API Node

ATELIER PROFESSIONNEL N°3

Léo LAROU-CHALOT
IPSSI | 25 RUE CLAUDE TILLIER, 75012 PARIS

Application Protocol Interface

Parce que les informations circulant sur le web ne sont que rarement dédiées qu'à une seule utilisation, les API on été pensées pour permettre une standardisation du protocole http afin de permettre l'accès à celles-ci par de mutliple applications.

Dans le cadre de ma scolarisation et de mon apprentissag, j'ai été amené à aborder ce sujet de nombreuses fois.
C'est dans le cadre de mes activités professionnelles que j'ai commencé à développer mes premières API.

Cette documentation a pour objectif de mettre sur papier ma reflexion, et permettre de présenter ma réalisation grâce à des exemples concrets.

1 LE CONTEXTE

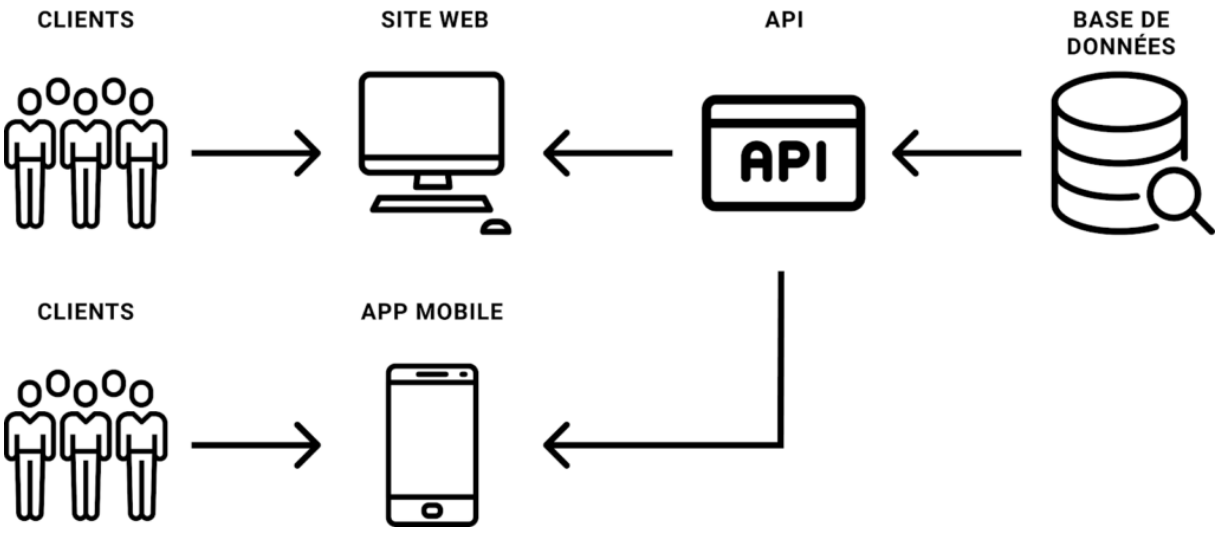
Dans le cadre des ateliers professionnels 3 et 4 j'ai du développer une API pour permettre à l'application Web et l'application Mobile de la maison des ligue de Loraine de partager certaines infromations entre elles.

2 UNE API ?

REST est un ensemble de contraintes architecturales. Il ne s'agit ni d'un protocole, ni d'une norme. Les développeurs d'API peuvent mettre en œuvre REST de nombreuses manières.

Lorsqu'un client émet une requête par le biais d'une API RESTful, celle-ci transfère une représentation de l'état de la ressource au demandeur ou point de terminaison. Cette information, ou représentation, est fournie via le protocole HTTP dans l'un des formats suivants : JSON (JavaScript Object Notation), HTML, XLT, Python, PHP ou texte brut. Le langage de programmation le plus communément utilisé est JSON, car, contrairement à ce que son nom indique, il ne dépend pas d'un langage et peut être lu aussi bien par les humains que par les machines.

Autre point à retenir : les en-têtes et paramètres jouent également un rôle majeur dans les méthodes HTTP d'une requête HTTP d'API RESTful, car ils contiennent des informations d'identification importantes concernant la requête (métadonnées, autorisation, URI, mise en cache, cookies, etc.). Il existe des en-têtes de requête et des en-têtes de réponse. Chacun dispose de ses propres informations de connexion HTTP et codes d'état.



3 LE CHOIX DU LANGAGE

Pour cette exercice, j'ai utilisé l'environnement d'éxecution NodeJs couplé au module Express pour répondre aux besoin de cette problématique.

4 ETAPES PRELIMINAIRES

La reflexion autour des « *endpoints* » aura été la plus fastidieuse. Lorsque l'on décide de mettre en place ce genre de solution, il faut au préalable choisir, et définir les différentes routes que nos applications auront besoin d'utiliser et de filtrer l'information à envoyer et recevoir afin d'optimiser les différents flux de données.

Il aura ensuite fallut découper les différents cas d'utilisation pour alléger le code et permettre une maintenance plus aisée dans le futur.

Certaines notions de sécurité et d'authentification seront abordées par la suite.