

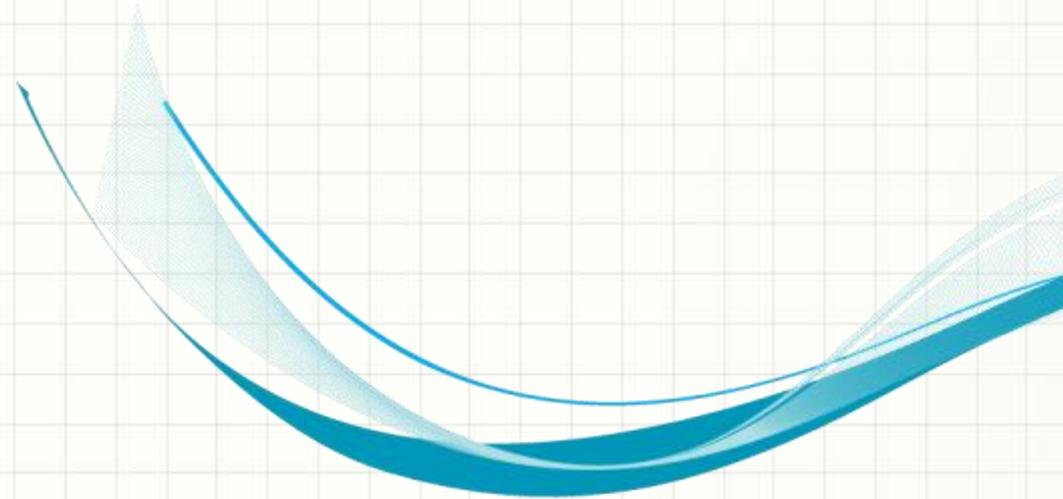


PRÉSENTATION DE RESTFUL

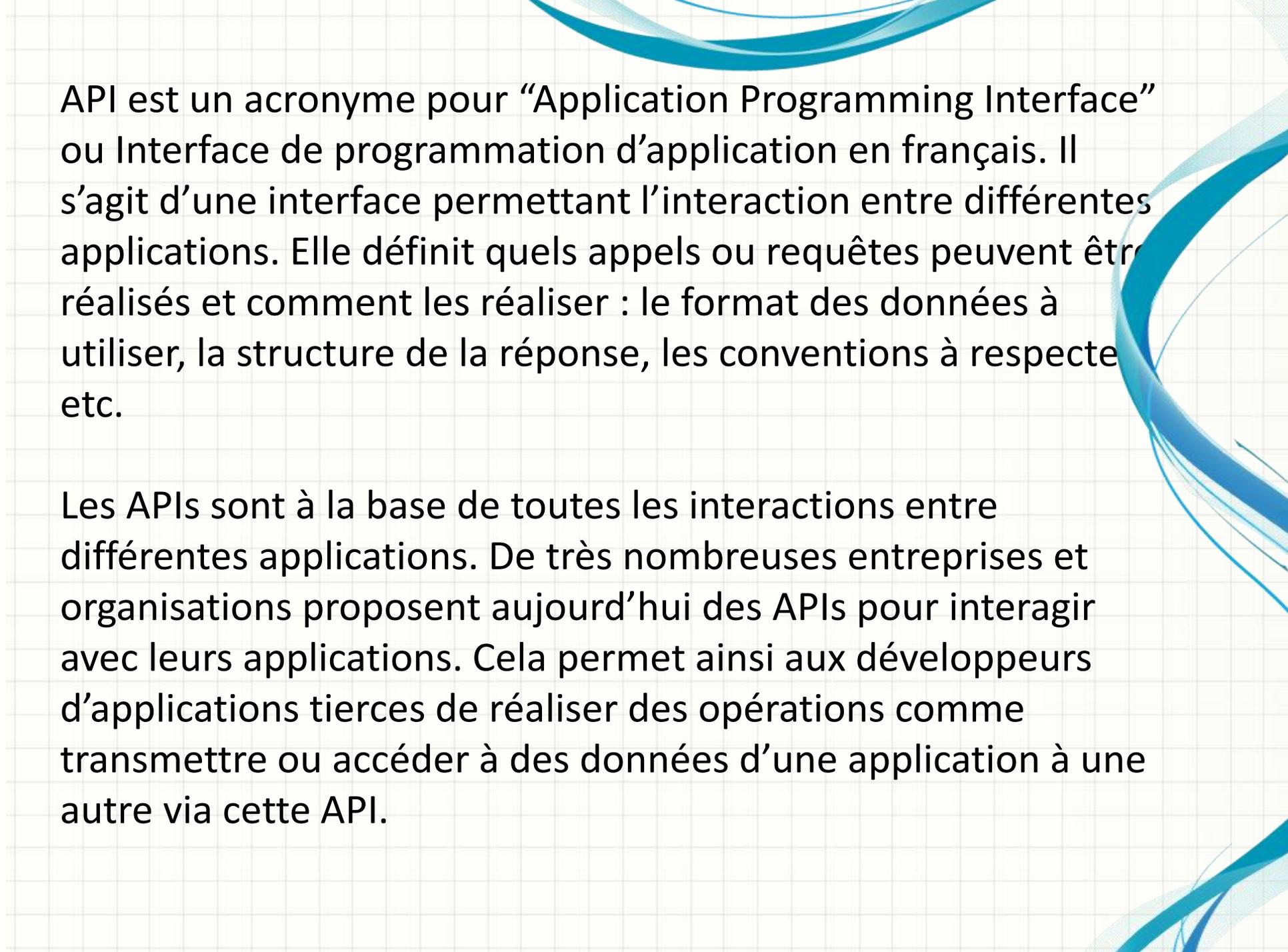
Tahraoui fadda

Introduction

- Rest est une architecture créée pour interagir avec les systèmes distribués. Il s'agit d'une architecture orienté ressource créée en 2000 par Roy Fielding pendant sa thèse de doctorat.
- Roy Fielding est le principal auteur de la spécification HTTP, aussi membre fondateur de la fondation apache et sprincipal développeur du serveur web Apache..
- Dans cet projet, je vous explique tout sur les API REST, leur utilité, leur différence avec d'autres standards et leurs principes de fonctionnement.

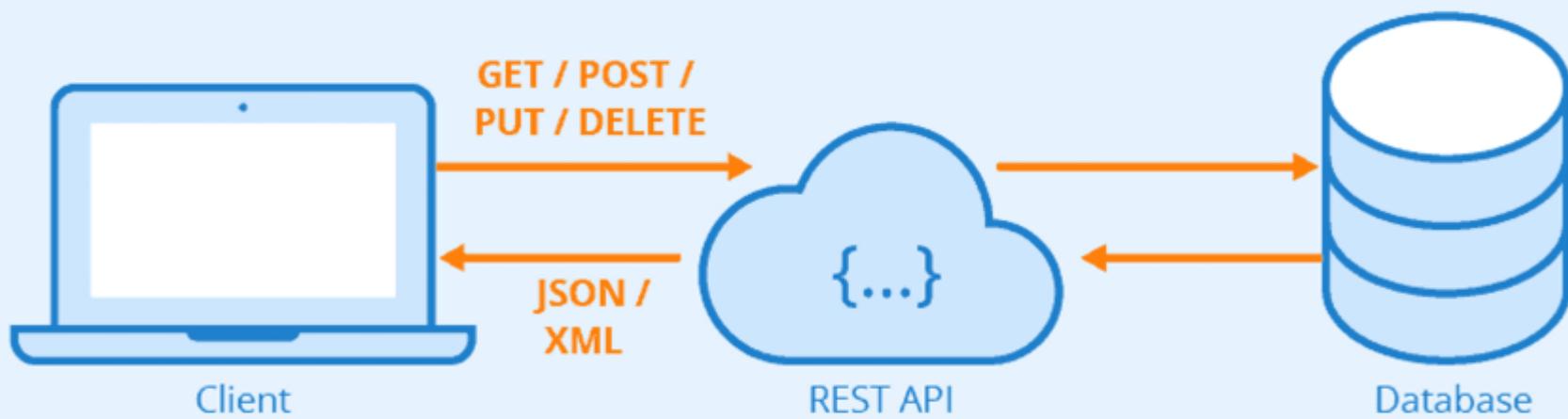


Qu'est-ce qu'une API
RESTful ?



API est un acronyme pour “Application Programming Interface” ou Interface de programmation d’application en français. Il s’agit d’une interface permettant l’interaction entre différentes applications. Elle définit quels appels ou requêtes peuvent être réalisés et comment les réaliser : le format des données à utiliser, la structure de la réponse, les conventions à respecter etc.

Les APIs sont à la base de toutes les interactions entre différentes applications. De très nombreuses entreprises et organisations proposent aujourd’hui des APIs pour interagir avec leurs applications. Cela permet ainsi aux développeurs d’applications tierces de réaliser des opérations comme transmettre ou accéder à des données d’une application à une autre via cette API.





Qu'est-ce que
le standard
d'API RESTful ?



REST (pour Représentationnel State Transfer) est un type d'architecture d'API permettant de construire des applications (Web, Intranet, Web Service), qui fournit un certain nombre de normes et de conventions à respecter pour faciliter la communication entre applications. L'architecture REST utilise les spécifications originelles du protocole HTTP, plutôt que de réinventer une surcouche (comme le font SOAP ou XML-RPC par exemple).

Les APIs qui respectent le standard REST sont appelées API REST ou API RESTful.

Les principes d'une architecture REST

1

- La séparation entre client et serveur

2

- L'absence d'état de sessions (stateless)

3

- L'uniformité de l'interface

4

- La mise en cache

5

- L'architecture en couches

6

- Le code à la demande

Remarque:

Une API REST ne peut être qualifiée de RESTful si elle ne respecte pas les six contraintes, mais on peut tout de même la qualifier d'API REST si elle n'enfreint que deux ou trois principes. REST est sans doute le standard le plus utilisé pour concevoir des architectures d'API

Les avantages des API RESTful:

Capacité de mise à l'échelle: Les systèmes qui implémentent des API REST sont capables de se mettre à l'échelle efficacement, car REST optimise les interactions client-serveur.

Flexibilité: Les services Web RESTful prennent en charge la séparation totale client-serveur. Ils simplifient et découplent les différents composants du serveur afin que chaque partie puisse évoluer indépendamment.

Indépendance: Les API REST sont indépendantes de la technologie utilisée. Vous pouvez écrire des applications client et serveur dans divers langages de programmation sans affecter la conception de l'API. Vous pouvez également changer la technologie sous-jacente de part et d'autre sans affecter la communication.

Comment fonctionnent les API RESTful ?

- La fonction de base d'une API (interface de programme d'application) RESTful est la même que celle de la navigation sur Internet. Le client contacte le serveur en utilisant l'API lorsqu'il a besoin d'une ressource. Les développeurs d'API expliquent comment le client doit utiliser l'API REST dans la documentation de l'API de l'application serveur. Ce sont les étapes générales de tout appel d'API REST :
- Le client envoie une requête au serveur. Le client suit la documentation de l'API pour formater la requête de manière à la rendre compréhensible pour le serveur.
- Le serveur authentifie le client et confirme que ce dernier a le droit de faire cette demande.
- Le serveur reçoit la demande et la traite en interne.
- Le serveur renvoie une réponse au client. La réponse contient des informations qui indiquent au client si la demande a abouti. La réponse comprend également toute information demandée par le client.
- Les détails de la demande et de la réponse de l'API REST varient légèrement en fonction de la façon dont les développeurs de l'API conçoivent cette dernière.

CONCLUSION

Inconvénients

- La sécurité est inexistante - Utilisation d'HTTPS + Authentification .
- Le client doit conserver des données localement:
 - Consommation en bande passante - Données de session .
 - Peut être problématique pour les Smartphones .

Résumé

- REST un style d'architecture permettant de construire des applications (Web, Intranet, Web Service). Il s'agit d'un ensemble de conventions et de bonnes pratiques à respecter et non d'une technologie à part entière.
- L'architecture REST utilise les spécifications originelles du protocole HTTP, plutôt que de réinventer une surcouche (comme le font SOAP ou XML-RPC par exemple).
- **REST n'est pas** : un standard , un protocole et un format .
- **REST c'est** : un bonne pratique aussi pas de spécifications de la W3C et un type d'architecture pour les systèmes distribués .

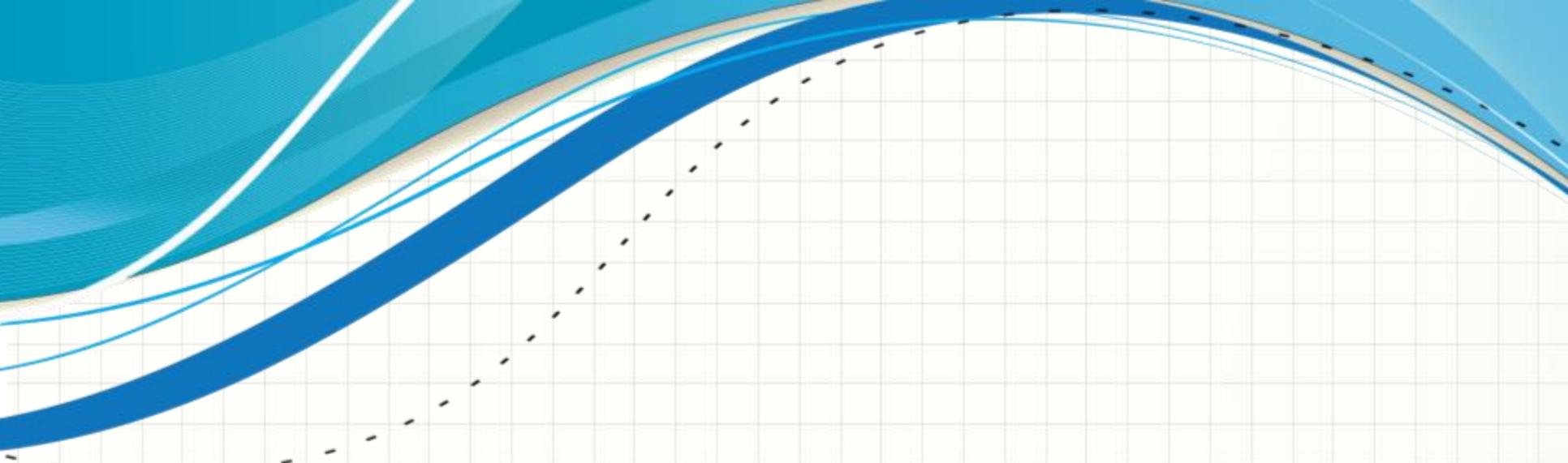
Quiz?

- Vous souhaitez tester vos connaissances sur les API Rest ?

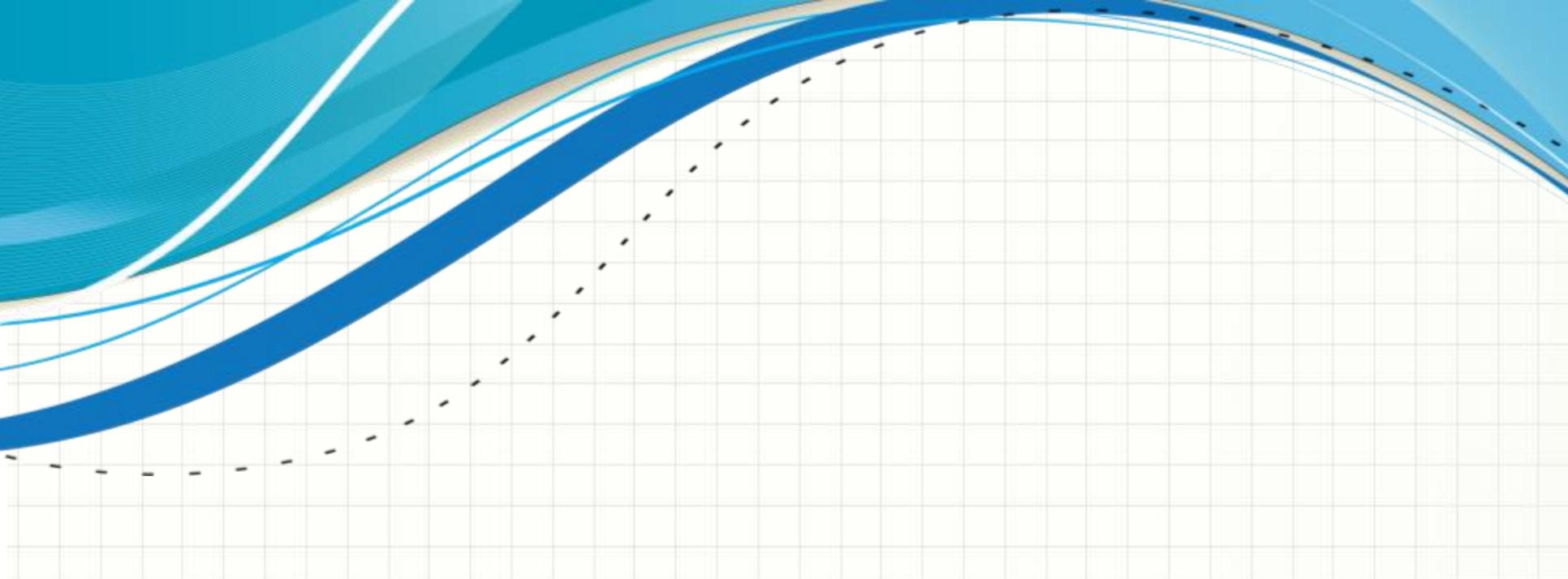
WeLoveDevs.com met a disposition gratuitement un test technique avec son corrigé sur les API REST

Ressources

- <Texte du site intranet ici>
<[lien https://welovedevs.com/fr/articles/api-rest/](https://welovedevs.com/fr/articles/api-rest/) ici>
- <Texte de support de lecture supplémentaire ici>
<[lien https://g.co/kgs/zK8B4A](https://g.co/kgs/zK8B4A) ici>



**VOUS AVEZ DES
QUESTIONS ?**



MERCI POUR VOTRE ATTENTION