

# Programmation Web – Avancé

JavaScript & Node.js

SPAs & routage des pages et composants



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

*Presentation template  
by [SlidesCarnival](#)*



# Routage des pages et composants d'une SPA

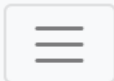
Comment afficher le bon composant sur ma SPA ?



## SPA : Demande d'affichage du composant de Login

### JavaScript & Node.js full course

MyCMS



Home

Register

Login



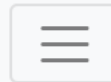
Demo : Monolith SPA

Welcome to MyCMS...

Comment implémenter le  
nouvel affichage ?

### JavaScript & Node.js full course

MyCMS



Home

Register

Login

Demo : Monolith SPA

Email

johndoe

Password

.....

Submit



## **SPA : gestion de l'affichage des pages et composants**

---

- Pas de standards
- Afficher / cacher :
  - des éléments d'un menu
  - des éléments d'une page
  - ...
- Problème :
  - pas d'historique !
  - pas d'URL spécifique quand clic sur un menu item



## Possibilités pour la gestion de l'affichage des pages

---

- Simple gestion de containers & HTML dynamique (via template literals)
- Web Components API [\[69.\]](#) :
  - Création d'éléments personnalisés réutilisables
  - Création de HTML templates
  - ...



## Possibilités pour la gestion de l'affichage de composants de la page

- Utilisation d'un langage de templating : **Handlebars** [\[67.\]](#), **Mustache**...
- Utilisation de bibliothèques légères : **Lit** [\[70.\]](#), **underscore** [\[71.\]](#)
- Utilisation de frameworks / bibliothèques : **React**, **Vue**, **Angular**...



## Architecture et fonctions recommandées dans ce cours pour le routage des pages et composants

- Structure de votre SPA en composants fonctionnels : **Navbar**, **LoginPage**, **RegisterPage**, **AddPizzaPage**...
- Chaque composant fonctionnel = 1 module JS
- Auto-render de chaque composant fonctionnel



## **Architecture et fonctions recommandées dans ce cours pour le routage des pages et composants**

- Routage lors d'un clic sur un élément de la barre de navigation :
  - Appel du composant associé à l'élément cliqué (et auto-render du composant)
  - Affichage dans le browser de l'URL associée à l'élément cliqué
  - Garder l'URL dans l'historique





## **Architecture et fonctions recommandées dans ce cours pour le routage des pages et composants**

- Routage lors du chargement du frontend :  
Appel du composant associé à l'URL en cours
- Gestion du routage lors de l'utilisation de l'historique du browser : appel du composant associé à l'URL



## **Architecture et fonctions recommandées dans ce cours pour le routage des pages et composants**

- Routage lors de redirection :
  - Appel du composant associé à la redirection (et auto-render du composant)
  - Affichage dans le browser de l'URL associée à l'élément redirigé
  - Garder l'URL dans l'historique



## Architecture et fonctions recommandées dans ce cours pour le routage des pages et composants

- Créer un routeur en Vanilla JS : Source d'inspiration [\[72.\]](#)
- Bibliothèques existantes : **page.js**, **Vaadin Router**...

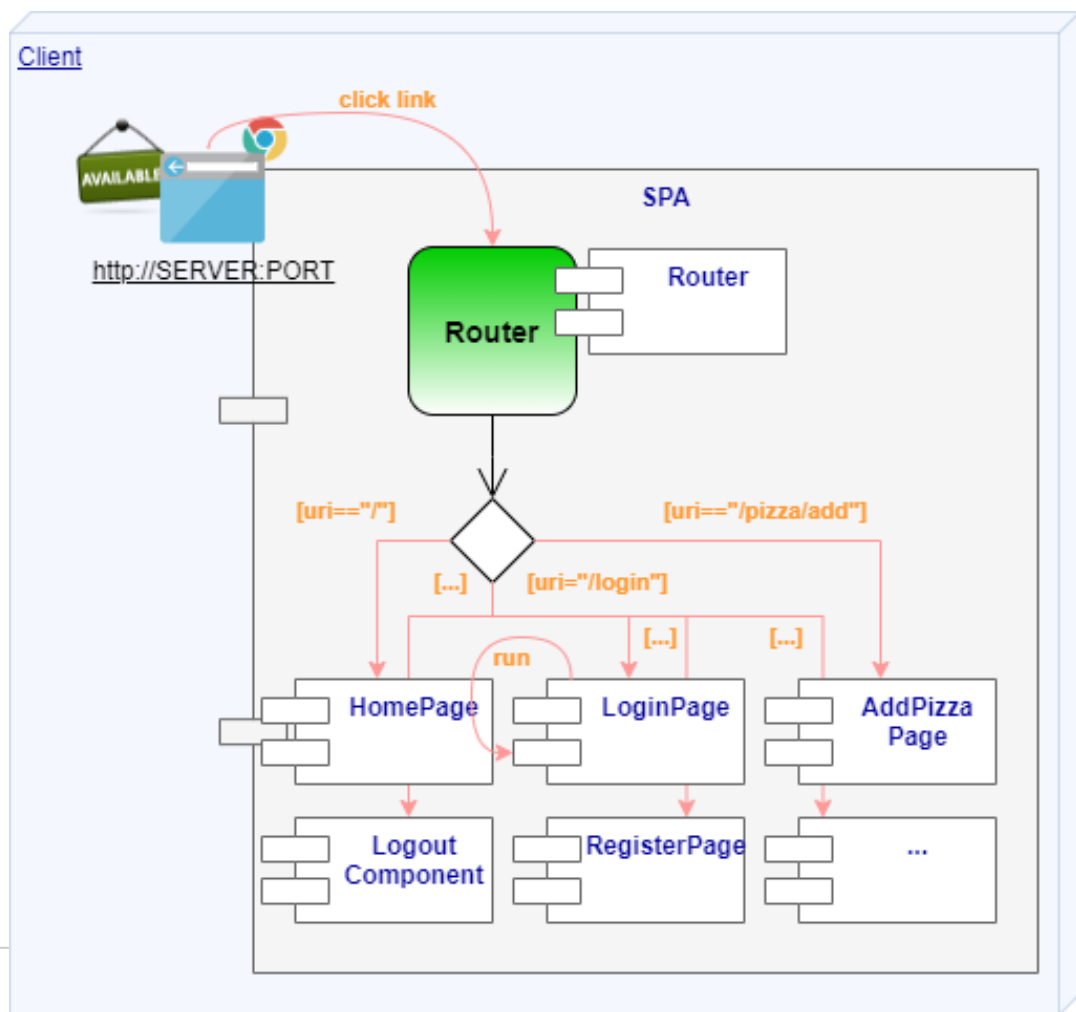


## Routage des pages et composants

---

- Utilisation de bibliothèques existantes : **page.js**, **Vaadin Router**...

**SPA : affichage du  
composant LoginPage suite  
à une action d'un  
utilisateur**





## Cacher de l'information dans l'HTML



- Option A : Utilisation classique des attributs « id », « name »... d'un élément
- Option B : Utilisation de « data-attributes » :
  - Création en HTML de « data attributes » pour un élément : **data-\***
  - Accès en JS : conversion de **-** en **camelCase**



## Cacher de l'information dans l'HTML



● Exemple de deux attributs de données :

```
<a class="nav-item nav-link" href="#" data-uri="/home" data-extra-info="Home sweet home">Home</a>
```



```
const onNavigate = (e) => {  
  let uri;  
  uri = e.target.dataset.uri;  
  console.log("extra info associated to this link:", e.target.dataset.extraInfo,  
    // ... extra info associated to this link: Home sweet home  
  );  
  navBar.addEventListener("click", onNavigate);  
}
```





## Routage des pages et composants d'une SPA

### 🕒 DEMO : Création d'une SPA pour une pizzeria : Step 2 : gestion du routage des pages composants

- Création des composants **Navbar** et **Logout**
- MàJ des composants **LoginPage**, **HomePage**... pour utiliser le router
- Mise en place du router lors des clics sur la **Navbar**
- Gestion du routage au chargement / refresh
- Routage lors de l'utilisation de l'historique du browser
- Routage en cas de redirection vers une URL





## Routage des pages et composants d'une SPA

### 🕒 DEMO : Création d'une SPA pour une pizzeria : Step 2 : gestion du routage des pages composants

- Comment gérer les données de session : username, token... ?
- Comment appeler des opération d'une RESTful API qui sont protégées ?