

Les API

Application Programming Interface

Introduction

- Imaginez que vous devez travailler sur une application permettant de louer simplement un vélo, ou bien de déposer un avis sur un restaurant.
- Pour que votre application soit vraiment utile, l'utilisateur doit pouvoir géolocaliser le vélo. Vous avez donc besoin d'un service de géolocalisation.
- Vous avez donc deux options, vous pouvez :
 - Passer des années avec plusieurs dizaines de développeurs à concevoir un service de carte du monde et de géolocalisation.
 - Utiliser un service de géolocalisation déjà existant : [ArcGIS](#), [Waze](#), [Open Street Map](#), etc.
- Si vous pouvez réutiliser un service déjà existant plutôt que de le développer... profitez-en !

Définition

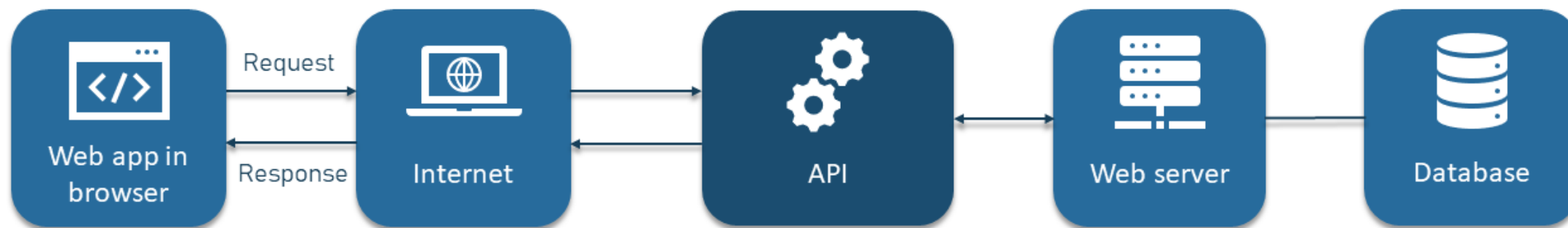
API est l'abréviation de "Application Programming Interface".

C'est une interface de programmation, ensemble de classes, de fonctions et de méthodes qui servent de façade à un logiciel.

- D'autres logiciels pourront donc accéder aux services de ce logiciel à partir de cette interface.
- L'API permet de faciliter la communication entre 2 produits ou services, comme par exemple votre application et un service de géolocalisation.
- L'API permet à ces 2 entités d'échanger des données sans en connaître les détails de mise en œuvre.

Fonctionnement

HOW API WORKS



Fonctionnement

Client - Serveur

- Pour fonctionner, une API a besoin d'un **serveur** et d'un **client**.
- On parle plus couramment **d'architecture client-serveur**.
 - un **serveur** permettant d'exécuter une API. C'est le fournisseur du service, celui qui propose au client d'accéder à ses données ou services
 - un **client** est un programme qui échange les données avec un serveur à travers l'API. C'est l'utilisateur de l'API.

Protocole Client - Serveur

- Pour la grande majorité des API disponibles sur le net, il vous sera demandé de vous inscrire au préalable afin d'obtenir **une clé ou token** permettant d'accéder aux données, aussi appelés ressources.
 - Cette clé peut être un identifiant/mot de passe, une suite de caractères ou une suite de nombres.



LES TYPE D'API

SOAP, REST et GraphQL

LES API SOAP

Simple Object Access Protocol

API SOAP

Définition

- **Simple Object Access Protocol** est un protocole officiel géré par le W3C (World Wide Web Consortium).
 - conçu pour que des applications développées avec différents langages sur différentes plateformes, puissent communiquer.
 - utilise une structure définie pour les messages et un ensemble de règles pour la transmission de ces messages
 - impose des règles intégrées qui augmentent la complexité et les coûts, ce qui peut ralentir le chargement des pages.
- Les standards de conformité intégrés dans SOAP incluent la sécurité, l'atomicité, la cohérence, l'isolement et la durabilité (ACID), qui permet d'assurer des transactions de base de données fiables.
- L'approche SOAP est hautement structurée et utilise le format de données XML.
- Gourmande en ressources et en bande passante.

API SOAP

Manipuler des ressources

- L'API SOAP peut être transmis sur de nombreux protocoles différents, tels que HTTP, SMTP et TCP.
- SOAP est également très flexible, car il peut être utilisé pour transmettre une grande variété de types de données, notamment les données structurées, les images et les fichiers audio et vidéo.

Aujourd'hui

- Bien que SOAP soit encore largement utilisé, de nombreux développeurs préfèrent désormais les API REST pour leur simplicité et leur flexibilité accrues.
- Cependant, pour certaines applications spécifiques, SOAP peut encore être un choix plus approprié en raison de ses fonctionnalités de sécurité et de fiabilité élevées.
- <https://www.sqlpac.com/fr/documents/java-protocole-soap.html>
 - Exemple d'utilisation de SOAP en Java

LES API REST

Representational State Transfer.

API REST

Définition

- REST signifie **Representational State Transfer**.
- REST permet à des logiciels qui sont incompatibles, qui ne parlent pas le même langage, de communiquer facilement.
- Tout service Web défini selon les principes de REST peut être appelé un Service Web **RestFull** si :
 - **client-server** : mode de communication
 - **stateless server** : requête qui contient toutes les informations nécessaires au traitement
 - **cache** : la réponse du serveur doit être mise en mémoire côté client.
 - **uniform interface** : la méthode de communication doit être sous la forme d'une url
- Ce n'est pas un protocole **mais un modèle architectural**.
- L'API REST est **performante** car beaucoup de clients peuvent se connecter sur un serveur à la fois
- **Moins gourmande en bande passante**.
- simplicité de mise en œuvre et plus grande tolérance à la panne (un serveur peut être en panne mais un autre pourra vous répondre), et donc sa fiabilité !

API REST

Manipuler des ressources

- L'API REST est une architecture qui repose sur le protocole **HTTP**, c'est-à-dire que l'API REST permet d'accéder à un service web via les 4 opérations de HTTP, ou verbes (**POST**, **GET**, **PUT**, **DELETE**).
- Ces opérations ont donc 4 fonctions résumées par l'acronyme **CRUD** (Create, Read, Update, Delete)

Les clés d'une API REST

- L'architecture REST s'articule autour de 4 principes clés :
 - une ressource distribuée sur un serveur distant (**mécanisme client-serveur**)
 - un identifiant de la ressource (les **URL** et protocole **HTTP**)
 - des « verbes » **HTTP** permettant d'agir sur la ressource (**GET**, **POST**, **PUT**, **DELETE** non inclus dans l'URI)
 - une représentation de la ressource (la réponse peut avoir plusieurs représentations possibles : **HTML**, **JSON**, **XML**).

API REST

les codes de retour des verbes HTTP !

- Ces codes sont répartis dans les cinq catégories suivantes :
 - **1xx** : **informatif** : communique une information au client
 - **2xx** : **succès** : indique que la demande du client a été acceptée avec succès
 - **3xx** : **redirection** : indique que le client doit prendre des mesures supplémentaires pour terminer sa demande
 - **4xx** : **erreur du client** : indique une erreur de la part du client
 - **5xx** : **erreur du serveur** : indique une erreur de la part du serveur.

TESTER UNE API REST

- Préférez des API certifiées ou sécurisées (HTTPS).
- S'assurer de sa provenance.
- Postman <https://www.postman.com/>

GraphQL

Graph Query Language

GraphQL

Définition

- **GraphQL** est un langage de requête et un runtime pour les API, qui a été développé par Facebook en 2012.
- Publié open-source en 2015.
- Il a été conçu pour résoudre les problèmes rencontrés par les API REST traditionnelles, notamment la surcharge de données et la complexité accrue pour les consommateurs d'API.
 - Alternative aux API REST.

Principe

- Avec **GraphQL**, les développeurs peuvent spécifier exactement les données dont ils ont besoin pour leur application, sans avoir à recevoir plus de données que nécessaire.
- Cela permet de minimiser les requêtes réseau, ce qui peut améliorer les performances de l'application et réduire les coûts de bande passante.
- GraphQL est un langage de requêtes de données pour API. QL, comme dans SQL, veut dire Query Language.
- GraphQL permet de manipuler de la donnée de façon simple, flexible et très précise.
- GraphQL n'est pas un langage de programmation ou un framework.
- GraphQL est une spécification pour implémenter ton API.

GraphQL

Avantages

- Syntaxe simple et cohérente pour les requêtes.
- Obtention détaillée des schémas d'API GraphQL
- Flexible avec de nombreux types de données, notamment les données structurées, les images et les fichiers audio et vidéo.
- Il peut également être utilisé avec de nombreux langages de programmation, y compris JavaScript, Ruby, Java et Python.

Site

- <https://graphql.org/>

Conclusion

- Les API REST, SOAP et GraphQL sont tous des types d'API populaires utilisés pour connecter les applications web et les systèmes.
- Chacun d'entre eux offre ses propres avantages et inconvénients en termes de performance, de sécurité, de flexibilité et de fonctionnalités.
- Il est donc important de comprendre les différences clés entre ces types d'API afin de choisir celui qui convient le mieux à votre projet.

MERCI !

Jérôme BOEBION
Concepteur Développeur d'Applications
Version 2 - révision 2024

Afpa
se former, avancer

