

Programmation Web – Avancé

JavaScript & Node.js

Partie 19 : JSON

Version 2020



Attribution –
Partage dans les
Mêmes Conditions
4.0 International
(CC BY-SA 4.0)

*Presentation template
by [SlidesCarnival](#)*



Introduction à JSON

Comment structurer ses messages ? Comment structurer de l'info que l'on souhaite enregistrer ?



Introduction à JSON

- JavaScript **O**bject **N**otation = JSON
- Syntaxe pour enregistrer et échanger des données
- JSON : texte en notation JS



Introduction à JSON

- Types de données valides :
 - **string**
 - **number**
 - **object**
 - **array**
 - **boolean**
 - **null**
- Donc pas de **function**, **date** et **undefined**



Exemple de données JSON

```
[
  {
    "email": "raphael@voila.com",
    "fullname": "Raphael Baroni"
  },
  {
    "email": "jkj@herenqn.com",
    "fullname": "JK Roling"
  },
  {
    "email": "serena@gmail.com",
    "fullname": "Serena Here"
  }
]
```



Enregistrement de fichiers JSON côté serveur

- Type de fichiers : **.json**
- MIME : **“application/json”**



Sérialisation et désérialisation de données

- Conversion d'un objet JS en JSON pour envoi vers une application : **JSON.stringify(myObj)**

```
function saveUserListToFile(filePath, userList) {  
  const fs = require("fs");  
  let data = JSON.stringify(userList); // userList is an array of objects  
  fs.writeFileSync(filePath, data);  
}
```

Côté serveur

```
const setUserSessionData = (user) => {  
  const storageValue = JSON.stringify(user);  
  localStorage.setItem(STORE_NAME, storageValue);  
};
```

Côté client



Sérialisation et désérialisation de données

- Conversion des données JSON reçues d'une application en un objet JS :

JSON.parse(myJSON)

```
function getUserListFromFile(filePath) {  
  const fs = require("fs");  
  if (!fs.existsSync(filePath)) return [];  
  let userListRawData = fs.readFileSync(filePath);  
  let userList;  
  if (userListRawData) userList = JSON.parse(userListRawData);  
  else userList = [];  
  return userList;  
}
```

Côté serveur



Sérialisation et désérialisation de données

● JSON.parse(myJSON)

```
const getUserSessionData = () => {  
  const retrievedUser = localStorage.getItem(STORE_NAME);  
  if (!retrievedUser) return;  
  return JSON.parse(retrievedUser);  
};
```

Côté client



Introduction à JSON

🕒 DEMO : MPA & modèle persistant via un fichier JSON (hbs)

Reprise de notre application précédente afin de rendre les données des utilisateurs persistantes à l'aide d'un fichier JSON (mini BD).



Introduction à JSON

- NB: **DEMO : MPA & modèle persistant via un fichier JSON (pug)**