**Explication présentation mémoire**

**Introduction** :

Lorsque nous parlons du monde du développement web, les APIs ou Application Programming Interface font partie des composants des différents programmes. Utile pour permettre à un produit ou un service d’être en interaction avec d’autres éléments, l’API sert à simplifier le processus de développement d’une application. Malgré que les internautes ne soient pas en relation directe avec les APIs, ils participent activement grâce à l’utilisation de services web.

Pour les utilisateurs, l’interface de programmation d’application rend les données ou les fonctionnalités d’une application disponible. D’une manière plus concrète, une Application Programming Interface joue le rôle d’intermédiaire entre deux systèmes informatiques indépendants pour pouvoir interagir de façon automatique, en minimisant ou sans demander l’intervention d’un utilisateur.

L’utilisateur peut ajouter des fonctionnalités sur son site ou dans son application pour les rendre de plus en plus dynamiques, sans que l’internaute ne connaisse les détails du programme. Il est important de souligner que le fournisseur qui décide de se servir d’une Application Programming Interface peut intégrer quelques modifications dans son programme sans avoir à déranger ses utilisateurs.

Pour uniformiser le développement des API une spécification open source a été adopté en 2015 et elle est appelé OpenAPI :

Le travail effectué pour la mise en place de cette nouvelle spécification comprend quatre (4) parties:

* La spécification open source existante pour les APIs qui est ***OpenAPI***
* Le modèle de communication publication/abonnement sur lequel notre nouvelle spécification est basé
* La nouvelle spécification pour les APIs du modèle publication/abonnement appelé ***OpenPS***
* Le générateur de code ***OPENAPI-PS*** et un cas d’utilisation pour un producteur (qui publie) et un consommateur (qui s’abonne).

**OpenAPI**:

OpenAPI définie une interface standard, indépendante du langage de programmation pour les API REST. Cette interface permet aux consommateurs et aux machines de découvrir et comprendre les capacités des services sans avoir accès au code.

OpenAPI est constituée de huit (8) objets qui sont : openapi, info, serveur, paths, components, security, tags et externalsDocs.

Openapi : version de la spécification OpenAPI utilisé

Into : décrit l’entête de l’api

Serveur : description et url du serveur qui contient les services auxquels l’API est supposée partager

Paths : les différentes méthodes ou opérations d’accès aux services par le protocole http. Il décrit les manières d’accès aux services autorisées par l’API

C’est la sécurité de l’API décrit par l’objet security. Ces manières d’accès sont :

Query : pour la sécurité des requêtes ou sécurité API\_KEY

Header : dans l’entête des paquets

Components : il décrit les parties réutilisables de l’API pour éviter la duplication. Une fois ces parties définit dans component on la fait référence avec l’objet $ref fourni par OpenAPI.

Security : il décrit la sécurité utilisée par l’API pour avoir accès aux donnés. OpenAPI utilise quatre types de sécurité qui sont : API\_key, Aouth2 :

Tags : c’est array tableau de description d’un ensemble d’opération

ExternalsDocs : pour faire référence a un document externe qui peut guider l’utilisateurs de l’API.

Un document de cette spécification est utilisé par l’outil Swagger qui a l’origine était cette spécification Mais qui est renommé à partir de la version 3.0.0 en OpenAPI. Maintenant OpenAPI est la spécification et Swagger l’outil utilisant cette spécification. C’est avec Swagger qu’on décrit les documents APIs et qu’on génère les codes d’implémentations en différents langages. Il permet aussi de visualiser et de tester les APIs comme le fait Postman.

* C’est un projet de linux fondation soutenu par les grandes entreprises telles que : Microsoft, IBM etc.
* Avec un seul document de cette spécification nous pouvons générer les codes en différents langage.
* La description JSON et YAML Pour les ordinateurs, ce format se génère et s'analyse facilement. Pour les humains, il est pratique à écrire et à lire grâce à une syntaxe simple et à une structure en arborescence. JSON permet de représenter des données structurées (comme XML par exemple). Ce qui évalue et assure la qualité
* Génération de document interactive avec Swagger
* Publication de l’API avec sa définition ce qui permet aux développeurs une intégration facile de l’API.

Modèle publication/abonnement :