

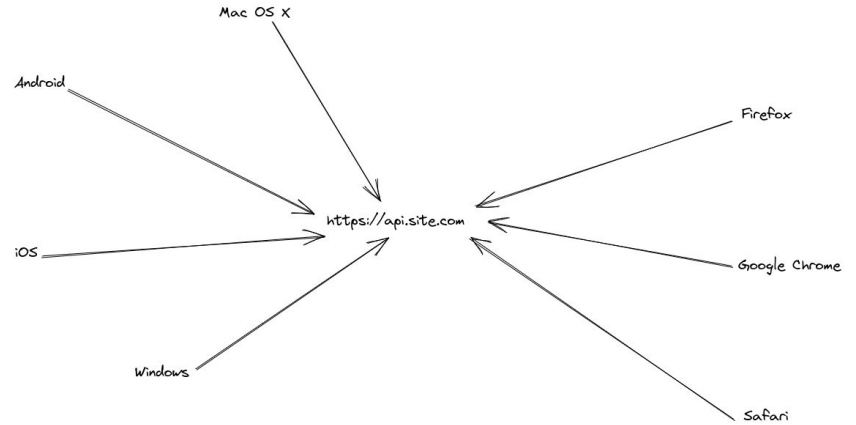
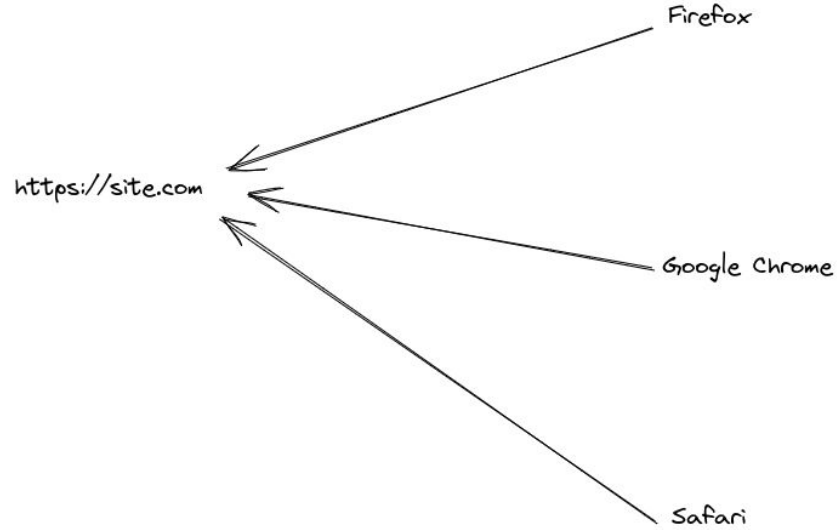
API HTTP

Amin NAIRI <anairi@esgi.fr>

Pré-requis

- [Éditeur de texte](#)
- [Navigateur Web](#)
- [PHP](#)
- [MySQL](#), [MariaDB](#) ou [PostgreSQL](#)
- [Postman](#)

Séance 1 — Pourquoi une API HTTP ?



Séance 1 — Protocole HTTP

- [Aperçu du protocole HTTP](#)
- [L'évolution du protocole HTTP](#)
- [Codes de réponse HTTP](#)
- [En-têtes HTTP](#)
- [Méthode de requête HTTP](#)

Séance 1 — Routeur

- `$ _SERVER["REQUEST_URI"]`
- `$ _SERVER["REQUEST_METHOD"]`
- `parse_url`

Séance 1 — Routeur

- GET /
 - Affiche le texte “Home”
- GET /users
 - Afficher le texte “Users”
- POST /users
 - Afficher le texte “User created”
- PATCH /users/:user
 - Renvoie le texte “Updated user #???”
- DELETE /users/:user
 - Renvoie le texte “Deleted user #???”
- Dans tous les autres cas
 - Renvoie le texte “Route not found”

Séance 2 — Requête et Réponse HTTP

- [Postman](#)
- [JSON](#)
- [http_response_code](#)
- [header](#)
- [json_encode](#)
- [php://](#)
- [json_decode](#)

Séance 2 — Réponse HTTP

- GET /
 - Affiche le texte “Home”
 - Renvoie un code de statut de 200
 - Renvoie une réponse au format JSON
- GET /users
 - Afficher le texte “Users”
 - Renvoie un code de statut de 200
 - Renvoie une réponse au format JSON
- POST /users
 - Accepte dans le corps de la requête une adresse email et un mot de passe
 - Afficher le corps brute de la requête
 - Renvoie un code de statut de 201
 - Renvoie une réponse au format JSON
- PATCH /users/:user
 - Accepte dans le corps de la requête une adresse email ou un mot de passe (ou les deux)
 - Renvoie le texte “Updated user #???”
 - Renvoie un code de statut de 200
 - Renvoie une réponse au format JSON
- DELETE /users/:user
 - Renvoie le texte “Deleted user #???”
 - Renvoie un code de statut de 200
 - Renvoie une réponse au format JSON
- Dans tous les autres cas
 - Renvoie le texte “Route not found”
 - Renvoie un code de statut de 404
 - Renvoie une réponse au format JSON

Séance 3 — Entité

- [PDO](#)
- [PDO::query](#)
- [PDO::exec](#)
- [PDO::prepare](#)
- [PDO::execute](#)
- [PDO::fetchAll](#)
- [password_hash](#)

Séance 3 — Entité

- GET /users
 - Affiche la liste des utilisateurs en base de données sous la forme d'un tableau associatif
- POST /users
 - Ajoute un utilisateur avec une adresse email et un mot de passe (stocker sa version chiffrée) en base de données
- PATCH /users/:user
 - Met à jour l'email ou le mot de passe ou les deux en base de données
 - Renvoie un code de statut 404 si l'utilisateur n'existe pas
- DELETE /users/:user
 - Supprime l'utilisateur en base de données
 - Renvoie un code de 404 si l'utilisateur n'existe pas