Programmation Web – Avancé

JavaScript & Node.js

Partie 25 : Hachage d'information

Version 2020

HAUTE ÉCOLE LÉONARD DE VINCI

Hacher les passwords côté serveur ?

```
[
  {
    "username": "teacher@vinci.be",
    "email": "teacher@vinci.be",
    "password": "$2b$10$yS7G6Ruhw9o.d47E
ZVM4v.UMzf2HDLFSbFM1kMS9mA/h4oq/.LiSW"
  }
]
```

- Recommandé
- Défense contre les « hash attacks » :
  - Salt
  - Salt round : nombre de fois que la hashage est fait
- Hachage sous Node.js : **bcrypt** [89.]
- Installation & initialisation : **npm install bcrypt**

```
const bcrypt = require("bcrypt");
const saltRounds = 10;
```

Hacher un password via bcrypt

- Existence de méthodes asynchrones ou synchrones : **hash()**, **hashSync()**, **compare()**, **compareSync()**
- Hasher le passsword : **hash()**

```
async save() {
    let userList = getUserListFromFile(FILE_PATH);
    const hashedPassword = await bcrypt.hash(this.password, saltRounds);
    userList.push({username: this.email,email: this.email,password: hashedPassword,});
    saveUserListToFile(FILE_PATH, userList);
    return true;
  }
```

Comparer un password haché via bcrypt

- Comparer le passsword : **hash()**

```
//asynchronous checkCredentials without async / await / new Promise()
  checkCredentials(email, password) {
    if (!email || !password) return false;
    let userFound = User.getUserFromList(email);
    // return a resolved promise
    if (!userFound) return Promise.resolve(false); // return false; : that would raise
        an error when calling checkCredentials.then()     // return the promise
    return bcrypt
      .compare(password, userFound.password)
      .then((match) => match)
      .catch((err) => console.error("checkCredentials:", err));
  }
```

Garder son password crypté ou hashé côté client ?

- Non recommandé

```
  {
    "username": "teacher@vinci.be",
    "email": "teacher@vinci.be",
    "password": "$2b$10$yS7G6Ruhw9o.d47E
ZVM4v.UMzf2HDLFSbFM1kMS9mA/h4oq/.LiSW"
  }
```

Crypter côté serveur

- **DEMO : Hachage des passwords au sein du backend**

  MyCMS : augmenter la sécurité des données côté serveur avec bcrypt