



Applications web et mobiles

TP n°7 : Breadcrumb et ajout de topics

Étape 35 – Le breadcrumb – partie 1 : les données

Le haut de la page des *topics* doit contenir ce que l'on appelle un « breadcrumb », c'est-à-dire une aide à la navigation :

« Tous les cours / nom_du_cours_sélectionné ».

Créez un nouveau composant `breadcrumb`. Ce composant sera utilisé à la fois pour les topics et les posts, vous allez donc le rendre assez générique. Pour cela, dans le fichier `breadcrumb.component.ts`, rajoutez la déclaration de l'interface suivante :

```
export interface BreadcrumbData {  
  nom: string;  
  route: string;  
}
```

qui représente les données d'un « *item* » du breadcrumb (par exemple « Tous les cours »). Le champ `nom` représente ce qui s'affichera sur votre navigateur et le champ `route` indique la route/le lien à suivre pour atteindre l'item correspondant. Par exemple la route correspondant au nom « Tous les cours » devrait être `'/cours'`. Les données nécessaires pour afficher tout un breadcrumb peuvent donc être stockées sous la forme d'un tableau de `BreadcrumbData`. Par exemple, le breadcrumb de la liste des topics du cours « Prog web et mobile » correspond au tableau :

```
[  
  { nom: 'Tous les cours', route: '/cours' },  
  { nom: 'Prog web et mobile', route: '' }  
]
```

Ici, on supposera que, si une route est une chaîne vide, c'est que l'item correspondant doit être affiché sans lien HTML.

Étape 36 – Le breadcrumb – partie 2 : transmission et récupération des données

Dans votre fichier `breadcrumb.component.ts`, faites en sorte que la classe `BreadcrumbComponent` récupère un tableau de `BreadcrumbData` de son parent (voir le décorateur `@Input` vu en cours). Appelons `paths` le champ qui reçoit ces données.

Pour ce TP, le parent du breadcrumb est la classe `TopicsComponent`. Rajoutez-lui un champ `breadcrumb` de type `BreadcrumbData[]` et remplissez le dans votre méthode `ngOnInit`. Enfin, dans le fichier

topics-component.html, passez à la balise `<app-breadcrumb></app-breadcrumb>` l'information contenue dans votre champ `breadcrumb`.

Étape 37 – Le breadcrumb – partie 3 : l'affichage des données

Dans le fichier `breadcrumb.component.html`, écrivez le code permettant l'affichage du breadcrumb. Ici, le plus simple est d'utiliser les breadcrumbs de bootstrap (cf. <https://getbootstrap.com/docs/5.3/components/breadcrumb>). Pour afficher tous les éléments du tableau de `BreadcrumbData`, il suffit donc d'utiliser une directive `*ngFor`. Pour afficher ou non un lien HTML, exploitez la directive `*ngIf` : vous pouvez par exemple écrire un code similaire à :

```
<span *ngIf="path.route.length != 0"><a [routerLink]="path.route">{{path.nom}}</a></span>
<span *ngIf="path.route.length == 0">{{path.nom}}</span>
```

ou encore mieux :

```
<span *ngIf="path.route.length != 0; then thenBlock; else elseBlock"></span>
<ng-template #thenBlock><a routerLink="{{path.route}}">{{path.nom}}</a></ng-template>
<ng-template #elseBlock>{{path.nom}}</ng-template>
```

Ainsi, si le champ `route` de la variable `path` est vide, c'est le lien `<a>` qui sera inclus dans la page web, sinon ce sera le texte sans ancre.

Étape 38 – Ajout de nouveaux sujets dans le backend

Dans votre *backend*, écrivez un script PHP `saveNewTopic.php` qui récupère dans sa variable `$_POST` l'identifiant d'un cours ainsi qu'une chaîne de caractères correspondant à un nouveau sujet. Le script doit vérifier que la chaîne est non vide et que l'utilisateur suit bien le cours, et que ce dernier ne possède pas déjà ce sujet. Le cas échéant, le sujet est créé dans la base de données et les propriétés du cours sont mises à jour dans la base (nombre de topics, *etc.*). Enfin, le script renvoie un message contenant l'identifiant du nouveau sujet. Si, en revanche, le sujet existe déjà ou bien si la chaîne est vide, le script renvoie un message d'erreur.

Testez votre script avec Postman.