APIWORDPRESS

SLIDE 1

TITRE

SLIDE 2

API (Interface de programmation d'application) est un ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications.

Elles permettent à notre produit ou service de communiquer avec d'autres produits et services sans connaitre les détails de leur mise en œuvre. Elles simplifient le développement d'applications et font gagner du temps et de l'argent.

Elle sert à intégrer des fonctionnalités dans nos applications ou sites Web. Elles se mettent à jour automatiquement.

SLIDE 3

API SOAP (Simple Object Access Protocol) : c'est une requête Post qui utilise le format XML (on appelle des liens)

API REST (Representational State Transfer) : c ‘est un style d’architecture. On utilise des fichiers en Json pour appeler les données

SLIDE 4

Les differents verbes/méthodes HTTP

HTTP définit un ensemble de **méthodes de requête** qui indiquent l'action que l'on souhaite réaliser sur la ressource indiquée. Ses requêtes sont souvent appelées verbes HTTP.

Chacun d'eux implémente une sémantique différente mais certaines fonctionnalités courantes peuvent être partagées par différentes méthodes (e.g. une méthode de requête peut être sûre (*safe*), idempotente ou être mise en cache (*cacheable*)).

Les différentes méthodes sont :

GET

La méthode GET demande une représentation de la ressource spécifiée. Les requêtes GET doivent uniquement être utilisées afin de récupérer des données.

HEAD

La méthode HEAD demande une réponse identique à une requête GET pour laquelle on aura omis le corps de la réponse (on a uniquement l'en-tête).

POST

La méthode POST est utilisée pour envoyer une entité vers la ressource indiquée. Cela entraîne généralement un changement d'état ou des effets de bord sur le serveur.

PUT

La méthode PUT remplace toutes les représentations actuelles de la ressource visée par le contenu de la requête.

DELETE

La méthode DELETE supprime la ressource indiquée.

CONNECT

La méthode CONNECT établit un tunnel vers le serveur identifié par la ressource cible.

OPTIONS

La méthode OPTIONS est utilisée pour décrire les options de communications avec la ressource visée.

TRACE

La méthode TRACE réalise un message de test aller/retour en suivant le chemin de la ressource visée.

PATCH

La méthode PATCH est utilisée pour appliquer des modifications partielles à une ressource.

Les différents codes de réponses HTTP

Ces codes sont des numéros à 3 chiffres, la plupart des codes correspondent chacun à un type d'erreur, quelques-uns correspondent à un type de succès. Ces codes permettent aux logiciels client HTTP de déterminer automatiquement si une requête a réussi, et sinon de connaître le type d'erreur.

Le premier chiffre est utilisé pour spécifier une des cinq catégories de réponse (informations, succès, redirection, erreur client et erreur serveur).

Les codes les plus courants sont :

- 200 : succès de la requête ;
- 301 et 302 : redirection, respectivement permanente et temporaire ;
- 401 : utilisateur non authentifié ;
- 403 : accès refusé ;
- 404 : ressource non trouvée ;
- 500, 502 et 503 : erreurs serveur ;
- 504 : le serveur n'a pas répondu.

SLIDE 5

Dans une application Web, vous pouvez simplement interagir avec une API Web coté client **en appelant les requêtes HTTP Get, Put, Post et Delete dans un Script JavaScript ou en utilisant jQuery**.

SLIDE 6

Sécuriser une API

- Il faut identifier les points faibles du système (API, vérifier si les droits d'accès, mettre des règles plus strictes pour accéder au Json, utilisé des limites de débit pour assurer la protection du back end API
- Mettre en place des Token (sécurité de connexion)
- Recourir au cryptage des données (cache les données aux utilisateurs/appareils qui n'y sont pas autorisé à déchiffrer les données
- Utiliser un OAUTH et OPENID connect:

  OAUTH permet de définir comment on accéde au token

  OPENID permet de vérifier l'identifier de qui accède au token,

- Créer une passerelle API GATEWAY (permet de créer une passerelle entre l'API et le client ça joue le rôle de proxy inversé

SLIDE 7

GraphQL = c'est un langage de requête et un environnement d'exécution coté serveur qui permet de simplifier l'utilisation des API.

Utilisé à la place de REST, GraphQL permet aux développeurs de créer des requêtes qui extraient les données de plusieurs sources à l'aide d'un seul appel d'API.

La maintenance est simplifiée avec garphQL, on peut ajouter ou retirer des champs sans perturber les requêtes existantes

Avantages et inconvénients de GraphQL

Avantages:

- Un schéma GraphQL définit une source unique de vérité dans une application GraphQL. Il permet à une entreprise de fédérer l'ensemble de son API.

- Les appels GraphQL sont gérés par un seul aller-retour. Les clients obtiennent précisément ce qu'ils ont demandé, rien de plus.
- Les types de données sont définis rigoureusement, ce qui limite les problèmes de communication entre le client et le serveur.
- GraphQL est introspectif. Un client peut demander une liste des types de données disponibles. Cette fonction est très utile pour la génération automatique de documents.
- GraphQL permet de faire évoluer l'API d'une application sans dégrader les requêtes existantes.
- Il existe de nombreuses extensions GraphQL Open Source qui proposent des fonctions indisponibles dans les API REST.

- GraphQL n'impose aucune architecture d'application spécifique. Il peut être mis en œuvre sur une API REST existante et fonctionner avec les outils de gestion des API existants.

Inconvéniants:

- Les développeurs qui travaillent habituellement avec des API REST devront apprendre à utiliser GraphQL.
- GraphQL déplace l'essentiel de la charge de travail d'une requête de données du côté serveur, ce qui complique la tâche des développeurs serveur.

- Selon la mise en œuvre, les stratégies de gestion des API nécessaires pour GraphQL peuvent différer de celles des API REST, notamment en matière de limites de débit et de tarifs.
- La mise en cache est plus complexe qu'avec les API REST.
- Les équipes chargées de la maintenance des API doivent écrire un schéma GraphQL supplémentaire compatible avec les opérations de maintenance.