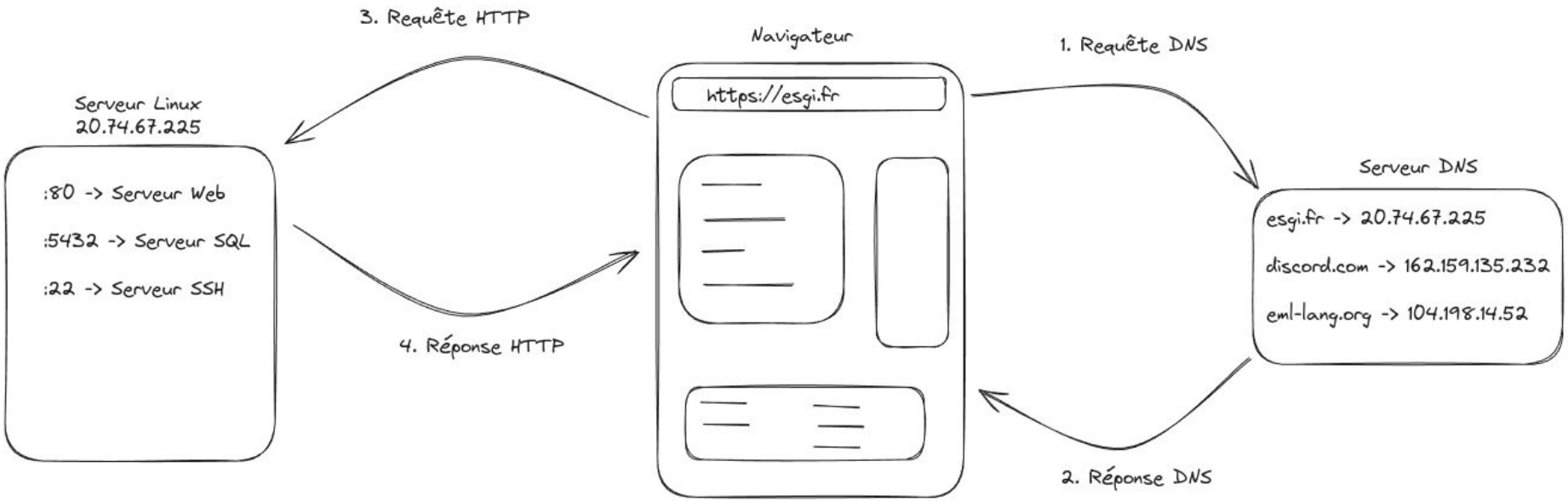
Protocoles d’Authentification

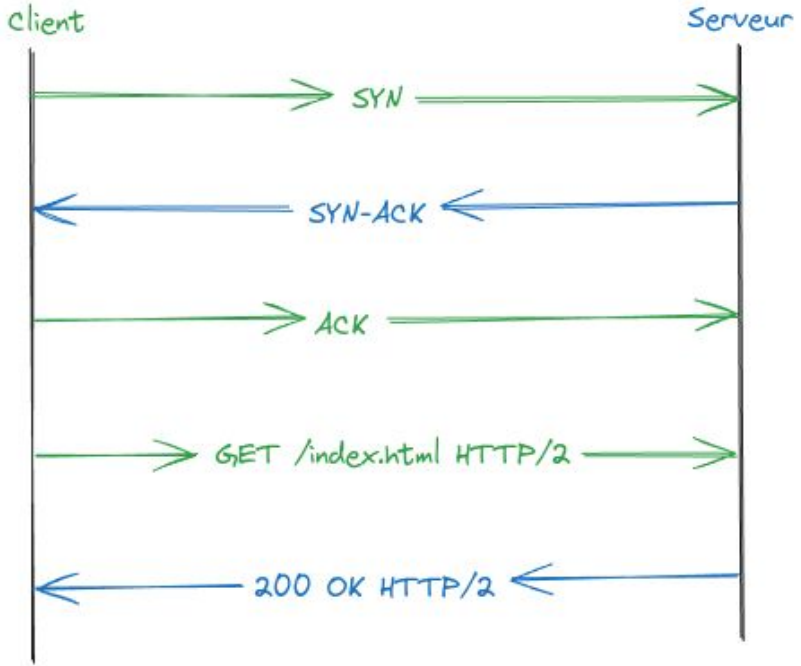
Amin NAIRI <anairi@esgi.fr>

https://slash.nairi.cloud/s/positionnement

Séance 1 : Introduction aux protocoles d'authentification



Séance 1 : Introduction aux protocoles d'authentification



Requête

```
POST /users HTTP/2
Host: esgi.fr
Content-Type: application/json
Content-Length: 1234

{
  "email": "user@domain.com",
  "password": "s3cr3t"
}
```

Réponse

```
200 OK HTTP/2
Content-Type: application/json
Content-Length: 1234

{
  "userId": "a1b2c3d4"
}
```

Séance 1 : Introduction aux protocoles d’authentification

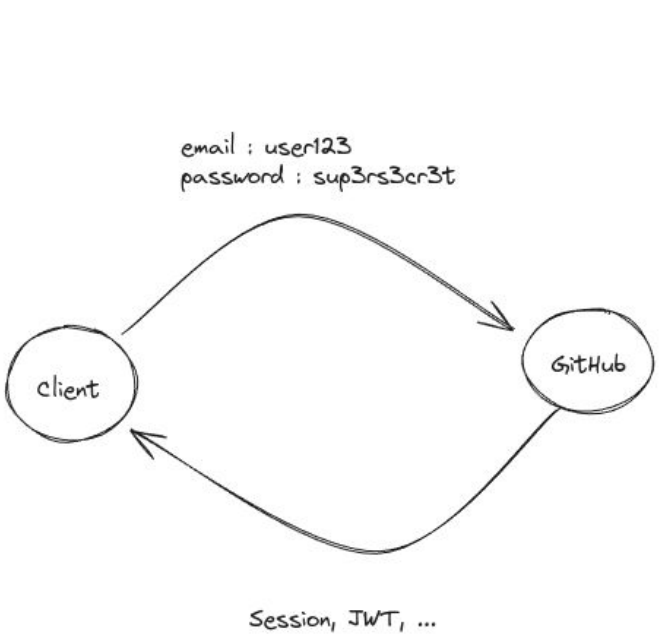
Time-based One-Time Password (TOTP)

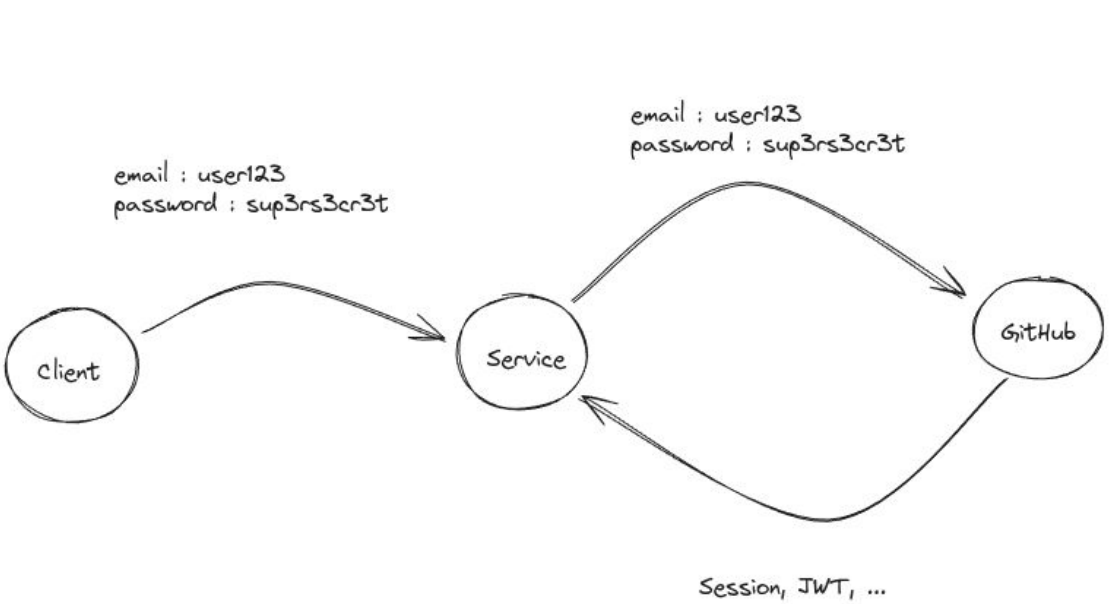
Initialisation	Utilisation
1. Création d'une chaîne de caractères aléatoire côté serveur	1. Inscription via email et mot de passe
2. Partage via un QRCode	2. Vérification des informations
3. Enregistrement de la chaîne de caractères dans une application supportée Ex: Google Authenticator	3. Demande d'entrée du TOTP
	4. Vérification côté serveur
	5. Autorisation

Algorithme

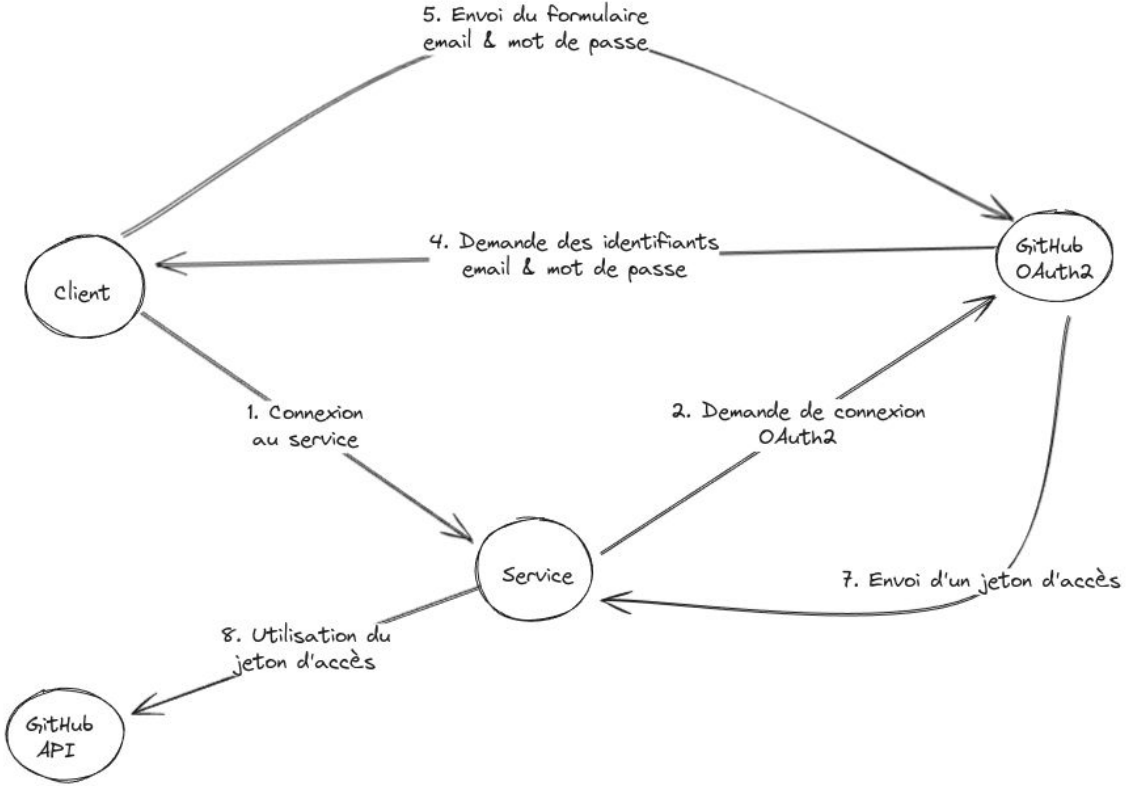
1. Récupération de la chaîne de caractères aléatoire
2. Récupération du nombre de secondes UNIX
3. Division par un intervalle (ex: 30 secondes)
4. Réunion des deux dans un appel de fonction (HMAC-SHA256)
5. Troncation aux 6 premiers nombres

Séance 1 : Introduction aux protocoles d’authentification

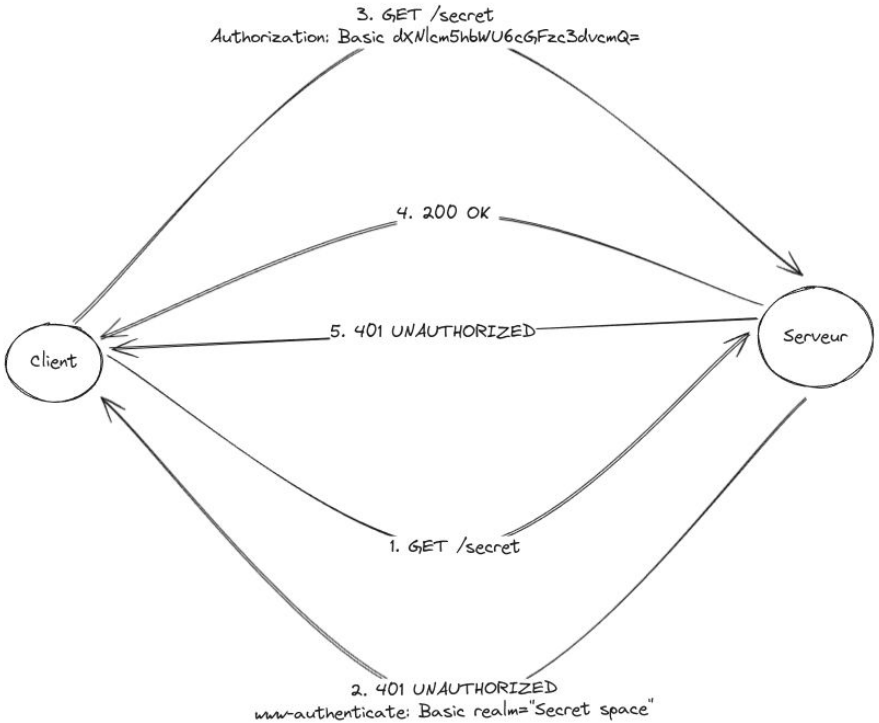


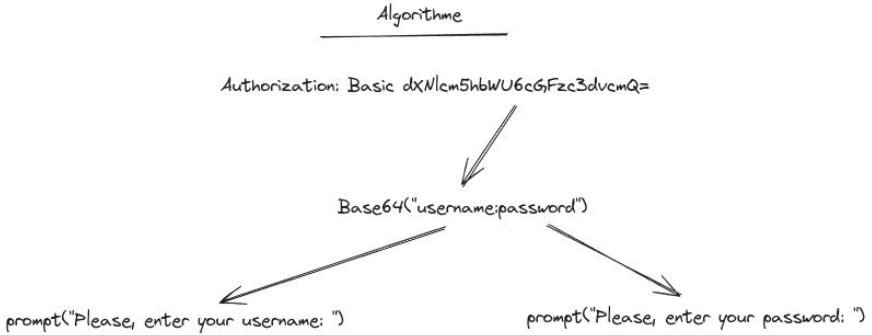


Séance 1 : Introduction aux protocoles d'authentification



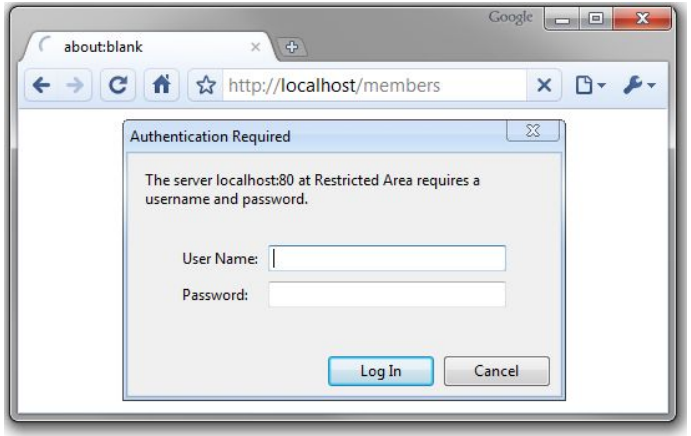
Séance 1 : Introduction aux protocoles d’authentification



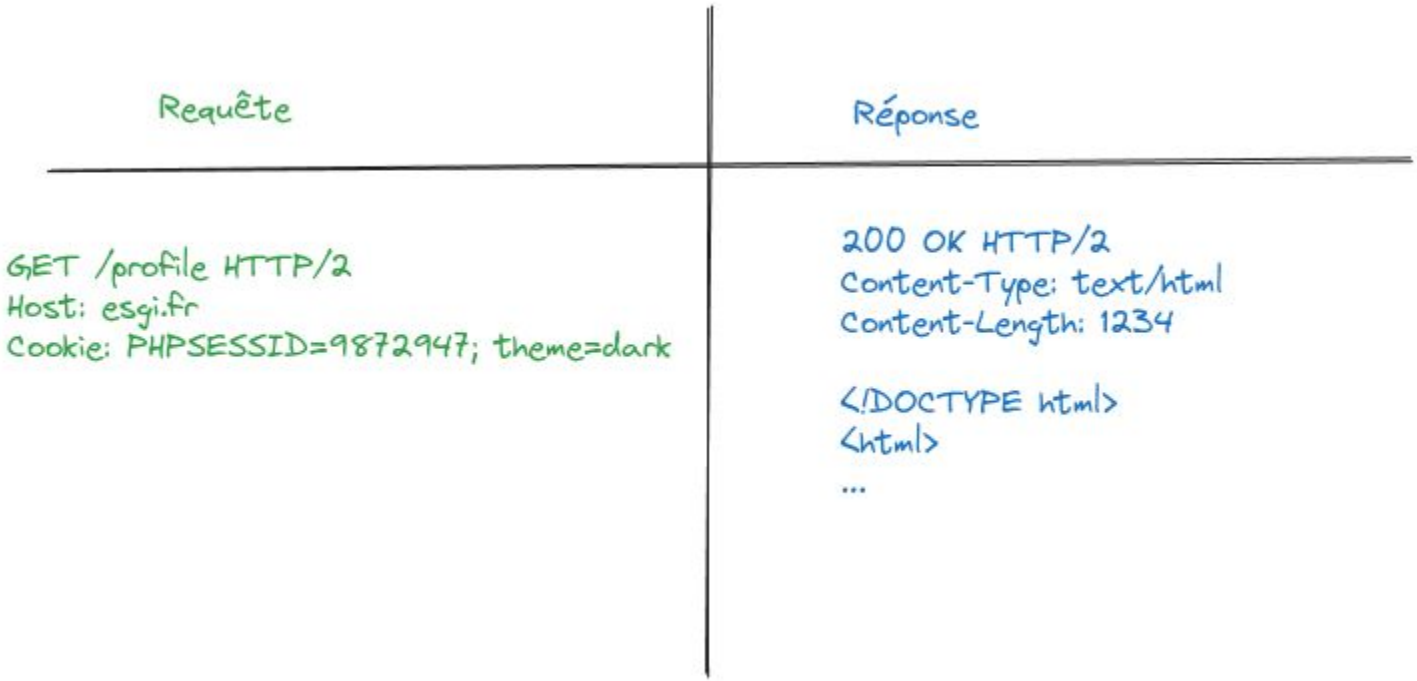


Séance 1 : Exercice

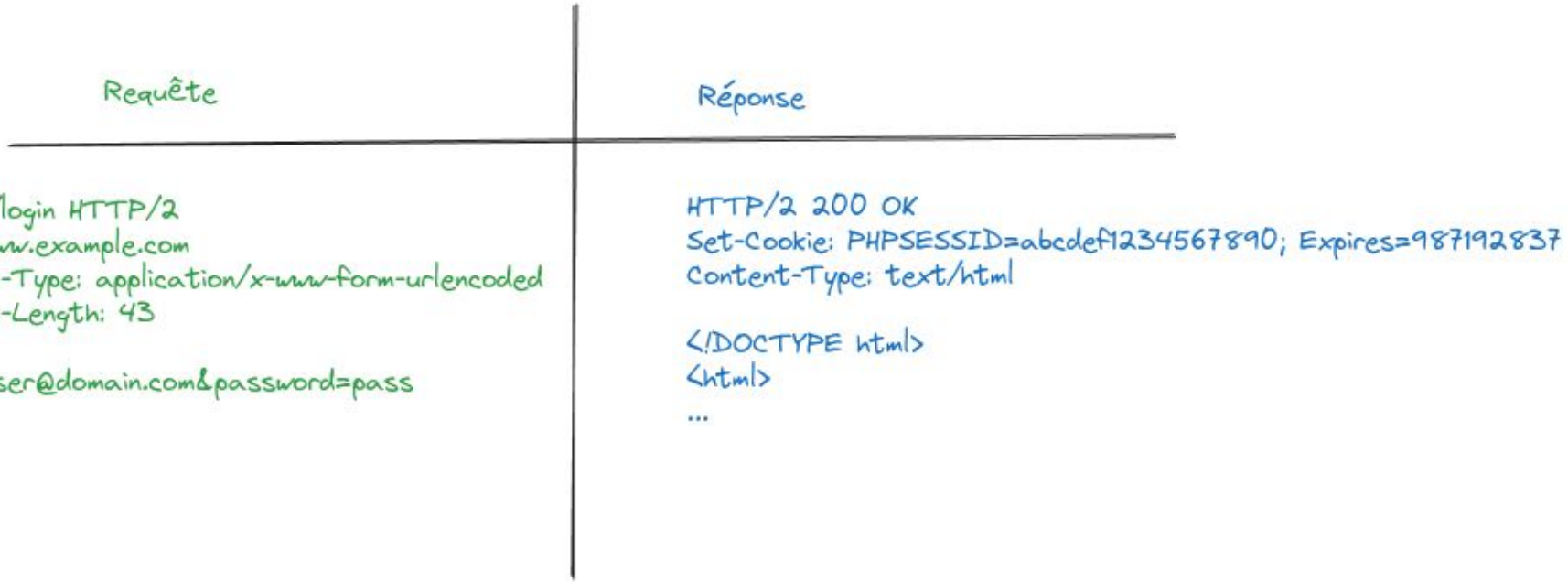
- Implémenter une authentification basique
- Utiliser Node.js pour le côté serveur
- Utiliser la librairie http
- Utiliser l’objet Buffer pour l’encodage
- Ne pas utiliser de HTML
- Envoyer une erreur au bout de 3 tentatives



Séance 2 : Cookies & Sessions



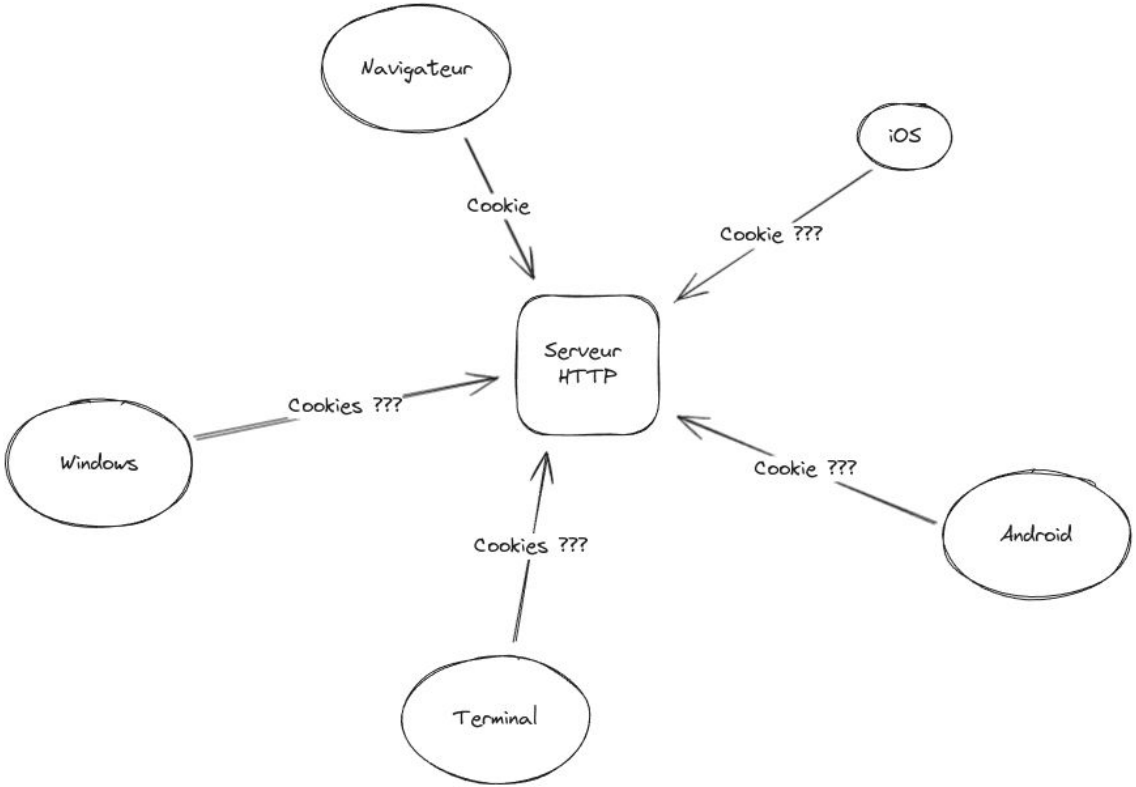
Séance 2 : Cookies & Sessions



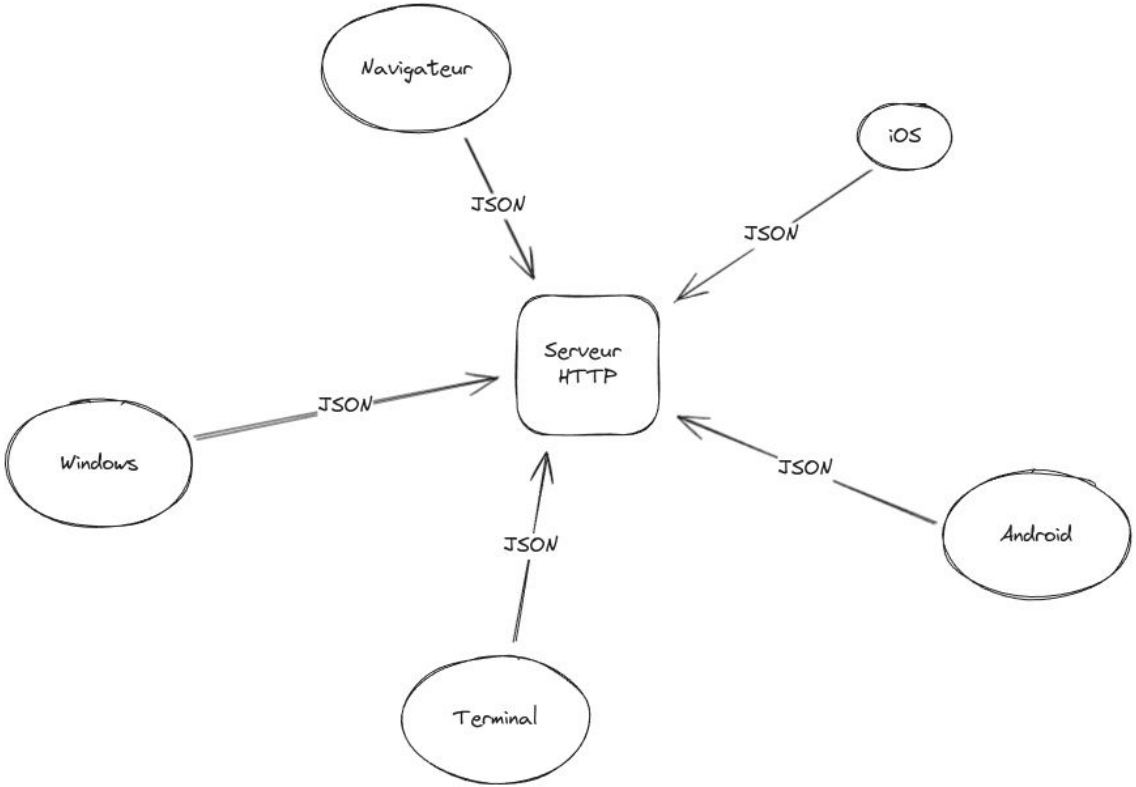
Séance 2 : Exercice

- Créer une route renvoyant un formulaire HTML permettant de se connecter à l'aide d'un email et d'un mot de passe
- Envoyer le formulaire sur un serveur Node.js
- Le serveur doit renvoyer une réponse de succès, avec un identifiant unique de session
- Créer un fichier qui contiendra les informations de session
- Stocker l'email dans le fichier de session
- Créer une seconde route renvoyant un document HTML permettant d'afficher l'email de l'utilisateur en fonction de sa session

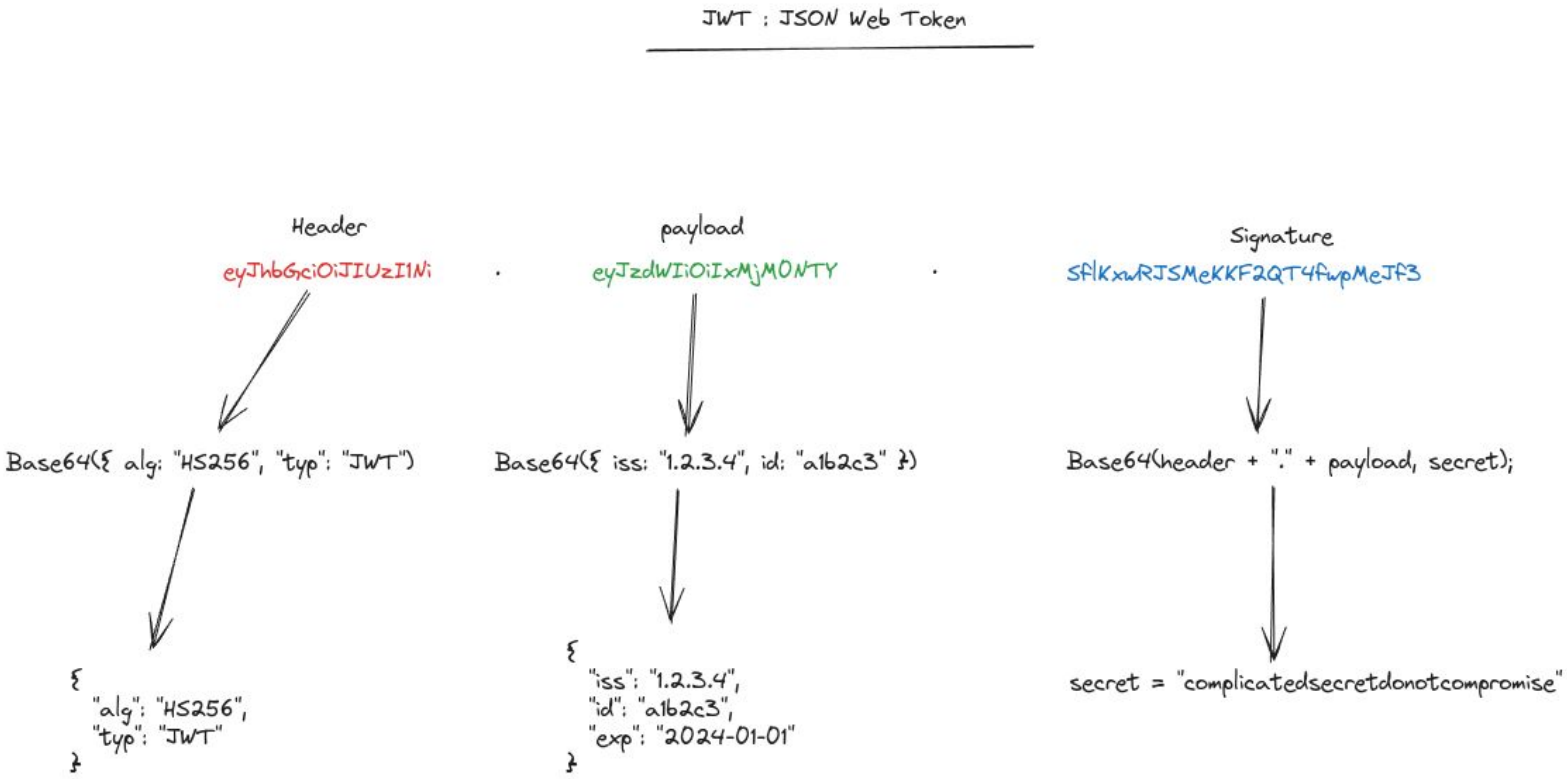
Séance 3 : JSON Web Token



Séance 3 : JSON Web Token



Séance 3 : JSON Web Token



Séance 3 : Exercice

- Utiliser la libraire jsonwebtoken
- Servir tous les fichiers en utilisant la librairie Express et Express Static
- Créer une page HTML avec un formulaire contenant un email et un mot de passe
- Utiliser une requête asynchrone avec l'API Web Fetch afin d'envoyer une demande de connexion
- Retourner un JWT et le stocker en utilisant l'API Web Storage
- Stocker l'adresse email dans le JWT
- Créer une page HTML permettant d'afficher l'adresse email d'un utilisateur connecté
- Envoyer une requête avec l'API Web Fetch afin de récupérer les informations de l'utilisateur
- Afficher l'adresse email de l'utilisateur dans la page

Séance 4 : Travail Pratique

- Durée : 1h30
- Sujet : non-communiqué
- Correction : 1h30

- Créer une page HTML permettant d’afficher l’adresse email d’un utilisateur connecté
- Envoyer une requête avec l’API Web Fetch afin de récupérer les informations de l’utilisateur
- Afficher l’adresse email de l’utilisateur dans la page