



Welcome home !

Règlement intérieur

Poster une issue

JavaScript

Backend & Strapi

Prérequis

👋 Semaine 1

Introduction au concept de backend

Bases de Strapi

👋 Semaine 2

Les références

Customisation des routes

👋 Semaine 3

L'authentification

🎥 Vidéo

Introduction

Principes généraux d'authentification

Principes d'authentification avec Strapi

La collection User

Envoyer des requêtes authentifiées

Les différentes BDD utilisables

Stocker des fichiers

Policy et nouvelles routes

👋 Semaine 4

Recherche et tri

Git

Git & Github

Northflank : héberger le serveur et la BDD

HTML / CSS

Principes généraux d'authentification

hash, salt et token

Il est important de ne **jamais stocker les mots de passe de vos utilisateurs dans la base de données** afin que même en cas de faille de sécurité, les mots de passe ne puissent être divulgués.

Les algorithmes de hash, tels que le `MD5`, `Bcrypt` ou le `SHA256`, permettent de **transformer** une chaîne de caractères en une autre, de telle sorte que la même chaîne de caractères **produit toujours le meme résultat** mais qu'il est difficile de faire le **calcul inverse**.

Quelques exemples de `hash` avec l'algorithme `MD5` :

- `hello` > `5d41402abc4b2a76b9719d911017c592`
- `123456` > `e10adc3949ba59abbe56e057f20f883e`
- `password` > `5f4dcc3b5aa765d61d8327deb882cf99`
- `s8nZT8eEQ4` > `e2474262007c8db4c2c8b85ff3bd1012`

Mais un simple MD5 n'est **pas suffisant**. En effet, il existe des dictionnaires qui permettent de retrouver les textes correspondants aux MD5 de mots de passe **fréquemment utilisés** :

Par exemple avec [ce site](#), on peut trouver certains des mots de passe ci-dessus :

- `5d41402abc4b2a76b9719d911017c592` > `hell`
- `0`