

# Programmation Web – Avancé

**BINV2150 A : JavaScript (& JAVA SERVLETS)**

**Week 7**

**R. Baroni / J.L. Collinet / C. Damas**



*Presentation template  
by [SlidesCarnival](https://www.slidescarnival.com/)*

0

# Table des matières

Tous les sujets traités pendant ce cours...



## Table des matières

---

1. Engagement pédagogique
2. Introduction au contexte d'utilisation de JS
3. Introduction au langage JS côté client, à l'utilisation d'APIs du navigateur et de librairies JS
4. Introduction aux communications (synchrone) client /serveur



## Table des matières

---

- 5. Introduction aux single-page web applications et aux communications asynchrones client / serveur
- 6. Introduction à l'authentification sécurisée d'un utilisateur et aux cookies
- 7. **Projet mettant en œuvre une SPA et des librairies JS (présentation)**

4

# Introduction aux communications (synchrone) client / serveur

Découvrons le backend Java...



## Table des matières :

### **4. Introduction aux communications (synchrones) client / serveur**

1. Quels types de communications ?
2. Quelles technologies ?
3. Introduction à un backend Java (Jetty)
4. Communications synchrones client /serveur
5. **Introduction à la gestion de sessions côté serveur (Jetty)**



## **Table des matières :**

**4. Introduction aux communications (synchrone)  
client / serveur**

**6. Introduction à JSON**

**7. Introduction à JSON côté Java : Genson**



## **Introduction à la gestion de session côté serveur (Jetty)**

---

- Session = identification d'un client au travers de plusieurs requêtes à un même serveur et enregistrement d'information au sujet de cet utilisateur
- Utilisation d'un cookie par Jetty pour gérer cette session





## **Introduction à la gestion de session côté serveur (Jetty)**

- Cas typiques de fin de session :
  - Redémarrage de l'application web.
  - Effacement des cookies côté client.
  - Fermeture complète du browser.



## Introduction à la gestion de session côté serveur : HttpSession

- Méthode **getSession()** de l'**HttpServletRequest** : permet de créer ou obtenir la **HttpSession**
- L'instance de Session agit comme une **Map<String,Object>** via **setAttribute** et **getAttribute**
- Une requête traitée par le serveur sans que ce dernier n'appelle **getSession()** ne possèdera donc pas de session.



## Introduction à la gestion de session côté serveur : HttpSession

- Une fois la session créée, **getSession()** retourne la même session jusqu'à son expiration.
- Détails :  
<https://docs.oracle.com/javaee/7/api/javax/servlet/http/HttpSession.html>



## Introduction à la gestion de session côté serveur : HttpSession

- Utilisation d'une session dans le cadre d'une authentification
  - Lors d'une authentification réussie, la session est remplie par l'id de l'utilisateur :

```
req.getSession().setAttribute("id", idUser);
```



## Introduction à la gestion de session côté serveur : HttpSession

- Utilisation d'une session dans le cadre d'une authentification
  - Lors des requêtes ultérieures, le serveur récupère l'id à partir de la session :

```
Object idUser=req.getSession().getAttribute("id");  
if (idUser!=null) {...}
```



## Introduction à la gestion de session côté serveur : HttpSession

- **DEMO-14** : suite au login de l'utilisateur, créer une session. Retournez l'email et l'ID de l'utilisateur pour toutes connections ultérieures de l'utilisateur à **/home/\*** pendant la session. En cas de connexion à **/home/\*** sans login, redirection vers **/connect**.



## Introduction à JSON

---

- JavaScript Object Notation = JSON
- Syntaxe pour enregistrer et échanger des données
- JSON : c'est du texte en notation JS
- **JSON.stringify(myObj)** : convertir des données d'un objet JS en JSON pour envoi à un serveur



## Introduction à JSON

---

- **JSON.parse(myJSON)** : convertir des données JSON reçues d'un serveur vers un objet JS.
- Quelques exemples :  
[https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)
- Type de fichiers : **.json**
- MIME : **"application/json"**





## Introduction à JSON

---

- Types de données valides :
  - string
  - number
  - object
  - array
  - boolean
  - null
- Donc pas de fonctions, dates et undefined



## Introduction à JSON

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



## Introduction à JSON côté Java : Genson

- En Java, plusieurs librairies permettent de transformer du JSON en Java.
- Nous utiliserons Genson : <http://genson.io/>
- A vous d'explorer cette librairie Java...



## Introduction à JSON côté Java : Genson

- **DEMO-15** : On souhaite utiliser JSON comme une BD côté serveur pour enregistrer les e-mails d'utilisateurs associés à leur nom complet. Quand on accède au endpoint **/users**, on voit la liste des personnes enregistrées.



## Introduction à JSON côté Java : Genson

- **DEMO-15** : Quand on accède au endpoint **/register**, on peut ajouter un compte utilisateur via un formulaire, puis on est redirigé vers **/users** afin d'avoir une nouvelle liste d'utilisateurs.

7

# Projet mettant en œuvre une SPA et des librairies JS

Faites ce qui vous tient à cœur et épatez vous en asynchrone ...



## **Projet mettant en œuvre une SPA et des librairies JS**

- Consignes détaillées : voir template du projet
- Développement :
  - Investissement proposé : 12h en classe + ~12h suppl.
  - Durée : 3 semaines (semaines 9 à 11)
  - Présentation et validation de vos objectifs (semaine 9)
  - Groupes de 2 étudiants, sauf exception
- Présentation (4h en classe) : semaine 12



# Exercices

A vous de jouer...





## Backend Jetty / JSON / GENSON

- ◎ **EX-08** : Créer un formulaire d'introduction de films (title, duration, producer, budget). Le formulaire est accessible via **/films/add**. La liste de film est disponible, sous forme d'un tableau, via **/films**. Après l'ajout d'un film, renvoyer à **/films**.



## Mise à disposition de vos animations JS

### 🕒 EX-09 :

A) Finalisez les exercices de la semaine 4.

B) Rendez accessibles, en travaillant le côté « esthétique », vos animations via un menu et les endpoints :

- **/words** (EX-03)
- **/tables** (EX-04A)
- Optionnellement : **/border** (EX-04B)



## Références

- |     |   |
|-----|---|
| [1] | MDN web docs, Introduction to web APIs. Lien :<br><a href="https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction">https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction</a> |
| [2] | MDN web docs, JavaScript Guide. Lien :<br><a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide</a>   |
| [3] | w3schools.com, JavaScript Tutorial. Lien :<br><a href="https://www.w3schools.com/js/default.asp">https://www.w3schools.com/js/default.asp</a>   |
| [4] | tutorialspoints.com, Javascript Tutorial : Lien :<br><a href="https://www.tutorialspoint.com/javascript/index.htm">https://www.tutorialspoint.com/javascript/index.htm</a>  |



## Références

[5]	Medium.com, Neal Burger, The end of life of IE11. Lien : <a href="https://medium.com/@burger.neal/the-end-of-life-of-internet-explorer-11-12736f9ff75f">https://medium.com/@burger.neal/the-end-of-life-of-internet-explorer-11-12736f9ff75f</a>
[6]	w3schools.com, JS HTML DOM. Lien : <a href="http://www.w3schools.com/js/js_htmlDOM.asp">http://www.w3schools.com/js/js_htmlDOM.asp</a>
[7]	MDN web docus, Basics of HTTP. Lien : <a href="https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP">https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP</a>