

Une api c'est ...

Automate programmable industriel

[...] est un ensemble d'éléments organisé en fonction d'un but défini par des fonctions et des Constituants.

- A la fin des années 60, Un fabricant américain de voitures décide de remplacer les systèmes de commande à base de logique câblée (relais électrique) par une logique programmée.

- Le premier API, modèle 084, a été inventé par Dick Morley en 1969



... au delà de la blague l'API des automates n'est pas très éloigné d'un API informatique.

Automate programmable industriel

<https://www.scribd.com/document/446061670/cours-API>

https://fr.wikipedia.org/wiki/Automate_programmable_industriel

L'API (pour Application Programming Interface) informatique ou interface de programmation applicative est avant tout une interface (au sens objet)

Une API n'est pas :

- Un logiciel : un logiciel n'est pas une API (même s'il peut se présenter sous la forme d'une API pour faciliter l'utilisation de ses fonctionnalités).
- Une interface utilisateur : une interface utilisateur n'est pas une API (mais elle peut s'exécuter sur une interface utilisateur).
- Un serveur : un serveur n'est pas une API (mais il peut héberger une ou plusieurs API qui fournissent les données et fonctions mises à disposition par le serveur)

Encore un peu de vocabulaire lorsqu'on appelle une api on dit qu'elle est consommée, ceci peut aider à se rappeler qu'un api est "stateless", sans état.

Un api serveuse vous amène des frites, mais vous consommez de suite vous ne pouvez pas demander de la mayo avec ça, parce que vous avez déjà consommé les frites, la serveuse ne sait donc pas à quoi vous faites référence

Un facteur important pour une interface est leur stabilité.

Il est important d'indiquer le niveau de stabilité d'une api à une personnes qui l'utilise, les notions de beta, de dépréciation prennent ici toute leur importance (mais nous y reviendrons)

L'autre est la documentation !

Elle complète l'interface et doit suivre ses évolutions.

On documente :

- Les routes / requete
- La réponses

L'histoire des api :

La première mention d'une API en tant que telles est dans un document intitulé "Data structures and techniques for remote computer graphics" (structures de données et techniques pour l'infographie à distance) présenté à une conférence en 1968.

Il présentait une couche d'abstraction pour gérer via une interface des éléments d'affichage, en s'affranchissant de la connaissance du matériel.

En 1974 les API sont introduites en base de données.

Avec l'avènement du WEB et des programmes décentralisés les API sont devenus un élément incontournable de la façon dont une architecture logicielle est construite.

Réduite à ça plus simple expression une API est ce qui permet à des systèmes disparates de communiquer de façon simple et stable.

Les systèmes d'exploitation sont aussi une sorte d'api (Cocoa pour Apple, Windows API pour ... ben windows), elles permettent d'écrire des programmes en utilisant les composants du système d'exploitation.

On peut distinguer si on veut chercher la petite bête une API d'un ABI (application binary interface) la dernière étant binaire, la première un code.

Mais finalement elle remplissent le même rôle : offrir une interface pour interagir avec un composant fermé.

En web l'une des API les plus utilisées est le DOM on la manipule avec Javascript

DOM

<https://dom.spec.whatwg.org/#what>

exemple d'interface pour Document :

<https://dom.spec.whatwg.org/#interface-document> avec en particulier document.getElementById, qui renvoie un element :

<https://dom.spec.whatwg.org/#element>

Bien que ce doc soit une norme, c'est une norme "vivante" et chaque navigateur en a une implémentation différente... D'ou les problèmes de compatibilité.

Différentes API :

- REST : L'api REST (Representational State Transfer), qui pour le web actuellement est souvent en JSON, et donc qualifiée de REST/JSON est peut être la plus récente architecture d'api mais ce n'est pas la seule parmi les autres on a :
- REST/XML : comme la précédente mais avec des réponses XML, voir peut répondre de deux façon
- SOAP (SIMPLE OBJECT ACCESS PROTOCOL) : la aussi on du xml mais normé d'une façon spécifique et propriétaire
<https://www.w3.org/2003/05/soap-envelope/>
- JSON/RPC : RPC = remote procedural call consiste à construire une api pour qu'elle appelle des méthodes distante (ou d'un interface dédiée, win32) on est donc plus dans une logique d'action que d'états
- XML/RPC : comme précédemment... mais en xml
- ET pourquoi pas d'autre ? Rien n'empêche de définir un api... REST/CSV par exemple

Définition API

https://fr.wikipedia.org/wiki/Interface_de_programmation

<https://www.ibm.com/docs/en/i/7.3?topic=interfaces-apis-overview>

<https://developer.ibm.com/articles/introduction-apis-and-messaging/>

<https://developer.ibm.com/videos/an-introduction-to-rest-apis/>

Dans tous les cas une API s'occupe de données pas de look.

Avantages :

- Efficacité : Soulager le front, facilement "scalable"
- Flexible : obeit a une interface, augmenter cette interfcace rend l'api flexible
- Centraliser les données Normalise
- Automatisation
- Simplification
- Intégration

Désavantages :

- Pas propriétaire / maître des données / dépendance
- Sécurité

- Coût : développement plus coûteux et en maintenance plus coûteuse cf réponse de [10:10] GUILLIN David desavantages : la multiplicité des fonctions création des routes

Avantage REST :

- Cacheable

Désavantage REST

- Un peu plus lent qu'un accès direct puisque communication

Concentrons nous sur le REST : une api rest est

- Simple
- sans état
- Cachable
- *client -serveur*

(et un peu plus en RESTful)

Pour le restful :

<https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces#soap-vs-rest>

<https://restfulapi.net/>

- En couche
- Code à la demande

Tests pour API :

<https://addons.mozilla.org/fr/firefox/addon/rest-client-apishub/> (mais pas trop mis à jour)

<https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdgggehcdcbncdddomop> (mais déprécié)

Le mieux à mon sens :

Postman <https://www.postman.com/downloads/>

Exercice :

Utiliser des api pour avoir la météo à mydigitalschool (1 Rue Pierre de Maupertuis. 35170 Bruz)

Conversion de l'adresse en latitude longitude :

<https://www.geoapify.com/>

Avec lat et long :

<https://www.weatherapi.com/>

Pour avoir la météo dans la journée

TDD : test driven developpment

On écrit les test d'abord, puis on implémente le code

Api appelé sur : <url>/address=texte

En GET

Doit rendre la température actuellement à l'adresse donnée sur le format JSON avec

{“temperature”:”XX”} pour XX est un entier

Si address est présent mais vide, renvoyer une erreur

Interface = contrat

API <-> appli mobile
<-> site web
<-> application tierce qui consomme les données

Créer des entrée = Create

Relire ces entrée = read

Update = mettre à joUr

Détruire les données = delete

Dans une API REST on va utiliser des “verbes” / méthodes HTTP :

- Create = post
- Read = get
- Update = post / put / patch
- Delete = delete

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

<https://github.com/public-apis/public-apis>

<https://rapidapi.com/hub>

<https://octopush.com/api-sms-octopush/>

Révision pour Jeff

Choisir une api publique pour faire des tests (par groupe ?)

Implémenter l'exercice postman en node

<http://localhost:8081/?address=1 Rue Pierre de Maupertuis. 35170 Bruz, france>

Salomé

- Api spaceX
- Parcours exemple : équipage,
Chaque crew member a une id qu'on peut réutiliser
- Puis un dragon
Puis un dragon avec l'id

Arthur

- Spotify avec Authentification
- Recupère une playlist pour un user
- Token en environnement, puis les titres avec un identifiant
- Pour les tracks sauvegardées : on a pas la permission dans l'offre gratuite pour avoir les infos
- Première implémentation en python pour le wanted

David (Massé)

- Api imdb api documentation
- Renseigner des information dans les headers
- On opticien un film, puis le cast, puis l'acteur depuis la filmographie

Mickaël

- Api sport, football
- Avoir une saison
- Les équipe
- Avoir infos d'une équipe
- Puis les stats d'une équipe sur une saison

Benjamin

- Api pokemon
- Avoir la liste des types
- Puis les pokemon d'un type
- Puis un pokemon pour ces abilités

Nicolas

- Api RIOT pour LoL
- Clef de 24h
- On trouve un pseudo, on récupère un id

- On peut trouver les matchs joués
- On peut avoir les infos sur un match
- Et retrouver les participants
- Solution alternative : insomnia api