WebCryptoAPI

Faire de la crypto côté client

WebCryptoAPI

- Faire de la crypto côté client
  - Pour des raisons de performances
  - Pour des raisons de sécurité
  - Pour des raisons de confidentialité !!! ... par rapport au serveur
- Norme W3C officiellement supporté par tous les navigateurs modernes
  - Norme flexible, multi algorithme
  - Orienté sécurité

WebCryptoAPI

- Basé sur le concept de promise inclus dans ECMAScript 6
- window.crypto
- asm.js
  - Structure de donnée performante
- Exemple
  - Calculer un hash d’un mot de passe côté client
  - Chiffrer nos donnés pour qu’elles soient ensuite entreposé dans une base type NoSQL

Hasher des mot de passe

- SHA-1
- SHA-256
- SHA-384
- SHA-512

Hasher des mot de passe

```
window.crypto.subtle.digest(
        {
        name: "SHA-256",
        },
        new Uint8Array([1,2,3,4]) // La donnée type source
)
.then(function(hash){
        // hash contient le résultat sous forme de ArrayBuffer
        console.log(new Uint8Array(hash));
})
.catch(function(err){
        console.error(err);
});
```

Importer une clé

```
window.crypto.subtle.importKey(
        "raw",                                  // format de la clef
        ArrayBuffer([1, 2, 3, 4 …, 256]),       // clef au bon format
        {
            name: "AES-CBC",                    // Algorithme cible
            length: 256
        },
        false,
        ["encrypt", "decrypt"]
)
.then(function(key){
    // retourne l'objet clef sous forme de KeyObject
})
.catch(function(err){
});
```

Chiffré

```
window.crypto.subtle.encrypt(
    algo,    // Paramètres d'identification de l'algorithme cible
    key,     // clef au format interne générée par generateKey ou importKey
    data     // Donnée claire au format ArrayBuffer
)
.then(function(encrypted){
    // encrypted sont les données chiffrées sous forme d'ArrayBuffer
})
.catch(function(err){
});
```

Déchiffré

```
window.crypto.subtle.decrypt(
     algo,   // Paramètres d'identification de l'algorithme cible
     key,    // clef au format interne générée par generateKey ou importKey
     data    // Données chiffrées au format ArrayBuffer
)
.then(function(decrypted){
    // decrypted sont les données claires sous forme d'ArrayBuffer
})
.catch(function(err){
});
```