

Instruction TP

Partie 1

- Ajouter un fichier de comportement au composant

créer un fichier de comportement lié au composant image

NB:

- ember generate <typeGénéré> <[chemin/]nomDuFichier>
- Pour ember le typeGénéré pour créer un comportement est composant-class, et doit avoir le même nom que le composant

- Ajouter une variable isLarge initialisé à false dans le image.js

```
export default class RentalImageComponent extends Component {
  constructor(...args) {
    super(...args);
    this.isLarge = false;
  }
}
```

ou

```
export default class RentalImageComponent extends Component {
  isLarge = false;
}
```

Cette syntaxe est permit grâce à ember (attention si la première notation ne marche pas passer a la deuxième).

- Puis rajouter la modification de l'image en fonction de la variable dans le composant:

```
{{#if this.isLarge}}
<div class="image large"> <!--la Classe large permet de changer la taille
grâce au css prédéfinie-->
<img ...attributes>
<small>View Smaller</small> </div>
{{else}} <div class="image">
<img ...attributes>
<small>View Larger</small>
```

```
</div> {{/if}}
```

vous pouvez tester voir si l'image est bien agrandie sur le site en mettant la variable `isLarge=true`.

- Demander à ember de suivre les changement de la variable `isLarge` avec le decorator `@tracked` (a placer dans le js)
pensé à importer la fonction `tracked` :

```
import { tracked } from '@glimmer/tracking';
```

```
export default class RentalImageComponent extends Component {  
  @tracked isLarge = false;  
}
```

- Ajouter une fonction pour changer la variable `isLarge` avec le decorator

```
@action toggleSize() {  
  this.isLarge = !this.isLarge;  
}
```

n'oublier pas d'importer le decorateur:

```
import { action } from '@ember/object';
```

- Remplacer les `<div>` par des `<button>` et leurs ajouter un onclik avec la fonction que vous avez créé.

```
<button type="button" class="image large" {{on "click" this.toggleSize}}>
```

Bravo!

Partie 2 (crée une page et crée un bouton vers cette page):

- générer une page (template)
- rajouter une route a `app/router.js` avec le nom de la page
- crée un `linkTo` vers cette template depuis une page accessible (ex: `index.hbs`)
- remplir la template de votre nouvelle page avec le code suivant:

```
<Jumbo>  
  <LinkTo @route="généralité" class="button">Fin</LinkTo>  
</Jumbo>
```

- aller sur la page et cliquer sur le bouton