

# Quelques notions et outils indispensables au dev Web

---

Lionel KITIHOUN

Web et HTTP

REST et OpenAPI

Outils pour tester et développer des APIs

Git

# Web et HTTP

---

# Web et HTTP

---

## Généralités

Système hypertexte public fonctionnant sur Internet. Il permet de consulter, avec un navigateur, des pages ou médias accessibles sur des sites<sup>1</sup>.

---

1. [https://fr.wikipedia.org/wiki/World\\_Wide\\_Web](https://fr.wikipedia.org/wiki/World_Wide_Web)

## Autres services fonctionnant grâce à Internet

- Transfert de fichiers (FTP)
- Messagerie électronique (SMTP)
- VoIP
- etc.

- Définition ?

---

2. [https://fr.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

- Définition ?
- Hypertext Transfer Protocol.

---

2. [https://fr.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol)



- Définition ?
- Hypertext Transfer Protocol.
- Protocole de communication client-serveur développé pour le World Wide Web<sup>2</sup>.

---

2. [https://fr.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

- Définition ?
- Hypertext Transfer Protocol.
- Protocole de communication client-serveur développé pour le World Wide Web<sup>2</sup>.
- HTTPS est la variante sécurisée par l'usage des protocoles Transport Layer Security (TLS).

---

2. [https://fr.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

- Tout développeur se doit de le connaître.
- Parce qu'il est fort probable qu'un jour vous devrez vous y frotter.
- Web, mobile, système, IoT...

- Tout développeur se doit de le connaître.
- Parce qu'il est fort probable qu'un jour vous devrez vous y frotter.
- Web, mobile, système, IoT...

- Tout développeur se doit de le connaître.
- Parce qu'il est fort probable qu'un jour vous devrez vous y frotter.
- Web, mobile, système, IoT...

# Web et HTTP

---

## Méthodes

Le protocole fonctionne sur la base de **verbes** ou **méthodes**, qui indiquent le type de requête initiée par le client.

Le protocole fonctionne sur la base de **verbes** ou **méthodes**, qui indiquent le type de requête initiée par le client.

Il y en a plusieurs, notamment :

- GET
- HEAD
- POST
- PUT
- PATCH
- DELETE
- OPTIONS



Un serveur met à disposition de ses clients des ressources, accessibles via une URL. Au début du web, les ressources étaient généralement des pages HTML, des documents, des images ou des vidéos.

Avec l'évolution de web, les ressources peuvent représenter beaucoup d'autres choses, les entités stockées dans une base de données, des relevés d'impôts, des données de natures diverses.

C'est la méthode la plus courante pour demander la représentation d'une ressource. Elle ne doit entraîner aucune modification côté serveur.

Cette méthode ne demande que des informations (métadonnées) sur la ressource, sans demander la ressource elle-même.

Utile par exemple pour la mise en cache, pour savoir si une ressource a été modifiée par exemple.

# POST

Cette méthode est utilisée pour transmettre des données en vue d'un traitement.

# PUT

Cette méthode est utilisée pour remplacer intégralement une ressource.

# PATCH

Cette méthode est utilisée pour modifier partiellement une ressource.

# DELETE

Comme son nom ne l'indique pas, cette méthode est utilisée pour supprimer une ressource sur le serveur.

Bien entendu, le serveur devra s'assurer que le client qui requiert cette action en a le droit.

Cette méthode permet d'obtenir les options de communication d'une ressource ou du serveur en général.



# Web et HTTP

---

## Codes de réponse HTTP

Les codes de statut de réponse HTTP sont utilisés par le serveur pour indiquer si une requête HTTP a été exécutée avec succès ou non.

Les codes de réponse sont regroupées en cinq classes. Elles apparaissent dans l'entête de la réponse.

- Les réponses informatives (100 - 199).
- Les réponses de succès (200 - 299).
- Les messages de redirection (300 - 399).
- Les erreurs du client (400 - 499).
- Les erreurs du serveur (500 - 599)

- Les réponses informatives (100 - 199).
- Les réponses de succès (200 - 299).
- Les messages de redirection (300 - 399).
- Les erreurs du client (400 - 499).
- Les erreurs du serveur (500 - 599)

- Les réponses informatives (100 - 199).
- Les réponses de succès (200 - 299).
- Les messages de redirection (300 - 399).
- Les erreurs du client (400 - 499).
- Les erreurs du serveur (500 - 599)

- Les réponses informatives (100 - 199).
- Les réponses de succès (200 - 299).
- Les messages de redirection (300 - 399).
- Les erreurs du client (400 - 499).
- Les erreurs du serveur (500 - 599)

- Les réponses informatives (100 - 199).
- Les réponses de succès (200 - 299).
- Les messages de redirection (300 - 399).
- Les erreurs du client (400 - 499).
- Les erreurs du serveur (500 - 599)

- Les réponses informatives (100 - 199).
- Les réponses de succès (200 - 299).
- Les messages de redirection (300 - 399).
- Les erreurs du client (400 - 499).
- Les erreurs du serveur (500 - 599)



# Web et HTTP

---

## Entêtes HTTP

Elles permettent au client et aux serveurs d'ajouter des informations supplémentaires aux messages HTTP.

Certaines entêtes sont propres aux requêtes, d'autres aux réponses. D'autres peuvent être présentes dans les deux.

- **Content-Type**

Utilisé pour indiquer le type MIME du corps du message.

- **Content-Length**

Utilisé pour indiquer la longueur (en octets) du corps du message.

- **Accept**

Utilisé par le client pour indiquer les types MIME qu'il souhaite pour la réponse.

- **Location**

Utilisé par le serveur pour indiquer une redirection.

- **Host**

Utilisé par le client pour indiquer à quelle site la requête est envoyée.

Plusieurs sites ou applications peuvent être hébergées sur un même serveur.

Cet entête est donc très important.

- **Authorization**

Utilisé pour l'authentification. Utilisé par exemple par les web services et les APIs web pour la gestion des accès.



- **Cookie, Set-Cookie**

Pour la gestion des cookies (Miam).

# Web et HTTP

---

## Cookies

Un cookie est une information générée par un serveur web et stocké chez le client suite à une requête.

Ils sont utilisés à des fins de personnalisation du contenu, d'identification des utilisateurs, pour la publicité et le suivi des individus sur le web.

HTTP est un protocole sans état. Le serveur n'a aucun moyen de savoir que certaines requêtes proviennent d'un même client.

Les cookies étaient utilisés par les sites pour faire le lien entre les requêtes successives d'un utilisateur. Le serveur ajoute des cookies à une réponse et le client renvoie les mêmes cookies lors d'une autre requête. Le serveur peut ainsi les reconnaître.

Malheureusement, l'utilisation des cookies a été pervertie.  
Jetez unœil aux cookies stockés dans votre navigateur. ;)

# Web et HTTP

---

## Faire des requêtes HTTP

# S'amuser à faire des requêtes HTTP

Il existe de nombreux outils pouvant aider à lancer des requêtes HTTP vers un serveur.

- **Votre navigateur web**

Vous déclenchez un nombre incalculable de requêtes quand vous naviguez.

- Les outils destinés aux développeurs

- Postman
- Insomnia

- Les outils en ligne de commande

- telnet
- curl (la référence)
- netcat (ncat)
- httpie

# S'amuser à faire des requêtes HTTP

Il existe de nombreux outils pouvant aider à lancer des requêtes HTTP vers un serveur.

- **Votre navigateur web**

Vous déclenchez un nombre incalculable de requêtes quand vous naviguez.

- **Les outils destinés aux développeurs**

- Postman
- Insomnia

- **Les outils en ligne de commande**

- telnet
- curl (la référence)
- netcat (ncat)
- httpie



# S'amuser à faire des requêtes HTTP

Il existe de nombreux outils pouvant aider à lancer des requêtes HTTP vers un serveur.

- **Votre navigateur web**

Vous déclenchez un nombre incalculable de requêtes quand vous naviguez.

- **Les outils destinés aux développeurs**

- Postman
- Insomnia

- **Les outils en ligne de commande**

- telnet
- curl (la référence)
- netcat (ncat)
- httpie

# S'amuser à faire des requêtes HTTP

Il existe de nombreux outils pouvant aider à lancer des requêtes HTTP vers un serveur.

- **Votre navigateur web**

Vous déclenchez un nombre incalculable de requêtes quand vous naviguez.

- **Les outils destinés aux développeurs**

- Postman
- Insomnia

- **Les outils en ligne de commande**

- telnet
- curl (la référence)
- netcat (ncat)
- httpie

## Quelques requêtes HTTP

- GET
- HEAD
- POST

# REST et OpenAPI

---

## REST

- REpresentational State Transfer

## REST

- Style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web<sup>3</sup>

---

3. [https://fr.wikipedia.org/wiki/Representational\\_state\\_transfer](https://fr.wikipedia.org/wiki/Representational_state_transfer)

## REST

- Les services web REST permettent aux systèmes effectuant des requêtes de manipuler des ressources web via leurs représentations textuelles à travers un ensemble d'opérations uniformes et prédéfinies sans état.

REST est souvent utilisé avec HTTP.

Un mapping est donc utilisé entre les méthodes HTTP et les opérations CRUD sur les entités manipulées.

- GET
- POST
- PUT/PATCH
- DELETE



## Exemple de ressources REST

- GET /persons
- POST /persons
- GET /persons/1
- PUT /persons/1
- PATCH /persons/1
- DELETE /persons/1

Standard utilisé pour décrire les opérations offertes par une API ou web services.

Le format utilisé pour la description est le YAML (ou le JSON).

# Example

```
openapi: 3.1.0
info:
  title: Tic Tac Toe
  description: This API allows to play Tic Tac Toe.
  version: 1.0.0
paths:
  # Whole board operations
  /board:
    get:
      summary: Get the whole board
      description: Retrieves the current state of the board and the winner.
      responses:
        "200":
          description: "OK"
          content:
...

```

# Outils pour tester et développer des APIs

---



POSTMAN





# Git

---





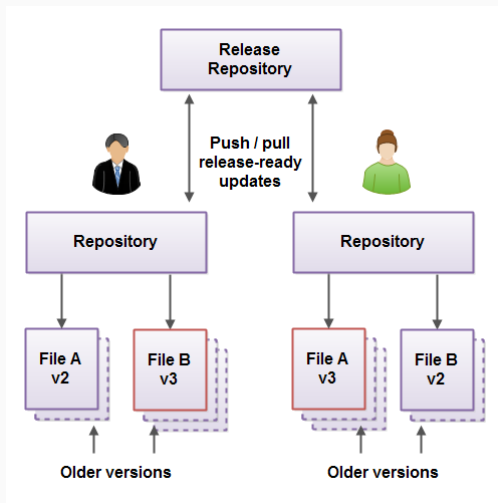
- Logiciel de gestion de versions décentralisé
- Logiciel libre
- Créé par Linus Torvalds
- Site web : <https://git-scm.com>

- Logiciel de gestion de versions décentralisé
- Logiciel libre
- Créé par Linus Torvalds
- Site web : <https://git-scm.com>

- Logiciel de gestion de versions décentralisé
- Logiciel libre
- Créé par Linus Torvalds
- Site web : <https://git-scm.com>

- Logiciel de gestion de versions décentralisé
- Logiciel libre
- Créé par Linus Torvalds
- Site web : `https://git-scm.com`

# Fonctionnement



- **git init**

Pour initialiser un nouveau dépôt.

- **git add**

Pour enregistrer des changements.



- **git commit**

Utilisé valider les changements.

- **git push**

Pour pousser son travail vers le dépôt central.

- **git pull**

Pour se synchroniser avec le dépôt central.

- **git clone**

Pour faire une copie d'un dépôt.

- **GitHub**

Propriétaire, mais accessible

# GitHub

- **GitLab**  
Opensource



On parle de **pull request** lorsqu'un développeur demande que les changements ou ajouts qu'il a effectués sur sa copie soient intégrés dans le dépôt principal.

Le propriétaire du dépôt peut accepter (après validation des modifications) ou non la requête.

Le rejet d'un **pull request** donne parfois naissance à un **fork**.