

# TD n°2

	<p>Tous les exercices utilisent <a href="#">Bootstrap</a> pour la partie CSS.  <code>ember install ember-bootstrap</code></p> <p>pour les fonts :  <code>ember install ember-font-awesome</code></p>
---	--

- [Bases emberJS](#)
- [Computed properties](#)
- [Actions](#)

Créer le projet **td2**

## - Exercice : choix multiples

### - Objectifs

1. Création de Computed properties
2. Manipulation de Tableaux d'objets
3. Création de helper et de component

### - Fonctionnalités

1. Lister les produits disponibles
2. Ajouter un ou plusieurs produits disponibles aux produits sélectionnés
3. Ajouter tous les produits disponibles aux produits sélectionnés
4. Retirer un ou plusieurs produits sélectionnés
5. Retirer tous les produits sélectionnés
6. Valider le choix de produits sélectionnés
7. Revenir au choix des produits

### - Helper/component

Création d'un composant permettant d'afficher une liste HTML composée d'un tableau d'éléments :

Créer le helper **array-contains**

```
ember g helper array-contains
```

```
import Ember from 'ember';

export function arrayContains(params) {
  const [items, value] = params;
  return items.includes(value);
}

export default Ember.Helper.helper(arrayContains);
```

Créer le composant **multi-select** :

```
ember g component multi-select
```

Composant modifié le 12/02/2019 16h53

```
import Component from '@ember/component';
import { computed } from '@ember/object';
import $ from 'jquery'

export default Component.extend({
  tagName: 'select',
  classNames: ['form-control'],
  selectedIds:[],
  attributeBindings: ['multiple','size'],

  multiple: true,
  size: 10,
  selectedElements: computed('selectedIds.[]', function() {
    return this.get('selectedIds').map((id) => {
      return this.get('elements').findBy('id', id);
    });
  }),
  change(event){
    const selectedIds = $(event.target).val();
    this.set('selectedIds', selectedIds || []);
  },
  doubleClick(event){
    const selected = $(event.target).val();
    this.sendAction('dblClick',selected,this.get('id'));
  }
});
```

Template :

```
{{#each elements as |elm|}}
  <option value="{{elm.id}} selected={{array-contains selectedIds
elm.id}}>{{elm.title}}</option>
{{/each}}
```

## - Classes/Routes

	<b>Listes</b>
	dispoltems Tableau des produits disponibles (Liste 1)
	includedItems Tableau des produits à inclure (Liste 2)
	dispoltemsIds_ Tableau des ids des éléments sélectionnés dans la liste des produits disponibles
<b>Classe</b>	includedItemIds_ Tableau des ids des éléments sélectionnés dans la liste des produits à inclure
	dispoltems_* Tableau des éléments sélectionnés dans la liste des produits disponibles (* sur dispoltemsIds_[])
	includedItems_* Tableau des ids des éléments sélectionnés dans la liste des produits à inclure (* sur includedItemIds_[])
	<b>actions</b>
	addToIncluded() Ajoute les produits disponibles en surveillance ( <b>dispoltems_</b> ) à la liste des produits à inclure ( <b>includedItems</b> )
	addAllToIncluded() Ajoute tous les produits disponibles ( <b>dispoltems</b> ) à la liste des produits à inclure ( <b>includedItems</b> )
	removeFromIncluded() Retire les produits sélectionnés en surveillance ( <b>includedItems_</b> ) pour les remettre dans la liste des produits disponibles ( <b>items</b> )
	RemoveAllFromIncluded() Retire tous les produits à inclure ( <b>includedItems</b> ) pour les remettre dans la liste des produits disponibles ( <b>dispoltems</b> )

Remarque : Il est possible de factoriser les actions en ajoutant uniquement une action dont le prototype est le suivant :

```
addTo:function(source,destination,items){...}
```

Il faudra dans ce cas passer les paramètres nécessaires à l'action **addTo** dans le template.

**dispoltems** : à intégrer au model hook

```
[  
  {  
    "id": "1",  
    "url":  
"https://tutorialzine.com/2018/01/what-every-developer-should-know-about-cryptocurrency",  
    "title": "What Every Developer Should Know About Cryptocurrency",  
    "image":  
"https://tutorialzine.com/media/2018/01/everything-you-should-know-cryptocurrency.png"  
  },  
  {  
    "id": "2",  
    "url":  
"https://tutorialzine.com/2018/01/15-interesting-javascript-and-css-libraries-for-january-2018",  
    "title": "15 Interesting JavaScript and CSS Libraries for January"
```

```
2018",
    "image":
"https://tutorialzine.com/media/2018/01/15-interesting-js-and-css-libraries-2018.png"
},
{
    "id": "3",
    "url":
"https://tutorialzine.com/2017/12/the-best-javascript-and-css-libraries-for-2017",
    "title": "The Best JavaScript and CSS Libraries for 2017",
    "image":
"https://tutorialzine.com/media/2017/12/the-best-javascript-and-css-libraries-for-2017.png"
},
{
    "id": "4",
    "url": "https://tutorialzine.com/2017/10/getting-started-with-graphql",
    "title": "Getting Started With GraphQL",
    "image":
"https://tutorialzine.com/media/2017/10/getting-started-graphql.png"
},
{
    "id": "5",
    "url":
"https://tutorialzine.com/2017/11/9-free-icon-packs-for-web-developers",
    "title": "9 Free Icon Packs For Web Developers",
    "image": "https://tutorialzine.com/media/2017/10/9-free-icon-packs.png"
},
{
    "id": "6",
    "url":
"https://tutorialzine.com/2018/01/10-free-programming-books-you-should-read-in-2018",
    "title": "10 Free Programming Books You Should Read in 2018",
    "image":
"https://tutorialzine.com/media/2018/01/10-free-programming-books-you-should-read.png"
},
{
    "id": "7",
    "url":
"https://tutorialzine.com/2016/12/the-languages-frameworks-tools-you-should-learn-in-2017",
    "title": "The Languages and Frameworks You Should Learn in 2017",
    "image":
"https://tutorialzine.com/media/2016/12/the-languages-and-frameworks-you-should-learn-in-2017.png"
}
]
```

## - Vues

### - Interface

**ex1/step1 :**

**Sélectionnez vos produits**

**Produits disponibles :**

- 50 Must-have plugins for extending Twitter Bootstrap
- Making a Super Simple Registration System With PHP and MySQL
- How To Make a Digital Clock with jQuery and CSS3
- Smooth Diagonal Fade Gallery with CSS3 Transitions
- Your First Backbone.js App – Service Chooser

**Produits sélectionnés :**

- Mini AJAX File Upload Form
- Create a slide-out footer with this neat z-index trick

2 produits sélectionnés

[Etape suivante](#)

**ex1/step2 :**

**Votre choix**

**Produits sélectionnés :**

-  Create a slide-out footer with this neat z-index trick
-  Smooth Diagonal Fade Gallery with CSS3 Transitions
-  50 Must-have plugins for extending Twitter Bootstrap

[Retour au choix précédent](#)

**- Logique applicative / comportement de l'interface**

- Les 4 boutons d'ajout ou de retrait des services à inclure s'activent ou se désactive (disabled) en fonction du contenu des listes et/ou des éléments sélectionnés
  - bouton ►| : ajoute le(s) élément(s) sélectionnés de la liste de gauche à la liste de droite
  - bouton ►►| : ajoute tous les éléments sélectionnés de la liste de gauche à la liste de droite
  - bouton |◀ : retire le(s) élément(s) sélectionnés de la liste de droite et les place dans la liste de gauche
  - bouton |◀◀ : retire tous les éléments sélectionnés de la liste de droite et les place dans la liste de gauche
- Un double clic sur l'un des éléments des 2 listes ajoute ou retire l'élément et le passe dans la liste associée
- Le nombre de produits inclus se met à jour en fonction du nombre d'éléments dans la liste de droite
- Le passage à l'étape suivante n'est possible que si la liste de droite n'est pas vide
  - le bouton "Passer à l'étape suivante" affiche la vue suivante (en passant step à 2)
  - la vue affiche la liste des éléments ajoutés dans les éléments à inclure

**Tests**

Réalisez des tests d'**acceptance** pour votre application.

From:  
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**



Permanent link:  
<http://slamwiki2.kobject.net/richclient/emberjs/td2>

Last update: **2019/08/31 14:21**