Bourezak
Yacine

Dossier personnel et professionnel de fin de formation

Référent : Rafik Lachaal

Table des matières

Introduction

La formation de Développeur(se) Data vise à intégrer le monde de la donnée. Depuis quelques années maintenant les données sont une source d'information de masse, celles-ci peuvent être récupérées (open data), traitées, optimisées et mises en valeurs (data visualisation).
Les données sont actuellement dans tous les domaines qu'on connaît sport, santé, administration, commerce, les données ouvertes peuvent être publiques ou privés, elles peuvent être définies sous trois angles :

La **disponibilité et l'accès :** Les données doivent être disponibles pour un coût raisonnable de reproduction, de préférence téléchargeables.

La réutilisation et la redistribution : Les données doivent être disponibles sous une licence autorisant la réutilisation et la redistribution incluant le croisement avec d'autres ensembles de données

L'universalité : un accès non restrictif, tout le monde, sans aucune discrimination liée à l'activité pratiquée, doit être en mesure d'utiliser la data mise à disposition. Pour exemple, des restrictions 'non-commercial' qui empêcheraient les utilisations de ce type ou des restrictions d'utilisation à certaines fins ne sont pas autorisés.

Il faut savoir que la formation de Développeur(se) Data reposait autour d'une méthodologie d'apprentissage, celle-ci est la **pédagogie active**. La pédagogie active, a pour objectif de rendre l'apprenant acteur de ses apprentissages afin qu'il construise ses savoirs à travers des situations de recherche, de documentation, de support et de pratique.

Pour ce qui est de la planification des projets nous avons utilisé une méthode essentielle de gestion de projet qui est la méthode agile. Celle-ci part du principe que de spécifier et planifier dans les détails l'intégralité d'un produit avant de le développer (approche prédictive) est contre-productif. La méthode Agile recommande de se fixer des objectifs à court terme. Le projet est donc divisé en plusieurs sous-objectifs. Une fois l'objectif atteint, on passe au suivant jusqu'à l'accomplissement de l'objectif final.

Au bout du cheminement de cette formation, j'ai été amené à travailler sur deux projets principaux.
Le premier étant mon projet fil rouge (projet de fin de formation) et l'autre est mon projet de stage. Ce dernier, à cause de la crise sanitaire que nous avons vécue a été arrêté, cependant il me reste une bonne petite partie à vous présenter.

Projet fil rouge :
L'API (Application Programming Interface) Interventions incendies et secours est une interface web.

Illustration 1 : Rendu d'affichage de l'API interventions incendies et secours, elle est composée d'une chaîne déroulante permettant de choisir la catégorie d'incendies, et une d'une chaîne déroulante permettant de choisir l'année.

Projet de stage au sein du Consulat général du Sénégal à Marseille :
Application de suivi des demandes d'actes d'état-civil

Illustration 2 : Aperçu du site web du consulat général du Sénégal à Marseille

I) Première partie ; Projet fil-rouge :

1) Cas d'étude : API Interventions incendies et secours

L'incendie n'arrive malheureusement pas qu'aux autres : dans ses campagnes de communication, le ministère de l'Intérieur rappelle qu'il se produit plus de 250 000 incendies domestiques chaque année en France. Ce nombre impressionnant signifie qu'**un incendie se déclare en moyenne toutes les deux minutes**, et qu'un français sur trois risque d'être touché par un incendie. À l'heure du détecteur de fumée obligatoire, il y a un petit horizon des statistiques à connaître concernant les incendies.

Chaque année, on estime qu'environ 250 000 **incendies domestiques sont déclarés** aux assurances. Tous ne sont heureusement pas dramatiques, mais il convient tout de même noter que ce chiffre a **doublé en vingt ans**. À quoi attribuer cette hausse ? Un incendie domestique sur quatre est dû à une installation électrique défectueuse ou non conforme aux normes en vigueur.

Les pompiers rappellent régulièrement que la réactivité en cas de sinistre est essentielle pour limiter le préjudice. Et pour cause : il ne faut pas plus de trois minutes à une pièce en proie aux flammes pour voir sa température ambiante atteindre les 600 °C. Pour bien comprendre la vitesse à laquelle se propage un incendie, dites-vous qu'il faut :

- Un verre d'eau pour éteindre le départ du feu, lors de la première minute ;
- Un seau d'eau une fois la deuxième minute atteinte ;
- Une citerne à partir de la troisième minute…

Après réflexion de ce cas d'étude j'ai choisi de répondre à un besoin de visualisation de données, sur l'évolution des incendies en France sur trois ans. En effet, sur ce type de donnée, je n'ai pas trouvé de visualisation pertinente mettant en évidence l'évolution des incendies en France en fonction des départements. Pour ce faire j'ai choisi comme outils Flask afin de créer une API web simple.

L'API Interventions incendie et secours a été conçu pour avoir une vision claire de l'évolution du nombre d'incendies en France par départements. L'API mettra en valeurs le nombre d'incendies en fonction des départements et des catégories de feux dans l'hexagone. Elle donnera un rendu graphique et cartographique représentant les incendies sous différentes catégories de feux.

2) La traduction technique et choix technique du projet

a) SGBDR, MySQL, SQL, Workbench

J'ai choisi de modéliser une base de données pour accueillir les données concernant les incendies en utilisant le SGBD (voir page des sigles) MySQL.

À savoir qu'un SGBD est un logiciel qui permet de stocker des informations dans une base de données. Un tel système permet de lire, écrire, modifier, trier, transformer ou même imprimer les données qui sont contenues dans la base de données.

Dans mon projet j'ai choisi d'utiliser un SGBDR (Système de gestion de bases de données relationnelles) qui est un type particulier de SGBD, l'information est organisée dans des tableaux à deux dimensions appelés *tables*. Selon ce modèle *relationnel*, une base de données consiste en une

ou plusieurs relations. Les lignes de ces relations sont appelées des n-uplets ou enregistrements. Les colonnes sont appelées des *attributs*. C'est ce que l'on nomme le modèle Entité-Association.

Les logiciels qui permettent de créer, utiliser et maintenir des bases de données relationnelles sont nombreux :

- phpMyAdmin
- Adminer
- SQLectron
- MySQL WorkBench (IDE choisi)

À savoir que pratiquement tous les systèmes relationnels utilisent le langage SQL pour interroger les bases de données. Ce langage permet de demander des opérations telles que l'intersection, la sélection et la jointure.

J'ai choisi d'utiliser MySQL Workbench comme interface graphique. Le client graphique est une application de base de données qui permet d'avoir une interface graphique ergonomique (accessible).

MySQL Workbench (anciennement *MySQL administrator*) est un logiciel de gestion et d'administration de bases de données MySQL créé en 2004. Via une interface graphique intuitive, il permet, entre autres, de créer, modifier ou supprimer des tables, des comptes utilisateurs et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données. Pour ce faire, il doit être connecté à un serveur MySQL.

L'interface permet de réaliser un modèle de base de données, incluant les interconnexions entre les tables, et de générer le code SQL qui découle du modèle. Par ailleurs, une connexion peut être réalisée avec une instance MySQL afin d'importer un modèle sur un environnement de développement ou de production. De même, l'opération inverse est également possible, afin de déduire la modélisation d'une table à partir d'une instance réelle d'une base de données MySQL.

Parmi l'étendu des fonctionnalités, il est aussi possible de citer les points suivants :

- Sauvegarde et restauration de tables
- Tableau de bord interactif pour évaluer la performance des applications tournant sur MySQL
- Export des bases de Microsoft SQL Server vers MySQL (Pas utilisé au cours de la formation)

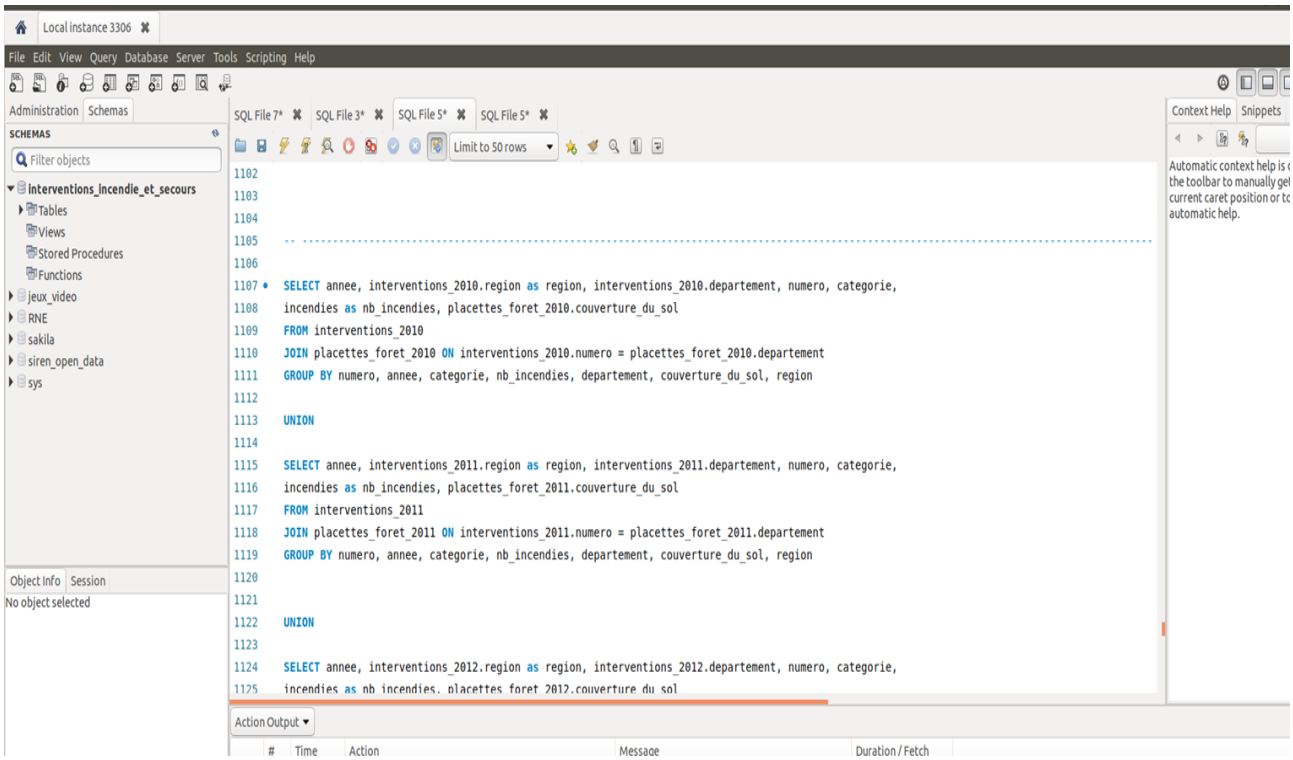


Illustration 3 : Interface graphique Workbench montrant du traitement de données par une requête.

L'utilisation de l'outil **MySQL Workbench** apporte de nombreux **avantages** : regrouper les différentes opérations de telle manière qu'il est possible avec un seul outil de gérer à la fois les bases de données elles-mêmes, mais également les éléments de configuration du serveur (taille maximum du tampon, nombre de connexions, etc.) ce qui n'est pas possible de faire par exemple avec phpMyAdmin.

L'inconvénient est qu'il n'est pas encore très stable, on peut probablement rencontrer des beugues en l'utilisant de façon intensive.

b) Jupyter Notebook, Python et ses librairies

Pour l'importation des données en bases, j'ai réalisé un programme Python, j'ai donc utilisé Python comme langage de programmation qui est un langage orienté objet. Python est un langage qui est également très utilisé dans la data visualisation (de nombreuses librairies python permettent d'analyser et de visualiser des données par exemple : plotly, pandas, numpy). Concernant l'outil de rédaction de mon code j'ai utilisé JupyterNotebook.

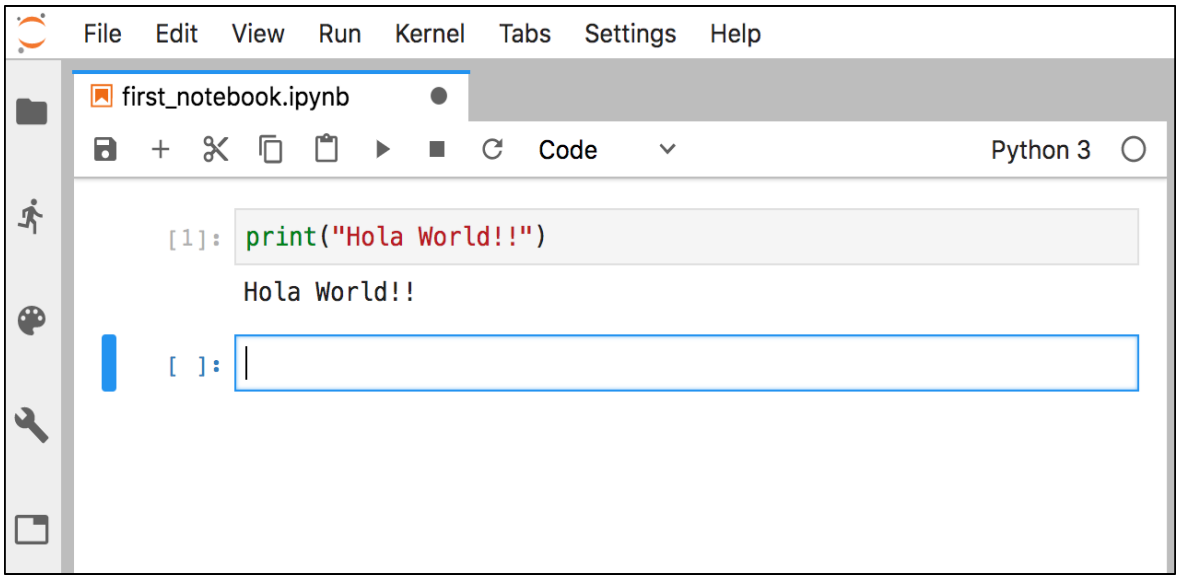


Illustration 4 : Exécution d'une fonction Python sur un Notebook Jupyter

Le notebook-Jupyter est une application Web open source qui vous permet de créer et de partager des documents contenant du code natif, des visualisations et du texte narratif. Les utilisations incluent : le nettoyage et la transformation des données, la simulation numérique, la modélisation statistique, la visualisation des données, l'apprentissage automatique et bien plus encore.

Illustration 5 : Logo Jupyter

Jupyter permet de réaliser des notebooks, c'est-à-dire des programmes contenant à la fois du texte en markdown et du code en Julia, Python, R… Ces notebooks sont utilisés en science des données pour explorer et analyser des données.

Les notebooks ne sont pas autonomes. C'est le principal **inconvénient** à l'utilisation de **Jupyter**. Ses **notebooks** nécessitent un runtime (Environnement d'exécution), de même que les bibliothèques que l'on souhaite utiliser. Il existe quelques stratégies pour créer des cahiers **Jupyter** autonomes, mais aucun d'eux n'est officiellement supporté.

c) Flask

Flask est un micro Framework de développement web écrit en python. Ce qui le différencie d'autres solutions web, Flask est livré avec le strict minimum à avoir :

- Un moteur de Template (Jinja 2).
- Un moteur web de développement (Werkzeug) (C'est un serveur web WSGI écrit en python) Web Server Gateway Interface.
- Un système de distribution de requête compatible REST (dit RESTful).
- Un support de débogage intégré au serveur web.
- Un micro Framework doté d'une très grande flexibilité.
- Une très bonne documentation.

J'ai choisi Flask pour mon projet parce qu'il utilise python et que cela permet de garder une certaine homogénéité étant donné que le traitement préalable de mes données est réalisé avec la librairie Pandas.

Inconvénient :

- Nativement moins fournit que Django pour réaliser un développement de niveau professionnel (Authentification, gestion sécurisé, administration).

Avantages :

- Permet de produire les 1re pages rapidement.
- Plus léger -> optimisation de ressource.
- Nombreuses extensions et populaire.

L'Anatomie d'un projet Flask :

• L'élément le plus important d'un projet Flask est la structure des répertoires utilisée pour stocker les différents éléments du projet.
• Ci-contre la structure de base du projet Flask pour le projet « mon-projet » avec quelques éléments complémentaires pour illustrer la mise en place.

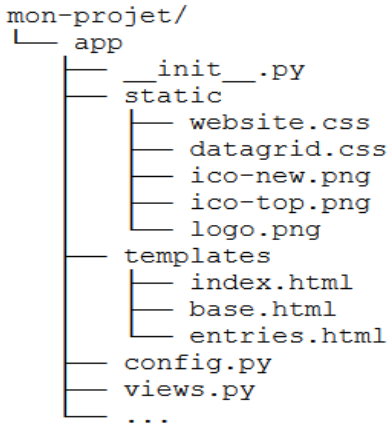


Illustration 6 : Anatomie type d'un projet Flask

II. Seconde partie – Projet fil-rouge : Mise en œuvre du projet

1) Gestion de projet :

Au début de la création de mon API, j'ai été amené à réaliser un plan méthodique. Pour ce faire j'ai dû établir la problématique de mon projet :

Est-ce que la visualisation des données du nombre d'incendies par départements et par catégorie de feu suffisent à prévenir les risques d'incendies ?

Au cours de ma formation j'ai dû apprendre à entreprendre mes projets en méthode agile. Concernant mon projet qui est un projet fictif, j'ai été à la fois le Product Owner (responsable de la **définition** et de la conception d'un produit), le Scrum Master mais également le Backlog Scrum.
Pour commencer j'ai réalisé un Trello qui est un outil de gestion de projet en ligne utilisé en entreprise qui permet de réaliser des plannings interactifs. De ce fait, j'ai mis en place le planning d'évolution de mon projet sous quatre mois avec trois étiquettes pour chaque tache à faire accompagner de pastille verte, jaune et rouge :

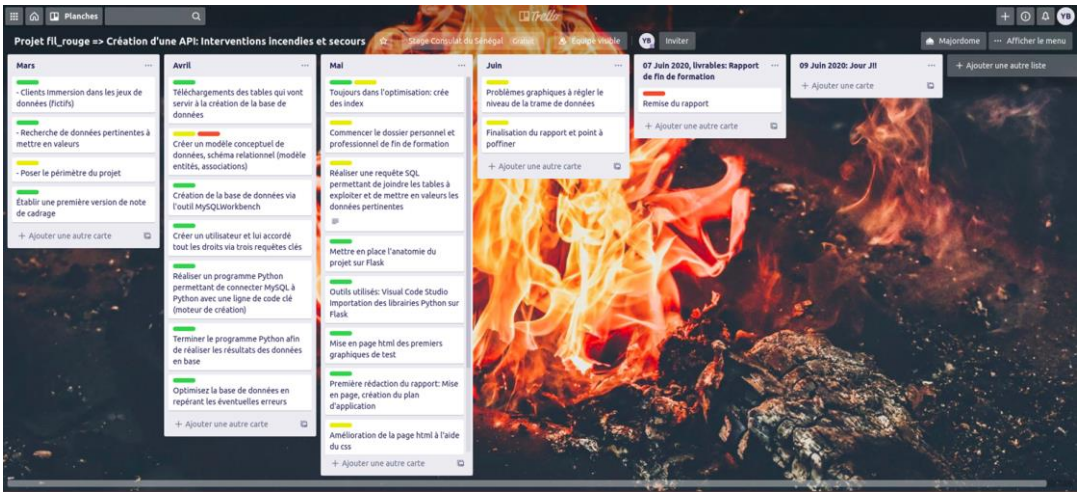


*Illustration 7 : Image de gestion de mon projet avec Trello : **Fait**, **En cours**, **Pas fait***

Parallèlement à la planification du projet via Trello, j'ai mis en place une note de cadrage. Celle-ci permettant d'avoir une vision claire sur les objectifs de mon API, sur les besoins auxquels elle répondra, les bénéfices dont on pourra en tirer, les points à surveiller et enfin les technologies utilisées. La nécessité de cadrer son projet est importante afin d'éviter toute digression, en revanche il faut savoir que cela n'empêche en rien l'itération d'autre demande ou de fonctionnalité que le client pourrait réclamer.

Date :28 mars 2020
Équipe : Yacine Bourezak

Référent : Rafik Lachaal
Version : 0.0.1

Note de cadrage – Projet fil rouge – Intervention incendies et secours

Projet : Création d'une API via la base de données Interventions incendies et secours
Objectifs du projet **:**

- *Création d'une base de données relationnelles, importations des données en base*
- *Construire un modèle conceptuel de données en utilisant la méthode Merise permettant de mettre en relation les différentes tables à disposition.*
- *Mettre en évidence des données d'incendies dans une perspective démonstrative à l'aide de graphique (Data Visualisation).*
- *Cette API aura pour but de mettre en avant le nombre d'incendies par catégories de feu et par départements sur trois années spécifiques à l'aide de données claires et structuré utilisant des couleurs diverses.*
- *Périmètre cartographique de démonstration : France.*
- *Elle permettra un rendu graphique et cartographique significatifs sur toute la France*
- *Exemple :*
  -Recenser un grand nombre de données d'interventions de secours incendies, faire des jointures concrète, qui permettront par la suite de manipuler le jeu de données en fonction de l'événement à montrer.

Besoin **:**

- *Cibler l'intensité du nombre d'incendies par catégorie de feu en fonctions des départements.*
- *Avoir une vision claire de l'évolution du nombre des différentes catégories d'incendies par départements.*

Bénéfices **:**

- *Établir un meilleur plan d'action de secours à la suite des observations claires de l'évolution des incendies sur toute la France.*
- *Renforcer les services de secours dans les régions et les départements les plus touchés.*
- *Déboucher sur une étude plus avancée, si le temps le permet*

Itération possible en fonction du délai des livrables :

- *Établir une étude plus poussée*
- *Mettre en évidence une relation de cause à effet entre le nombre d'incendies et les températures*
- *Mettre en évidence une relation de cause à effet entre le nombre d'incendies et les données forestières*

Points à surveiller **:**

- *Avoir un rendu graphique et cartographique claire et lisible.*

Technologie du client :

- *Système d'exploitation : Linux 18.04 64 bits.*
- *MySQL Workbench.*
- *Anaconda.*
- *JupyterNotebook.*
- *Flask*

A propos du Jury Simplon Marseille **:**

- *Adresse : 20 Boulevard Rémusat 13013 Marseille.*
- *Responsable : Maître de Jury.*

2) Réalisation de projet :

Recensement des données :

Pour le recensement des données j'ai tout d'abord recherché des données d'incendies sur un site appartenant à l'état proposant des données ouvertes (data.gouv.fr). J'ai pu trouver des données d'interventions réalisées par les services d'incendie et de secours, les données étaient ordonnées sur plusieurs années allant de 2010 à 2018.

Illustration 8 : Plateforme du site Data. Gouv, montrant les fichiers d'interventions

Pour avoir une base de données optimale j'ai également recensé des données de températures, récupérées sur le site de data.Gouv. Celles-ci rassemblent les températures moyennes dans l'hexagone au niveau régional et départementale. Dans mon cas d'étude, j'ai choisi de travailler avec des données plutôt départementales en fonction du nombre d'incendies, le recensement de ses données me permettra de mettre en place une relation de cause à effet entre le nombre d'incendies par catégories de feu en fonction des températures en France.

Illustration 9 : Plateforme du site data.gouv.fr montrant les données de température

Organisation technique du projet

I. *Voir Annexe partie sur MySQL*

Après avoir téléchargé la totalité des données qui m'intéressait en format csv en local, j'ai pu passer à la conception de ma base de données en utilisant MySQL Workbench. J'ai choisi de nommer ma base de données « Interventions incendies et secours ». Une fois la création de la base de données réalisée, j'ai pu me pencher sur la création des tables destinée à accueillir les fichiers contenant les données. Concernant les tables interventions, j'ai finalement choisi d'utiliser seulement les tables d'interventions de 2016 à 2018.
Les tables s'organisent ainsi :

- Interventions
- Placettes forêt
- Température-quotidienne-départementale

À la suite de la création de mes tables ainsi que des colonnes les comportant, j'ai choisi de travailler avec la méthode MERISE (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise) pour l'agencement de mes données. Celle-ci est une méthode d'analyse de conception et de gestion de projet informatiques. Le principe de cette méthode repose sur la séparation des données et des traitements. J'ai donc commencé à réaliser mon modèle conceptuel de données. Le **modèle conceptuel** des **données** (MCD) a pour but d'écrire de façon formelle les **données** qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des **données**, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités et d'association entre les tables.
À la suite, j'ai créé un utilisateur nommé « yacine-fil-rouge » et je lui ai accordé tous les privilèges à l'aide d'une requête spécifique.

II. *Voir annexe partie sur l'importation des tables*

Une fois mes tables créées et mon modèle conceptuel fait, j'ai pu me tourner vers l'importation des données en base. Pour ce faire, j'ai utilisé JupyterNotebook, à l'aide d'un programme Python, j'ai pu importer les données qui allaient alimenter mes tables. Celui-ci comporte tout d'abord l'import de bibliothèques essentielles pour la bonne marche des lignes de code le composant :

- Pandas (permet de manipuler des tableaux de données avec des étiquettes de variables : colonnes, lignes).
- Time (donne accès au temps et aux conversions).
- Create_engine de la bibliothèque SQL Alchemy (permet une connexion entre MySQL et Python).

Le programme comporte une fonction chargement, à qui on va fournir une URL, depuis laquelle le programme va télécharger le fichier csv et aller l'inscrire les tables de la base de données préalablement créée avec un script SQL.

L'intérêt d'avoir créé la fonction chargement m'a permis, de répéter les instructions pour chaque fichier csv à inscrire dans la base de données. Une explication plus détaillée du programme est fourni accompagnée d'une capture d'écran en annexe.
Après avoir récupéré l'intégralité des données qui m'intéressait en base, j'ai pu me concentrer sur la mise en valeurs de ses données. En utilisant le langage SQL via Workbench, j'ai pu taper une requête

spécifique permettant d'afficher les données, des catégories d'incendies, du nombre d'incendies, des départements, et des trois années sur lesquelles, je voulais faire ma démonstration.

- *Colonnes de départements,*
- *Catégories d'incendies,*
- *Nombre d'incendies pour chaque année. EN ANNEXE*

La visualisation de mes données sur Workbench m'a permis de voir quelles données j'allais mettre en évidence dans mon API.
Pour la conception de mon API, j'ai utilisé le Framework Flask, il faut savoir que Flask a une architecture particulière organisée en trois compartiments, on a d'un côté le dossier **static** comprenant le fichier ccs, le dossier **templates** comportant le fichier index et graph en html. Les fichiers config init et views demeure dans le dossier **app**.

III. *Voir annexe partie Flask*

App :
config.py→ Le fichier correspond à la configuration du Framework
views.py→ Le fichier correspond au rendu visuel du fichier Template
__init__.py→ Le fichier correspond à l'initialisation de l'API

Templates :
index.html → Le fichier correspond à la conception des balises HTML (HyperText Markup Language) et inclus du jinja2 pour la création du formulaire de la première page.
graph.html → Le fichier correspond au rendu graphique de l'API, comportant du code en JavaScript dans la deuxième page contenant les Dashboard

Static
style.css → Sert à donner une mise en page avec des couleurs et des images.

Anatomie du projet fil-rouge :

```
├── app
│   ├── config.py
│   ├── __init__.py
│   ├── models.py
│   ├── __pycache__
│   │   ├── config.cpython-37.pyc
│   │   ├── __init__.cpython-37.pyc
│   │   ├── models.cpython-37.pyc
│   │   └── views.cpython-37.pyc
│   ├── static
│   │   ├── image
│   │   │   └── incendies.jpeg
│   │   └── style.css
│   ├── templates
│   │   ├── graph.html
│   │   └── index.html
│   └── views.py
```

L'API nous donne donc un rendu graphique et cartographique tout deux codés dans le fichier config.py, puis via le fichier views.py un rendu de **Templates** qui nous donnera les premiers aperçus de l'interface web, deux chaînes déroulantes et un bouton submit (« Envoyer ») à son actif conçu dans

le fichier index.html pour nous envoyer vers la deuxième page de l'API présentant les Dashboard. Jinja servira d'interpréter mon code html, en utilisant des balises de contrôle de flux pour envoyer sur le root. Le fichier graph.html est à l'origine des rendus de Dashboard présent sur la deuxième page de l'API. Enfin le fichier style.css sera à l'origine de la mise en page ainsi que de la coloration et des images.

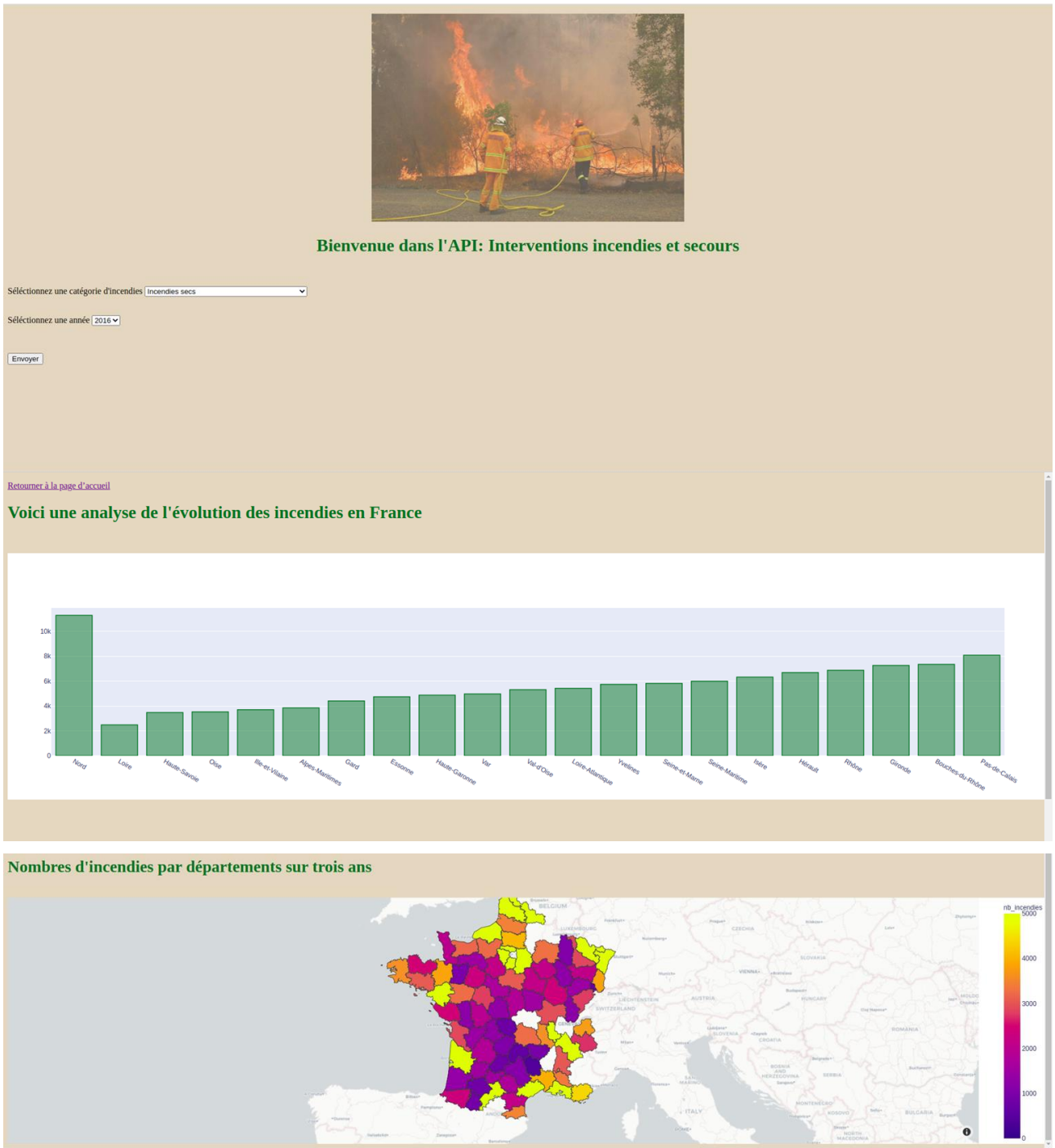


Illustration 10 : API Interventions incendies et secours

Flask a été la partie la plus complexe, due à son architecture particulière, mais aussi à l'intégration de code HTML, de JavaScript et de CSS dans le projet. Étant donné que j'étais en formation de Développeur Data, j'ai dû entreprendre des cours rapides sur Open Classroom pour réussir à contrer la difficulté du langage web à laquelle j'ai dû faire face. L'expérience a été intéressante. La complexité

du moteur de Template Jinja n'a pas non plus été une tâche facile, je me suis également penché sur Open Classroom mais aussi sur YouTube. Certains des apprenants m'accompagnant m'ont aussi aiguillé.

Le fonctionnement de Flask a été difficile à assimiler, après plusieurs opérations et d'apprentissage sur des plateformes vidéo YouTube, j'ai pu surmonter certaines difficultés.

3) Projet fil-rouge – Bilan et améliorations

Mon projet de fin de formation avait pour but de faire un travail de visualisation de données (Data visualisation) tout en répondant à un cas d'étude qui été ici la cause du grand nombre d'incendies en France. Durant la progression de mon projet, j'ai pu faire le constat d'un nombre important d'incendies au niveau des départements comme le Nord pour la catégorie de feu secs.

Ce projet m'a donné la possibilité de découvrir de nouvelles perspectives de développement, comme le web avec Flask conçu pour la conception d'interface web simple.
Il m'a également permis de forger ma capacité à avoir les bons réflexes du « Learning by doing » un des piliers de la pédagogie active ; apprendre à chercher l'information imiter, adapter, transposer à mon projet.

En prenant du recul, j'ai pu constater que mon API aurait pu bénéficier d'améliorations, à commencer par l'ajout de données supplémentaires, bien que présente dans le MCD, les données forestières auraient pu faire partie de mon API. Celles-ci une fois asséchées par des températures élevées jouent le rôle de combustible et donc d'amplificateur d'incendies. Les données des températures auraient également pu contribuer à la démonstration d'une relation de cause à effet avec des facteurs climatiques et environnementaux.

L'ajout de Dashboard divers et variés accompagnée d'un coefficient de corrélation ***** aurait pu pareillement contribuer à l'amélioration de mon API en apportant des résultats plus pertinents.

Le projet sera suivi de nombreuses itérations telles que cités plus haut.

coefficient de corrélation :
La valeur obtenue est bornée par -1 et 1.
• Si proche de 0 les variables sont non-corrélées.
• Si proche de 1 les variables sont corrélées.
• Si proche de -1 les variables sont anti-corrélées.

III. Première partie – Projet de stage au Consulat général du Sénégal

1) Compréhension du besoin client

Pour le besoin client je me suis référé à M.Abdourramane Koita le consul, M.Ging était quand à lui notre tuteur de stage.

Le consul nous avait fait part de sa demande, qui consistait à faire une étude préalable sur la conception d'une application de suivi des demandes d'actes d'état-civil, pour les ressortissants Sénégalais.

Pour répondre au mieux au besoin nous avons mon équipe et moi dans un premier temps laissé place à l'écoute du problème qu'il rencontrait :

- Quelles sont les attentes et les besoins de l'application ?
- Quels sont les objectifs ?
- Quels vont être les contenus ? Comment vont-ils être organisés ?
- Quels sont les services proposés ?

2) Gestion de projet

Notre déploiement et la mise en place de la méthode agile s'est fait dans une première approche d'analyse du besoin. Nous avons pris du recul et nous nous sommes donnés un temps imparti pour évaluer et pour creuser au maximum les besoins afin d'éviter toute digression sans autorisation.

Pour ce faire nous nous sommes réparti le travail en deux équipes : l'équipe en présentiel, dont j'en étais le Product Owner, et L'équipe en back log.

L'équipe en présentiel devait réaliser une interview du personnel. Le travail consistait à mettre en place des questions spécifiques pour la bonne marche du projet. C'est alors que nous avons sorti une première version d'interview.

L'équipe en back log quant à elle s'est chargeait de la mise en place du Trello.

Pour ce projet ils nous étaient essentielles de réaliser un modèle conceptuel de communication. Grâce au MCC, on peut avoir une vue d'ensemble sur le fonctionnement du Consulat concernant les demandes d'acte d'état-civil. Cela nous permet de voir qui sont les acteurs concernés, quels sont leurs rôles, comment transitent les informations.

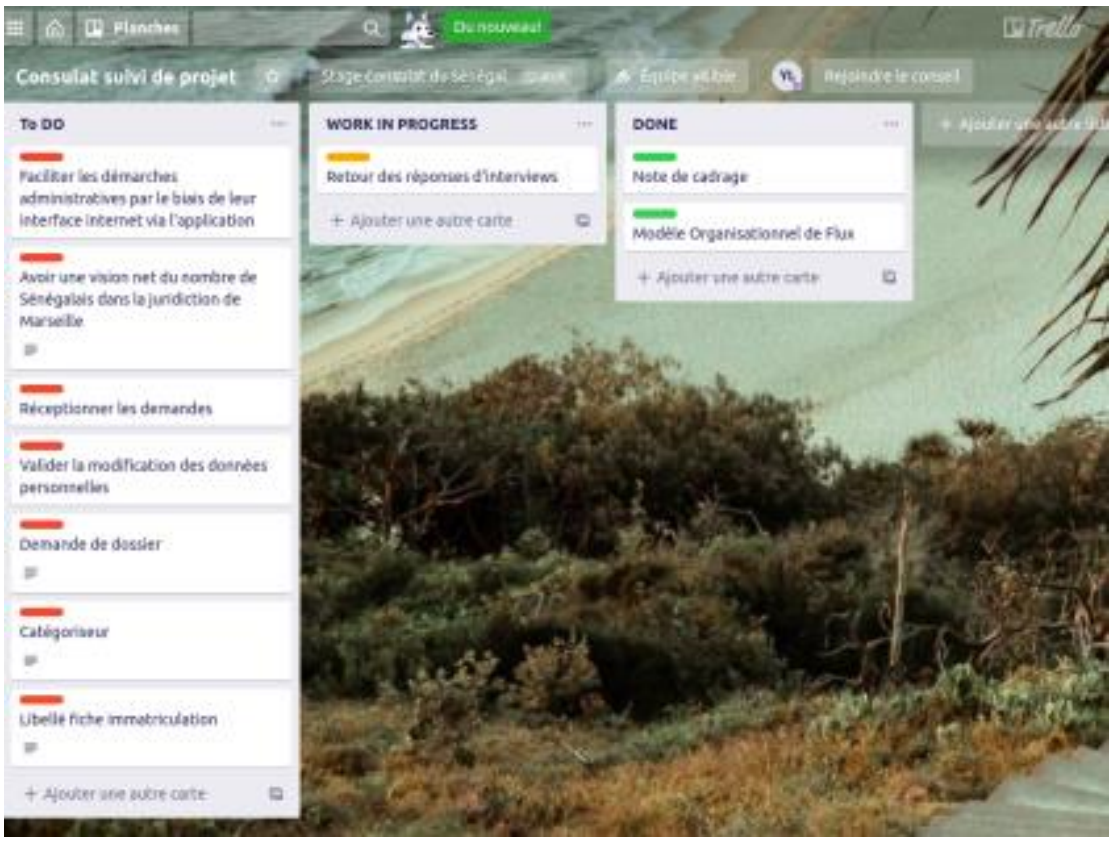


*Illustration 11 : Image de gestion de projet de stage avec Trello : **A faire**, **en cours**, **fait***

Le but était d'obtenir un maximum d'informations durant toute la première semaine afin de proposer une première version d'un MCC. Le modèle conceptuel de communication est un système qui peut être coupé en trois étapes :

- La première étape de ce modèle est d'arriver à isoler le système en le délimitant. Il s'agit donc de définir le système et les éléments externes avec lesquels il échange des flux d'information. Ces éléments extérieurs sont appelés *acteurs externes* (ou partenaires).

- La seconde consiste à découper l'organisation en entités appelées *acteurs internes* (ou domaines). Lorsque les domaines d'une organisation sont trop importants, ils peuvent être décomposés eux-mêmes en *sous-domaines*.

- La dernière est l'analyse des flux d'information, c'est-à-dire la définition des *processus*.

En plus du MCC, nous voulions également proposer en premier lieu une première version de note de cadrage.
Une fois le plan établi et les premières questions faites, la première version d'interview vu le jour.

Compte rendu d'interviews : Projet M2 : Étude préalable Application de suivi des demandes d'actes d'état-civil

Déterminer les acteurs internes, acteurs externes : quelles sont les informations échangées ?

Les acteurs internes sont le secrétariat du consulat et le bureau consulaire y compris tous les services suivants :

- *Transcription des naissances*
- *Transcription mariages*
- *Cartes consulaires*
- *Sauf-conduits*
- *Certificat de coutumes*
- *Certificat de déménagement*
- *Passeport ordinaire*
  pour ce qui est des acteurs externes, ce sont les usagers

Comment s'y prennent les ressortissants sénégalais pour la demande acte d'état civil ?

Pour prendre un exemple selon la demande, exemple : demande de transcriptions de naissance, l'usager doit récupérer à la mairie une copie intégrale de l'acte de naissance ainsi qu'une pièce justificative d'identité sénégalaise venant du père ou de la mère

Voir avec le consul pour faire la note de cadrage : Monsieur Koita

Pourquoi a-t-il voulu faire cette API ?

L'objectif de cette API est de faciliter les démarches administratives directement sur le site, cela évite le déplacement des usagers ainsi qu'avoir une vision nette du nombre de ressortissants sénégalais en France

Souhaitez -vous enregistrer tous les ressortissants ? Si oui dans quelle juridiction ?

Les juridictions dans lesquelles on souhaite les enregistrer sont celles-ci : Alpes de Hautes-Provence (04), Alpes Maritimes (06), Bouches-du-Rhône (13), Hérault (34), Var (83)

Quelles sont les technologies utilisées (Audit tech) ?

- *Word*
- *Excel*
- *Fichier .PDF*
- *Power point*
- *Système d'exploitation : Windows 10*

Nombre de demandes annuelles pour Transcriptions naissance / Transcription mariage / Cartes consulaires / Saufs-conduits / Certificats de coutume / Certificats de déménagements/ Passeports ordinaires ?

D'après les estimations de statiques annuelles de 2018 et de 2019 que M. Gning nous a remis :

- *Transcriptions naissance :*
- *2018 : 240 demandes*
- *2019 : 228 demandes*

-Transcriptions mariage :

- *2018 : 4 demandes*
- *2019 : 11 demandes*
- *Cartes consulaires :*
- *2018 : 118 demandes*
- *2019 : 98 demandes*

-Sauf-conduits :

- *2018 : 155 demandes*
- *2019 : 228 demandes*

-Certificats de coutumes :

- *2018 : 68 demandes*
- *2019 : 67 demandes*

-Certificats de déménagement

- *2018 : 9 demandes*
- *2019 : 5 demandes*

-Passeports ordinaires

- *2018 : 787 demandes*
- *2019 : 771 demandes*

Notes : Suite à la demande de M. Gning, j'ai établi un récapitulatif des demandes avec ces services de demande ci-dessus seulement. À préciser avec M. Le consul.

À l'aide des informations que nous avait communiqués M.Gning, nous nous sommes penchés dans la réalisation d'une première version d'une note de cadrage, les réponses aux questions posées dans l'interview fournirent les éléments nécessaires pour poser les périmètres du projet

Première version de note de cadrage

Date : Mardi 10 Mars
Équipe sur le terrain : Yacine Bourezak, Lotfi Baya
Équipe en backlog: Sacha Kojic, Yoann Lucido
Référent : Samba Cissé
Version : 0.0.1

Projet – Note de cadrage – Consulat général du Sénégal à Marseille

M2: Étude préalable Application de suivi des demandes d'actes d'état-civil

Objectifs du projet :
- Faciliter les démarches administratives par le biais de leur interface internet via l'appli.
- Avoir une vision nette du nombre de Sénégalais dans la juridiction de Marseille dans les départements suivants… Alpes de Haute Provence (04), Alpes Maritimes (06), Bouches du Rhône (13), Hérault (34), Var (83)
- Exemple :

-Réceptionner les demandes (Acte d'état civil, visa, légalisation des documents…)
-Valider la modification des données personnelles avec M. Ning
-Demande de dossier : accusé de réception pour le reçu du passeport
dossier en cours de traitement, dispo…
-Mieux les connaître, les catégoriser (la juridiction dans laquelle sont les ressortissants qui font des demandes)
-Avoir les libellés de la fiche d'immatriculation
contact service technique pour la mise à jour (Samba Cissé)

Sécurité RGPD, quel volume, serveur local, cloud ? répondre à ces questions pour eux

Besoin :
- Transposer le prospectus (les infos pour la carte consulaire) en numérique.
- L'application permettra une amélioration capitale au niveau des démarches administratives
- Exemple pour la **carte consulaire** :

-Numérisation de la fiche d'immatriculation.
-Numérisation de la photo du ressortissant Sénégalais.
-Numérisation de la pièce d'identité sénégalaise.
-Numérisation du justificatif de domicile.

- Les passeports
- Acte d'état civil
- Les certificats de déménagements

Bénéfices :
- Permet d'avoir un gain de temps lors des procédures, facilite également l'optimisation des données et la limitation de l'utilisation manuscrite

Points à surveiller :
- Vérification du statut sénégalais de l'individu lors des enregistrements en ligne.

Technologie du client :

- Word
- Excel
- Fichier .PDF
- Power point
- Système d'exploitation : Windows 10

Services impliqués :

- Bureau consulaire
- Bureau des passeports

À propos du Consulat Général du Sénégal :

- Adresse : 446 Avenue du Prado 13008 Marseille
- *Responsable : M. Ning*

3) Réalisation de projet

À la suite, nous avons présenté cette première note de cadrage à Monsieur Gning. Une fois lu, et validé, en suivant les consignes qui nous était donné nous avons entamé la réalisation du MCC.

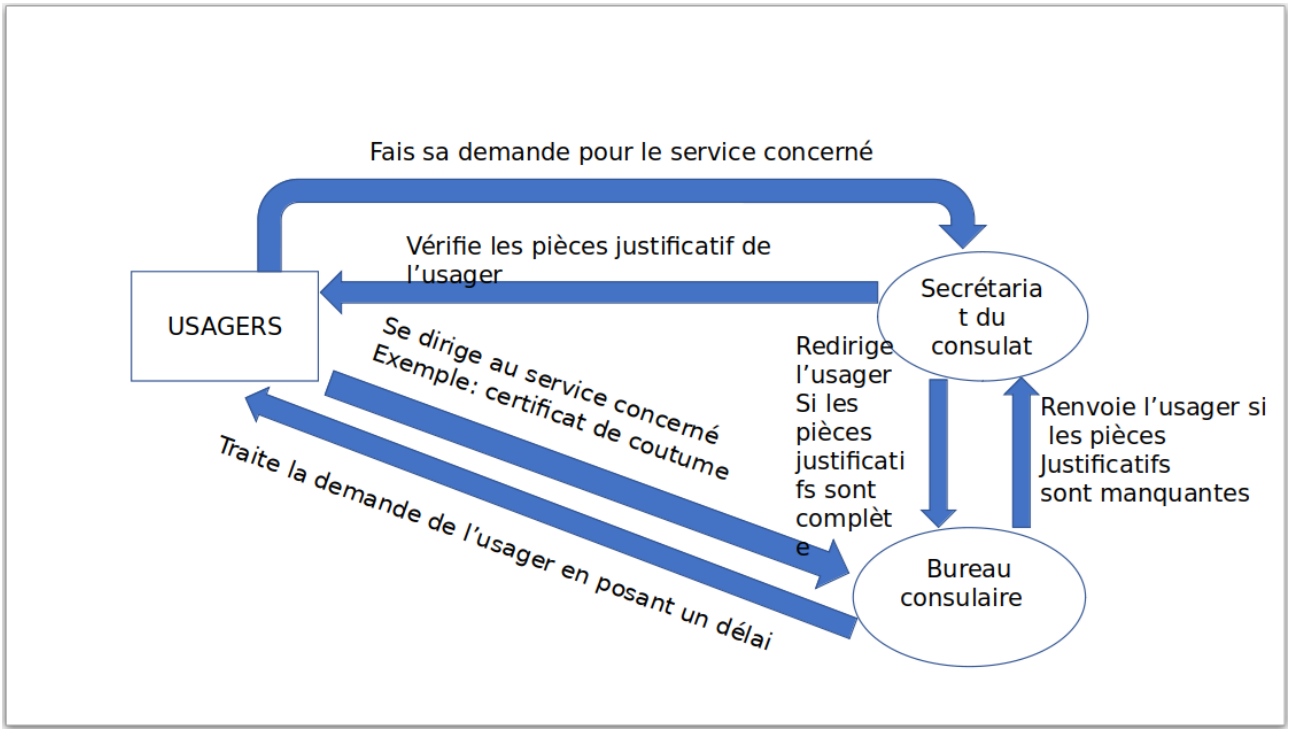


Illustration : Modèle Conceptuel de Communication du Consulat général du Sénégal

Le MCC a nécessité des évolutions au fur et à mesure que l'on a avancés dans le projet. On a donc réalisé une deuxième interview pour compléter la première.

Interviews : Projet M2: Étude préalable Application de suivi des demandes d'actes d'état-civil

- *Quels acteurs internes reçoit la demande d'acte d'état civil ?*
- *Que voulez-vous faire exactement, concernant les suivis de demandes d'actes-civil ?*
- *Quelle tâche voulez-vous automatiser, concernant le suivi des demandes d'actes d'état civil ?*
- *Qu'est-ce qu'une demande d'acte d'état civil précisément au sein du consulat pourrait vous nous donnez un exemple ?*
- *D'après M. Gning au sein du bureau consulaire, il y aurait plusieurs services qui traitent les demandes des usagers notamment ceux cités plus hauts ; Transcription de mariages, etc.., pouvez-vous nous confirmer sa ? (M. le consul)*

- *Si oui, comment travaillent les services entre eux ? Quels manquements aurait-il ? Dans le sens quel est le besoin d'amélioration au niveau de ses services ? (Question toujours adressé a M. Le Consule)*
- *Votre besoin est toujours bien le suivant ? « Dématérialisation des documents d'administration », Si oui, pouvez-vous nous donner un exemple ?*
- *Est-ce que tout le monde à un email ?*
- *Quelle information doit être automatisée ?*

Compte tenu de la situation, que la France a vécu à ce moment-là (crise sanitaire du Covid-19), le consulat général du Sénégal à Marseille ainsi que toutes les structures d'entreprises ont dû fermer leurs portes sur ordre du Président de la république pour entrer en confinement. Les circonstances exceptionnelles liées à l'épidémie du Corona n'ont pas permis de poursuivre le stage ni en présentiel ni en distanciel.

4) Projet de Stage au Consulat Général du Sénégal – Bilan et améliorations

Le stage au consulat m'a permis de mieux comprendre la méthode agile. Compte tenu du peu de temps que nous avons passé dans la structure, nous n'avons pas pu exploiter l'expertise technique acquise durant la formation. Le MCC était une technique de traitement de données que nous avions vu en formation, ainsi le stage m'a permis un approfondissement de mes connaissances sur cette notion.

La réalisation des interviews pour le client a été l'une de mes premières expériences en entreprise c'est-à-dire cerner le besoin du client, trouver la bonne approche, bien adapter les réponses au projet. Cette interaction entre le client et moi-même m'a permis de développer mes connaissances sur la gestion de projet et la méthode agile.

CONCLUSION

La formation de Développeur(se) Data a été une formation très enrichissante, tant au niveau technique mais également au niveau de la gestion de projet. Elle m'a permis de voir les méthodes d'entreprise, sous forme de nombreux projets réalisés tout au long du cursus, celle utilisée particulièrement ici est Merise. Elle m'a également permis d'acquérir un bon nombre de connaissance en utilisant la pédagogie active qui repose sur un thème qui est le « Learning by doing » en effet, nous avons travaillé, au cours de la formation, sur plusieurs projets en utilisant des méthodes d'entreprise. Ici la méthode agile était la méthode au centre de nos projets.

Au niveau technique, la formation m'a permis d'avoir une vue d'ensemble sur les bases du monde de la Data, en passant par les différents systèmes de gestion de bases de données, les technologies utilisées, à travers différents projets. J'ai pu voir en surface les techniques de programmation permettant de modéliser, de développer une base de données et de concevoir des programmes destinés à l'importation des données, ainsi que d'autres compétences, comme la création d'interface web avec Flask.

Je tiens à remercier toute l'équipe pédagogique ainsi que les apprenants qui ont grandement contribués à l'aboutissement de ce projet. Je salue particulièrement le chargé de projet de formation Emmanuel Rasquero, pour le soutien moral mais également pour les « meet up » et les « job dating » organisés avec les entreprises. Les formateurs Anne-Laure Mealier, Rafik Lachaal et également Samba Cissé qui nous ont accompagné et coacher tout au long de la formation.

ANNEXES

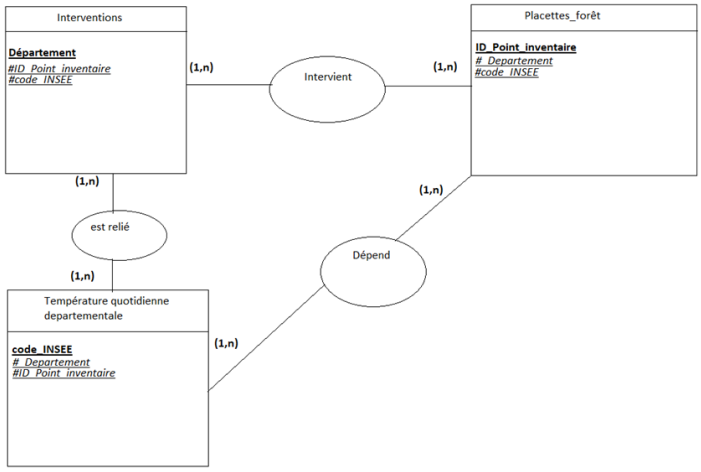
I. Partie sur MySQL Workbench

```sql
CREATE DATABASE interventions_incendie_et_secours;

CREATE TABLE interventions_2010
(
annee varchar(30),
zone varchar(50),
region varchar(50),
numero int(45),
departement varchar(50),
categorie int(20),
incendies int(70),
secours_a_la_victime int(100),
aides_a_la_personne int(100),
secours_a_la_personne int(100),
accident_de_circulation int(100),
risques_technologiques int(100),
protection_des_biens int(80),
divers int(80),
operations_diverses int(80),
total_interventions int(100)
);

```

Création de la base de donnée Interventions incendies et secours
Création de la table interventions_2010 avec les colonnes



Modèle conceptuel des données
- Interventions à Placettes_forêt : Les interventions interviennent au minimum 1 fois, et au maximum n fois
- Placettes_forêt à Interventions : Les placettes interviennent au minimum 1 fois et au maximum n fois
- Interventions à Température : Les interventions sont reliées au minimum 1 fois et au maximum n fois
- Température à Interventions : Les températures sont reliées au minimum 1 fois et maximum n fois
-Température à Placettes : Les températures dépendent au minimum 1 fois et maximum n fois

-Placettes à Température : Les placettes dépendent au minimum 1 fois et au maximum n fois

```sql
-- Création d'un utilisateur --
CREATE USER 'yacine-fil-rouge'@'localhost' IDENTIFIED WITH mysql_native_password BY 'yacine13';
-- Attribution de droit par exemple de créer des tables les supprimer ou autres
GRANT ALL ON *.* TO 'yacine-fil-rouge'@'localhost';
-- Accord de tous les privilèges
FLUSH PRIVILEGES;
```

Création d'un utilisateur, **lui attribuer** tous les droits, et lui **accordé tous les privilèges**

II. Programme d'importation des données en base

```python
#Importation des données
from sqlalchemy import create_engine
import pandas as pd
import time

#Établir la connection entre Python et MySQL
engine = create_engine("mysql+pymysql://yacine-fil-rouge:yacine13@localhost/interventions_incendie_et_secours")

def chargement(link, table):
    """
    Chargment des données dans les tables de la base de données Interventions incendie et secours
    """
    print("Lecture des données")
    csize = 60000
    nom_de_colonne = ["annee",
                      "zone",
                      "region",
                      "numero",
                      "departement",
                      "categorie",
                      "incendies",
                      "secours_a_la_victime",
                      "aides_a_la_personne",
                      "secours_a_la_personne",
                      "accident_de_circulation",
                      "risques_technologiques",
                      "protection_des_biens",
                      "divers",
                      "operations_diverses",
                      "total_interventions"]

    df = pd.read_csv(link, chunksize = csize,  skiprows = [0] , header = None, encoding ='latin-1',sep = ';', names = nom_de_colonne)
    # On assigne à la variable df un pd.read_csv pour lire les fichers csv des tables un chunksize
    #pour la taille des données,
    #le skiprows pour ne sauter aucune ligne,
    #header correspond à l'entête on lui dit aussi qu'il n'en a pas, l'encodage en latin-1 pour les fichiers qu'on a
    # Penser à mettre le sep = '; ou /t' pour bien séparer les données et les mettres chacunes dans leurs colonnes respectives
    # nom_colonne = names le nom de la variable
    print("Données lu")
    i = csize

    for chunk in df:
        chunk.to_sql(table, con = engine, if_exists='append', index = False)
        i += csize
        #Mise en place d'une condition pour chunk dans le dataframe
        print(i)
    return print("Fin")

chargement("https://www.data.gouv.fr/s/resources/interventions-realisees-par-les-services-d-incendie-et-de-secours/20170509-110128/interventions2010.csv", "interventions_2010")
```

Le programme d'importation des données comporte *:*

- *Une fonction « chargement ».*
- *Une attribution de variable au nom des colonnes disposant des tables,*
- *Une assignation au dataframe d'une lecture de fichier csv a l'aide de pandas ainsi que des conditions d'indexation,*
- *Et également d'autres conditions de lecture du dataframe.*
- *Une boucle for chunksize qui va servir a coupé les quantités de données en partie*
- *Et enfin l'appel de ma fonction chargement avec en paramètre l'URL de la table spécifique à importer provenant du site de data.gouv contenant les données à extirper. Le nom de la table en .csv était également présent en paramètre.*

Requête SQL permettant de faire un travail sur les trois tables

```sql
SELECT annee, interventions_2016.departement, categorie, incendies
FROM interventions_2016
JOIN temperature_quotidienne_regionale ON interventions_2016.region = temperature_quotidienne_regionale.region AND annee = %s AND categorie = '%s'
GROUP BY annee, categorie, incendies, departement

UNION

SELECT annee, interventions_2017.departement, categorie, incendies
FROM interventions_2017
JOIN temperature_quotidienne_regionale ON interventions_2017.region = temperature_quotidienne_regionale.region AND annee = %s AND  categorie = '%s'
GROUP BY annee, categorie, incendies, departement

UNION

SELECT annee, interventions_2018.departement, categorie, incendies
FROM interventions_2018
JOIN temperature_quotidienne_regionale ON interventions_2018.region = temperature_quotidienne_regionale.region AND annee = %s AND categorie = '%s'
GROUP BY annee, categorie, incendies, departement
ORDER BY incendies, categorie;
```

La requête, ***sélectionne*** la colonne année, la colonne département, la colonne catégorie, la colonne d'incendies ***de*** la table interventions_2016
À la suite j'ai fait une ***jointure*** sur la table température avec la table interventions avec la clé région, j'ajoute à ça un ***Group By***, permettant de grouper mes colonnes et enfin un ***Order By*** permettant d'ordonner mes colonnes du plus petit au plus grand. Une fois la requête terminée, j'ai placé une Union afin de joindre les trois tables interventions avec les trois années différentes avec lesquelles j'allais travailler.

III. Partie Flask

Le dossier app contient tous les fichiers qui suivent

```python
from app import app

app.run(debug = True, host = '127.0.0.1', port = 5000)
# Production
app.run(host='0.0.0.0')

```

Le fichier runapp.py comporte un host pour la production du Template

```
#!/usr/bin/env python
# coding: utf-8

# import library
from flask import Flask
from app import config

# Init flask app
app = Flask(__name__ )


app.config.from_object(config)



# import library
from app import views
from app import models

# le fichier init sert a initialiser l'app de flask
```

Le fichier init.py comporte l'importation de la librairie Flask et des librairies du fichier app

```python
import pandas as pd # procès de données, CSV file I/O (e.g. pd.read_csv)
from plotly.offline import init_notebook_mode, iplot, plot
import plotly
import plotly.graph_objs as go
import json
from flask import render_template
from sqlalchemy import create_engine
from urllib.request import urlopen
import plotly.express as px
 #Etablir la connection entre Python et MySQL
 engine = create_engine("mysql+pymysql://yacine-fil-rouge:yacine13@localhost/interventions_incendie_et_secours")


def create_plot1(annee, categorie):
    """Traitements des requêtes et création de figure"""


    query1 = pd.read_sql_query("""SELECT annee, interventions_2016.departement, categorie, incendies as nb_incendies,
    AVG(temperature_moyenne) as moyenne_temperature
    FROM interventions_2016
    JOIN temperature_quotidienne_regionale ON interventions_2016.region = temperature_quotidienne_regionale.region AND annee =
        %s AND categorie = '%s'
    GROUP BY annee, categorie, nb_incendies, departement

    UNION

    SELECT annee, interventions_2017.departement, categorie, incendies as nb_incendies,
    AVG(temperature_moyenne) as moyenne_temperature
    FROM interventions_2017
    JOIN temperature_quotidienne_regionale ON interventions_2017.region = temperature_quotidienne_regionale.region AND annee =
        %s AND categorie = '%s'
    GROUP BY annee, categorie, nb_incendies, departement

    UNION

    SELECT annee, interventions_2018.departement, categorie, incendies as nb_incendies,
    AVG(temperature_moyenne) as moyenne_temperature
    FROM interventions_2018
    JOIN temperature_quotidienne_regionale ON interventions_2018.region = temperature_quotidienne_regionale.region AND annee =
        %s AND categorie = '%s'
    GROUP BY annee, categorie, nb_incendies, departement
    ORDER BY annee, departement, nb_incendies;""" %(annee, categorie, annee, categorie, annee, categorie), con = engine)

  # Création de la trame 1
    trace1 = go.Scatter(
                        x = query1.moyenne_temperature,
                        y = query1.nb_incendies,
                        mode = "markers",
                        name = "Nombre d'incendies en fonction des moyennes des température",
                        marker = dict(color = 'rgba(16, 112, 2, 0.8)')
                        #text = df.
                        )

    data = [trace1]

    layout = dict(title = "Nombre d'incendies en fonction des températures moyenne dans l'hexagone",
                  xaxis = dict(title = "Moyenne des températures",ticklen = 5,zeroline = True),
                  yaxis = dict(title = "Nombre d'incendies",ticklen = 5,zeroline = True)
                  )
    fig = go.Figure(data = data, layout = layout)



    graphJSON = json.dumps(fig, cls = plotly.utils.PlotlyJSONEncoder)

    return graphJSON
```

Le fichier config.py contient tout d'abord l'importation des librairies nécessaire

- Pandas → servira ici pour la structure des données

- Plotly → servira ici pour le rendu en forme graphique des données
- json → permet l'échange de données inspiré par la syntaxe des objets littéraux de JavaScript
- render_template → le rendu de template, permet de donner un rendu d'un fichier html
- sqlalchemy → un accès à la base de données efficace et hautement performant,
- urllib → modules travaillant avec les URLs

engine = create_engine → servira ici à faire la connexion entre MySQL et Python

À la suite j'ai créé une fonction create_plot1 pour mon premier plot et je lui ai donné en paramètre, année et catégorie les colonnes qui m'intéressait

J'ai utilisé pandas pour lire ma requête sql avec le pd.read_sql

J'ai ensuite utilisé plotly pour le plot que je voulais afficher j'ai choisi les Bar comme Dashboard pour mes données

J'ai demandé ensuite de retourner le graph json afin de m'afficher le Dashboard dans le render_template.

```python
from app import app

# importer les autres éléments déclarés
# dans /app/__init__.py selon les besoins
# from app import db, babel
# importer les modèles pour accéder aux données

from app.models import *
from flask import render_template, request
from app.config import create_plot1, create_plot2



@app.route('/', methods = ['GET', 'POST'])
def index():
  return render_template('index.html')


@app.route('/dashbord', methods = ['GET', 'POST'])
def dashbord(): #Fonction permettant d'afficher le dashbord avec ses paramètres et attribut
  annee = request.form['annee'] # Assignation d'une variable annee avec la méthode de la bibliothèque request avec la méthode
    form qui a comme attribut le name annee de l'html
  categorie = request.form['categorie']


  plot1 = create_plot1(annee, categorie) #Création d'une variable, pour la valeur de create_plot qui est graphJSON
  plot2 = create_plot2() #Création d'une variable, pour la valeur de create_plot2 qui est graphJSON


  return render_template('graph.html', plot1 = plot1, plot2 = plot2, annee = annee, categorie = categorie) # Transcrire la
    valeur du paramètre de la méthode dans le render_template pour avoir un rendu visuel

  #return render_template ou rendu template, retourne la page html

  #Le fichier views.py réunit le config et les input html pour avoir la vue sur le navigateur
```

Le fichier views.py → Ce fichier va décider du contenu a affiché sur les pages, plus spécifiquement, ce sont les vues qui génèrent le contenu à renvoyer aux requêtes qui leur sont adressées.

Ici j'ai créé deux variables pour mes deux plots, avec chacun leurs assignations de méthode le plot 1 a les mêmes paramètres que dans le fichier config

À la suite un return_template avec en paramètre la transcription des valeurs du paramètre de la méthode dans le render_template pour avoir un rendu visuel.

```
<head class = 'conteneur'>
<link rel="stylesheet" href="static/style.css" /> <!--Le css est relié avec le html via ce lien pour faire la mise en page de
    toute les pages-->

  <p> <!-- picture -->
    <img src="/static/image/incendies.jpeg" alt = "incendies" />
  </p>

<title>Bienvenue dans l'API: Interventions incendies et secours</title>
</head>
<body>

  <h1>Bienvenue dans l'API: Interventions incendies et secours</h1><br />


  <form method = "post" action = "{{ url_for('dashbord') }}"> <!-- Un lien pour envoyer la deuxième route dashbord-->
    <p>
        <label for="categorie">Séléctionnez une catégorie d'incendies</label> <!--Création d'une liste de déroulante des
            categories d'incendies-->
        <select name="categorie" id="categorie">
            <option value="1">Feu secs</option>
            <option value="2">Feu gras</option>
            <option value="3">Feu de gaz(gaz naturels, gaz de pétrole)</option>
            <option value="4">Feu de métaux</option>
            <option value="5">Feu d'origine éléctrique</option>
        </select>
    </p><br />

        <label for="annee">Séléctionnez une année</label> <!--Liste déroulante des trois années-->
        <select name="annee" id="annee">
            <option value="2016">2016</option>
            <option value="2017">2017</option>
            <option value="2018">2018</option>
        </select>
    </p><br />
```

Le fichier index.html → comporte quant à lui des balises de title pour les titres en haut de page, des balises de contrôle de flux avec Jinja pour envoyer un lien sur le deuxième template, deux listes déroulantes et un bouton submit pour renvoyer à la deuxième page contenant les plots à afficher

```
    <p> <input type="submit" value="Envoyer" /> </p> <!--Création du bouton submit afin d'envoyer à la deuxième page présentant
        les dashboard-->
```

```
<!doctype html>
<html>
<head>
  <title></title>
</head>
<link rel="stylesheet" href="static/style.css" />
<body>
<p><a href= "{{ url_for('index') }}"> Retourner à la page d'accueil </a></p>
  <h1 class = 'incendies'>Voici une analyse de l'évolution des incendies en France</h1><br />
  <p></p>

  <div class="container">
    <div class="row">
        <div class="col-md-6">
            <div class="chart" id="scatter">
            </div>
        </div>
        <script src = "https://cdn.plot.ly/plotly-latest.min.js"> </script>
        <script src = "https://cdnjs.cloudflare.com/ajax/libs/d3/3.5 .6 / d3.min.js "> </script>
        <div class = "chart" id = "scatter">
            <script>
                var graphs = {{plot1 | safe}};
                Plotly.plot ('scatter', graphs, {});
            </script>
        </div>
        <br/>
```

```
            <h4>Nombres d'incendies par départements sur trois ans</h4>
              <div class="chart" id="scatter">
              </div>
          </div>
          <script src = "https://cdn.plot.ly/plotly-latest.min.js"> </script>
          <script src = "https://cdnjs.cloudflare.com/ajax/libs/d3/3.5 .6 / d3.min.js "> </script>
          <div class = "chart" id = "scatter2">
              <script>
                  var graphs = {{plot2 | safe}};
                  Plotly.plot ('scatter2', graphs, {});
              </script>
          </div>
      </div>

  </div>
</body>
</html>
```

Le fichier graph.html → contient quant à lui le code JavaScript qui servira à l'affichage de Dashboard, avec l'aide des balises de contrôle de flux.

```css
body{
    font-family: Georgia, 'Times New Roman', Times, serif;
    background-color: rgb(230, 212, 193);

}

h1{
    color: rgb(2, 102, 35);
    text-align: center;
}

h4{
    color: rgb(2, 102, 35);
    font-size: 30px;
}

.incendies{
    color: rgb(2, 102, 35);
    text-align: left;
}
img{

    margin-left: 35%;
    margin-right: 250%;
    width: 30%;
    height: 100%;
    opacity: 0.6;
}
```

Le fichier style.css → Sert ici à la mise en page et aux colorations du rendu de template, Les h1 et h4 représente les titres avec leur position sur la page ainsi que la couleur qu'on leur donnera.

BIBLIOGRAPHIE

Source : https://www.verisure.fr/guide-securite/risques/incendie/detecteur-fumee/5-statistiques-incendie

Source : https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_relationnelle

Source :
https://fr.wikipedia.org/wiki/MySQL_Workbench

Source : https://books.google.fr/books?id=dQjtjnp-ixoC&pg=PA419&lpg=PA419&dq=mysql+workbench+inconv%C3%A9nient&source=bl&ots=P6u_CFoOp-&sig=ACfU3U1W2VaUIdWNgk-v4cUQ6U-EnmhZFA&hl=fr&sa=X&ved=2ahUKEwjspIypvOvpAhUTCWMBHXEzCGoQ6AEwBnoECAoQAQ#v=onepage&q=mysql%20workbench%20inconv%C3%A9nient&f=false

Source : https://www.editions-eni.fr/open/mediabook.aspx?idR=9e52123d6abc4308bc54a20b69ef9176

Source : https://fr.wikipedia.org/wiki/Jupyter

Source :https://www.data.gouv.fr/fr/datasets/interventions-realisees-par-les-services-d-incendie-et-de-secours/