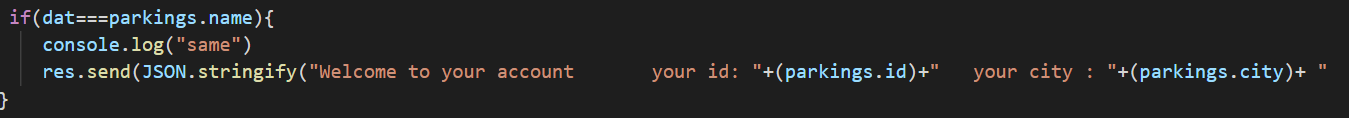
1. J'introduis dans ma méthode form (v.confra) une méthode post. Cette dernière me permet d’appeler la partie du code html réalisée précédemment au sein d’une requête post de mon code node.js.
2. Par la suite, j’ai réalisé ma partie concernant le node.js. Afin de réaliser ce code, j’ai utilisé différents modules mis à ma disposition.



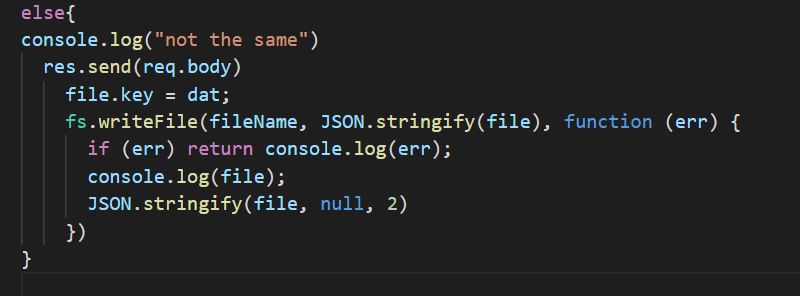
1. Le module Express (v.confra) à été utilisé. Le code parkings.json est le code en json qui comprend les données de l’utilisateur. Enfin, j’ai utilisé le body-parser. Body-parser est un second module. Il permet de récupérer et d’analyser le corps de requêtes entrant dans un middleware. Les middlewares sont des fonctions. Elles permettent d’accéder à des objets de type request et response. Elles permettent à la fois d’apporter des modifications aux objets mais aussi de terminer des cycles. Grâce à une variable nommée next(), si une fonction middleware ne se termine par il suffit d'appeler la variable next pour passer à la suivante.
2. Après avoir réalisé une méthode get pour afficher le formulaire. J’ai créé une méthode post.



1. Dans cette méthode post, je récupère la valeur donnée grâce à mon formulaire. Je l’inscrit dans ma console. On obtient alors, une valeur de type chaîne de caractère.
2. Par la suite, je détermine si le nom donné est inscrit dans la base de données des utilisateurs inscrit dans mon fichier parking.



1. Si le nom de l’utilisateur récupéré est inscrit au sein de ma base de données alors je transmet à ce dernier ses différentes informations personnelles.
2. Si le nom de l’utilisateur récupéré n’est pas inscrit au sein de ma base de données alors je suis la procédure expliquée ci-après.

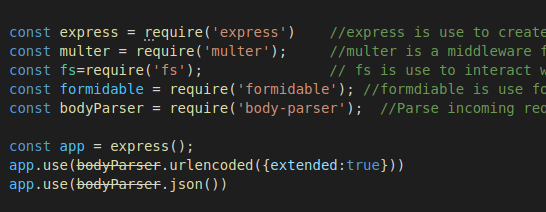


1. Dans le cadre de la deuxième hypothèse, précédemment citée, je commence par envoyer la valeur récupérée au sein de ma base de données,par mon formulaire, à l'utilisateur. Ensuite j’ajoute ce nouveau nom dans mon fichier JSON. Pour ce faire, j’utilise le module fs (v.confra). J'inscris alors le nom de l’utilisateur nouveau au sein du fichier par le biais de la méthode writeFile. Pour que je puisse ajouter la nouvelle clé au sein de mon fichier j’utilise la méthode stringify (ayant pour argument le nom donné par l’utilisateur). Enfin, le fichier JSON est mis à jour.
2. La confection d’un explorateur de fichier

Dans ce second code correspondant a mon principal projet de ce stage, mon but était de créer un explorateur de fichier. Dans celui-ci, je devais pouvoir supprimer un fichier, le renommer et enfin l'uploader.

Pour réaliser cela, il m'a fallu travailler sur linux. J’ai donc utilisé le logiciel Oracle virtualBox pour me créer un VM ou j’ai pu télécharger ubuntu. Ce travail sur linux est dû au fait que

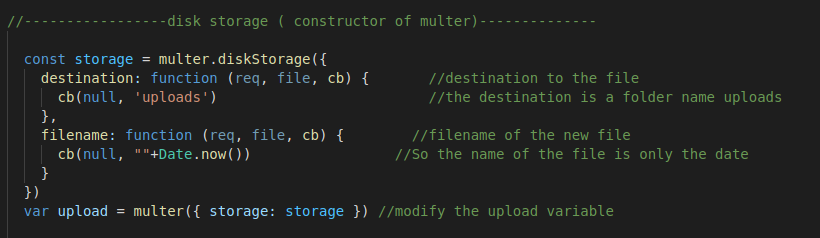
certains modules doivent être utilisés avec linux et non windows. Cela concerne notamment le module child process.



Voici les différents modules utilisés pour réaliser ce projet. Multer est un module appartenant a express. Néanmoins il a besoin d’être téléchargé préalablement grâce à npm. Multer permet la gestion de fichier. C’est un middleware principalement utilisé pour upload un fichier.

Formidable est lui principalement utilisé dans ce code pour récupérer des informations sur un fichier.

Ma première partie du code consiste à pouvoir upload un fichier JSON et l’afficher.On utilise donc le middleware multer



Pour utiliser Multer, il faut créer une sorte de constructeur. Ce constructeur permet d’instancier la destination et le nom du fichier que l'on veut télécharger. Les fonctions contenus dans ce constructeur possèdent trois paramètres. La requête, le fichier et cb. Ces fonctions retournent soit la destination ( ici uploads) ou alors le nom du fichier. Ici les noms des fichiers correspondront à la date du téléchargement de celui-ci.

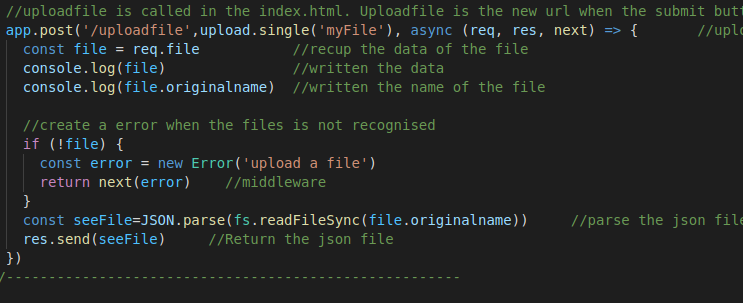
Ensuite il a fallu créer une méthode get. Cette méthode get permet de lire mon fichier html

Une méthode post est utilisée pour pouvoir upload un fichier. Ici ma destination sera uploadfile. Pour ma part je devais seulement réaliser le téléchargement d’un seul fichier. C’est pour cela que j’utilise upload.single.

Pour un téléchargement de fichier il faut utiliser une méthode asynchrone ( voir après). Ici on utilise une fonction asynchrone possédant la requête, le résultat, et enfin le middleware next( voir après)

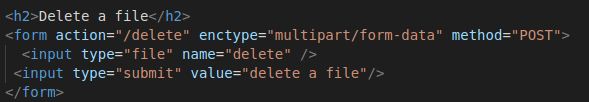
Dans un premier temps via la requete req.file, je récupère mon fichier puis j’obtient son nom.

Je réalise ensuite une boucle if puis else. Cette boucle a pour utilité si une erreur du fichier est trouvée. Si pas d’erreur se trouve alors je parse mon fichier JSON( voir après pour parse) puis je l’affiche.

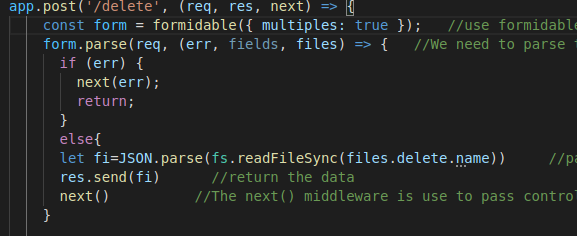


Ensuite, je me suis consacré à réaliser ma partie de suppression de fichier et de son affichage.

Pour pouvoir supprimer un fichier, je dois obtenir le nom du fichier. Pour ce faire il m’a fallu réalisé une seconde methode form, correspondant à ma première :



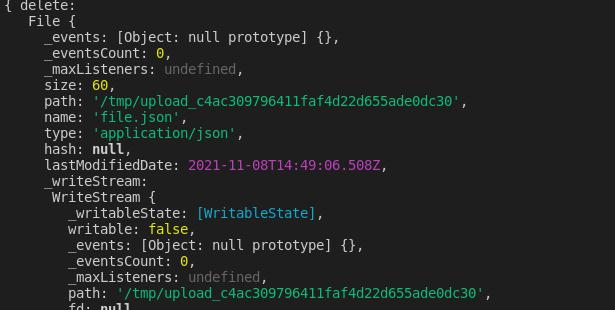
Puis ensuite le code JavaScript :



Dans cette partie j’ai eu de nombreuses difficultés. En effet après avoir essayer d’obtenir le nom du fichier avec le module Multer, après de nombreux essais j’ai découvert que l’on peut obtenir le nom du fichier avec multer seulement si on upload un fichier.

Après avoir regardé dans la documentation de l’API express, c’est le module formidable qui permet cela.

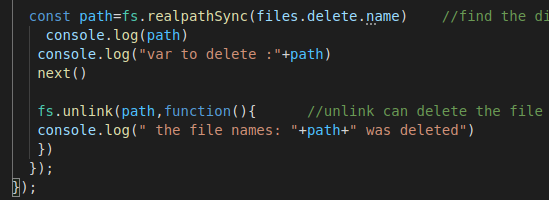
Après avoir créé une variable form permettant d’instancier formidable, on parse les données du fichier se trouvant dans ma variable files.Nous obtenons ceci:



Dans le résultat de files, nous pouvons obtenir une multitude de donnés dont bien sur le name. Ici je désire donc supprimer le fichier nommé file.json.

Pour appeler ce fichier dans mon code il me suffit d’appeler donc files puis delete correspondant au nom de mon formulaire en fin le name. Cela est possible car grâce à la méthode parse qui me permet de passer d’un format textuel à un format JSON.

Ensuite après avoir afficher le fichier, je dois donc supprimer le fichier.



Pour cela je recherche le chemin d’accès de mon fichier grâce a la méthode realPathSync.

Comme cette méthode ne se termine pas, le middleware next est nécessaire.

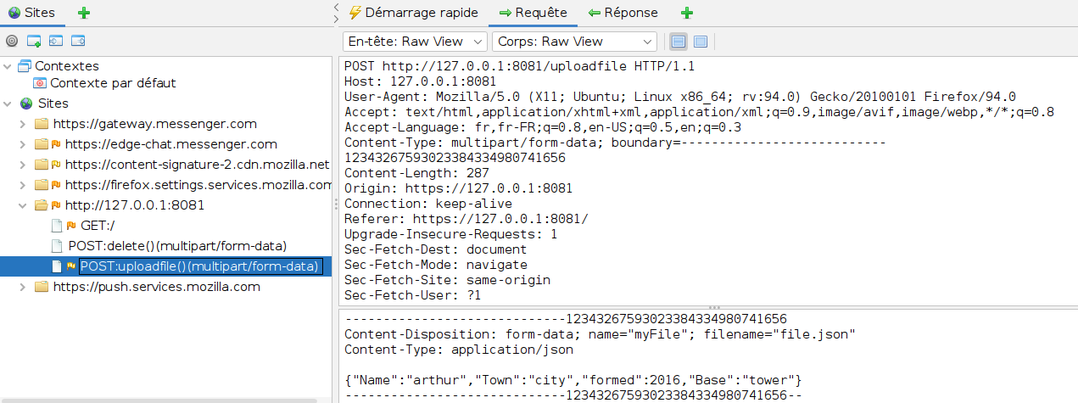
Le code peut donc passer à la fonction suivante, fs.unlink est utilisé et permet de pouvoir supprimer un fichier dont on connaît le chemin.

Partie renommage

partie wasp zap

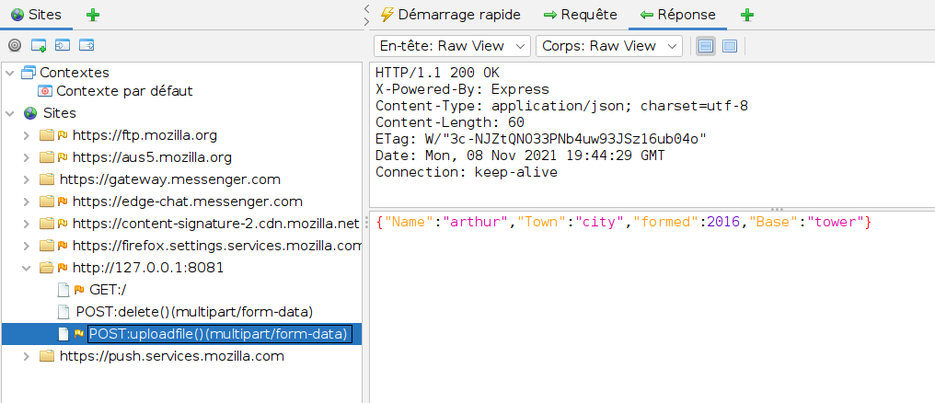
Pendant la réalisation de ce code j’ai régulièrement utilisé owasp zap. J’ai ainsi pu identifier et vérifier les différentes requêtes post et get. Pour réaliser cette tâche, j’utilise l’outil Manual Explorer. Celui-ci me permet de lancer le navigateur de mon programme ( il est nécessaire de d’abord lancer son serveur node.js puis de récupérer l’url et de l'inscrire dans OwaspZap). A l’aide du serveur proxy de Zap ainsi que du logiciel firefox ( ou des modifications de proxy sont aussi nécessaire dans les paramètres), on peut lancer le navigateur de OwaspZap.

Requete envoyé au serveur par la méthode post d’uploadFile



Dans la requête de POST, on remarque différentes informations. On peut voir que la requête post me renvoie mon url, l’univers ou mon code est chargé. On peut voir le contenu envoyé, la connexion qui s'effectue avec keep-alive(Un keepalive est un message envoyé par un appareil à un autre pour vérifier que le lien entre les deux est actif)...

Réponse envoyé au serveur pour la requete post d ‘uploadFile

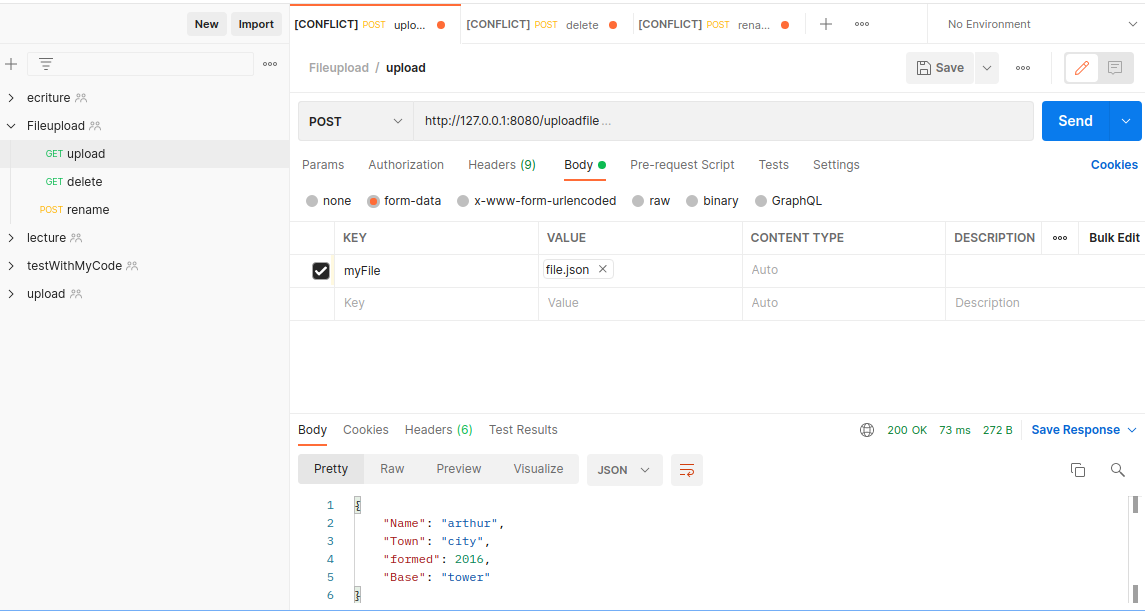


Dans le corps fr la réponse, on peut voir le document JSON envoyé et donc retourner par ma méthode POST. Dans la première partie ont peut voir qu’une requête http est utilisée. La valeur 200 est mon statut qui prouve que les requêtes se sont bien déroulées. Enfin la date de la requête.

partie Postman

Postman est un logiciel permettant de tester des requêtes sans nécessairement créer de fichier HTML.

Après avoir modifié quelque peu mon code ( suppression de la méthode get qui me renvoyait mon document html …)



Pour utiliser Postman il suffit de rentrer l’url, de sélectionner la méthode (post ou get) et enfin ici je veux envoyer un document donc j’utilise la partie form data du logiciel.Enfin il suffit d’envoyer au serveur et on peut voir la réponse de la méthode post.

* **Les indicateurs de réussite**

Pour réaliser à bien ce stage, je voyais régulièrement mon maître de stage. Je le voyais tous les lundi matin à 9h30 et soit le mercredi soir ou le jeudi matin selon ses disponibilités.

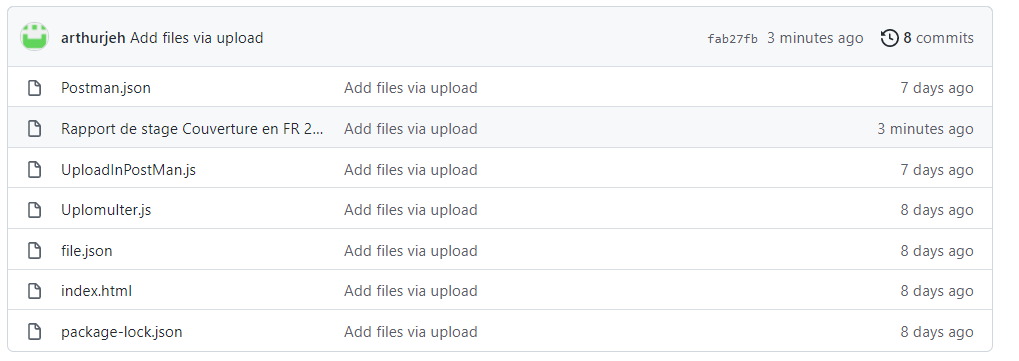
Lors de ses réunions d’une quarantaine de minutes, nous discutions de ce que j’avais réalisé, de mes différentes questions et des travaux que j’avais à faire pour la prochaine réunion.

Avant chaque réunion je devais mettre mon code sur le site internet github. Lors de la réunion j’affichais ces codes réalisés pendant les 2-3 jours sur une télévision permettant de suivre mon évolution.

J’ai utilisé 2 github :

Le premier pour les exercices : Le second pour mon explorateur de fichier :

https://github.com/arthurjeh/JSintersnhip https://github.com/arthurjeh/Web-file-explorer

Une image contenant table

Description générée automatiquement

Mon maître de stage me donnait les consignes à l’oral. Ainsi lorsque je n’étais pas sur des consignes ou alors que j’avais une questions je pouvais me rendre dans son bureau non loin de ma salle.

Sans trop de difficultés ces différents exercices ont été réalisés. C’est différents codes m’ont été très utiles pour l’apprentissage, et la réalisation d’un code plus volumineux que m’a demandé de réaliser mon maître de stage. Cet exercice est un jeu connu du grand public. Mon code JavaScript doit choisir un nombre aléatoire, et le joueur doit trouver ce chiffre en un minimum de coups.

* **Les contraintes**

J’ai pu réaliser une multitude de différents codes qui m'ont permis d'apprendre énormément de différentes librairies mais aussi de langages.

Le premier mois a été pour moi la découverte de l’environnement que ce soit les langages utilisés ou encore la compréhension et l’utilité des librairies…

Au départ, ce code de nature facile m’a posé de nombreuses difficultés. En effet, il m’à été difficile de commencer avec un code vide et de réaliser des boucles. J’ai essayé de réaliser différentes boucles mais sans succès. J’ai donc réalisé une boucle do while pour ce faire. De plus j’ai créé des alertes pour demander le nombre et aussi d'afficher dans la console si celui-ci est trop petit ou trop grand. Mon maître de stage m'avait donné 1 jour pour réaliser ce code.

J’ai dû donc réaliser ce code en peu de temps vu mon niveau sur ce langage au départ. Mais ce code m’a été grandement bénéfique et m'a permis de comprendre énormément de fonctions du javaScript.

TROISIÈME PARTIE : RAPPORT PERSONNEL

La réalisation de ce stage m’a apporté de nombreuses connaissances et apports personnels notamment au regard du plan professionnel (1), mais aussi sur le plan personnel (2). J’ai pu ainsi étoffer mon futur projet professionnel (3).

1. L'approfondissement de mes connaissances scolaires
2. Mes apports personnels au regard du plan professionnel
3. Mes apports personnels au regard du plan personnel
4. Un lien avec mon futur projet profesionnel

Annonce de plan :

* Apports personnels confrontés à l’expérience à l’université
* Résumé du PPP

Ce stage m’a apporté énormément de connaissance à la fois sur le plan professionnel mais aussi sur le plan personnel.

Ce stage à l'université de Malte m’a appris énormément sur l'informatique.

Passer la majeure partie du temps à l'université m’a permis de me rendre compte du fonctionnement universitaire et de ses dessous.

Au fur et à mesure du temps, côtoyer quelques personnes à l’université, côtoyer mon maître de stage régulièrement m’a permis de pouvoir acquérir de nombreuses connaissances en peu de temps.

Au cours de ce stage technique, mon maître de stage m’a laissé beaucoup d’autonomie. Cela à attisé ma curiosité de découvrir de nouvelles notions ainsi que d’apprendre et de les partager avec mon maître de stage.

Le fait que le node.js est récent(2009) et qu’il est en constante évolution notamment grâce aux nouveaux modules qui arrivent sur le marché me conforte dans l’idée que le domaine de l’informatique n’est pas répétitif. Ce dernier est un des adjectif qui me conforte dans l’idée de poursuivre dans l'informatique plutôt que dans l'électronique.

Les technologies se développent très rapidement, elles sont de plus en plus nombreuses et pointilleuses. Il faut donc continuellement se former à ces nouveaux moyens. De nature curieux, j’aimerai les découvrir et les appréhender.

De plus, au cours de ce stage j’ai notamment appris à hiérarchiser mon travail, faire des synthèses me permettant d’aller à l’essentiel quand il le faut mais aussi développer mon argumentaire lorsque cela est nécessaire.

Enfin, la parfaite maîtrise de la langue anglaise est essentielle. Il faut pouvoir comprendre, se faire comprendre à l’oral mais aussi à l’écrit. La langue est internationale.Mon maître de stage, me donnais toutes les consignes à l’oral, ainsi que m’expliquer des notions… Cela m’a permis de perfectionner mes connaissances et la maîtrise de la langue anglaise, dans un langage courant mais aussi spécifique. En effet, je me suis approprié de nouveaux mots informatiques techniques et particuliers du domaine.

CONCLUSION

Lors de ces 3 mois de stages, j’ai pu acquérir de nombreuses connaissances en informatique. Que se soit en JavaScript mais aussi grandement en JavaScript dans le contexte de node.js.

Cela m’a permis d’évoluer à la fois dans mon cursus scolaire mais aussi sur le plan personnel.

ANNEXES