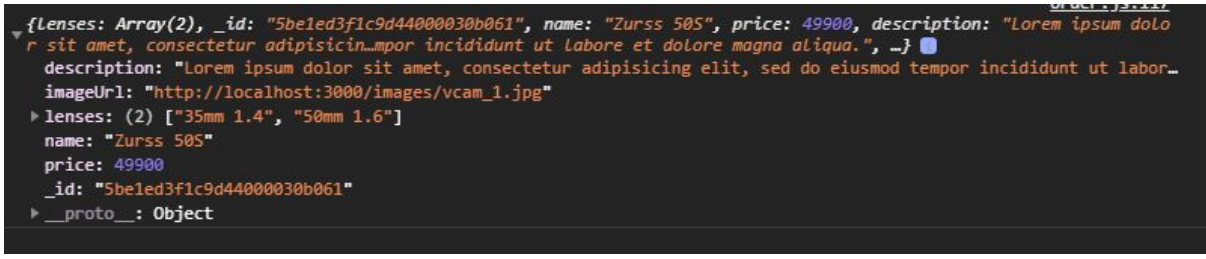
Procédure de Test Orinoco

Class Product.js :

- Récupération de tous les produits :
  - On appelle la méthode all() de la manière suivante : Product.all()
  - Le résultat devrait être équivalent à l'image :

- Récupération d'un produit par l'ID :
  - On appelle la méthode get() en lui passant l'ID du produit, de la manière suivante : Product.get("5be1ed3f1c9d44000030b061")
  - Le résultat devrait être équivalent à l'image :



- Génération de la Card du produit en HTML
  - On appelle la méthode show() en lui passant en paramètre un "Product" de type Object, exemple :

```
Product.show({
    id: '5be1ed3f1c9d44000030b061',
    description: 'Lorem ipsum dolor sit amet',
    price: 49900,
    imageUrl: 'http://localhost:3000/images/vcam_1.jpg'
})
```

  -
  - Show() renvoie une strings, comme dans l'image ci-dessous

```
<div class="col-md-4 ">
          <div class="card-product card border-primary mb-3" style="max-width: 20rem;" data-product_id="5be1ed3f1c9d44000030b061" data-
product_price="49900">
              <div class="card-header"><a href="product.html?id=5be1ed3f1c9d44000030b061">Zurss 50S</a></div>
              <div class="card-body" style="background-image: linear-gradient(rgba(0,0,0,0.5), rgba(48,48,48,0.5)),
url('http://localhost:3000/images/vcam_1.jpg'); background-size: cover !important;">
                  <p class="text-white">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua.</p>
                  <h4 class="float-right text-white">49900 €</h4>
              </div>

              <button type="button" class="btn btn-primary">Ajouter au panier</button>
          </div>
      </div>
```

Class Cart.js :

- Récupération de tous les éléments dans le panier
  - On appelle la méthode all(), comme suit: Cart.all()
  - Le résultat devrait ressembler à cela:

```
{id: "5be1ed3f1c9d44000030b061", count: 3, price: "49900"}
{id: "5be1ef211c9d44000030b062", count: 2, price: "309900"}
```

- Ajout d'un produit dans le panier
  - Utilisation de la méthode add(), comme suit:

```
Cart.add({
    id: product_id,
    count: 1,
    price: product.price
})
```

  - Si le produit est inexistant dans le panier alors il s'ajoute dans la liste, sinon la valeur s'incrémente de 1
  - Le résultat devrait être proche de ça :

```
{id: "5be1ef211c9d44000030b062", count: 2, price: "309900"}
```

- Compter le nombre de même produit ajouté dans le panier
  - On appelle la méthode count(), en lui passant l'identifiant du produit, de la manière suivante : Cart.count("5be1ed3f1c9d44000030b061")
  - Produit dans le panier :

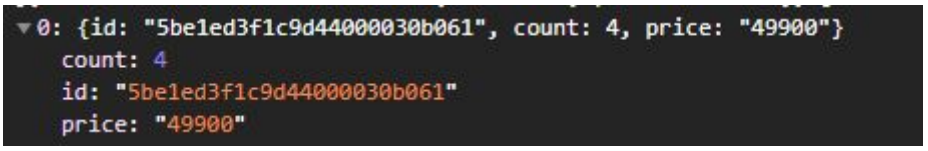
```
0: {id: "5be1ed3f1c9d44000030b061", count: 3, price: "49900"}
   count: 3
   id: "5be1ed3f1c9d44000030b061"
   price: "49900"
```

  - Résultat retourné par la méthode :

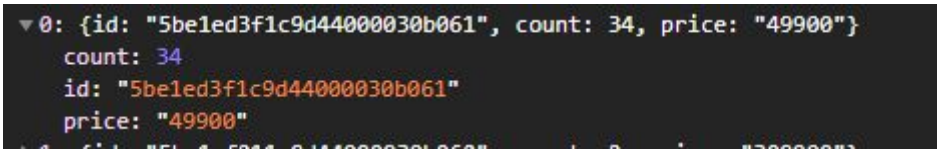
```
3
```

- Incrémenter un produit d'un produit, si il a le même ID
  - La méthode increment Count(), doit obligatoirement avoir l'ID du produit en paramètre, et peut facultativement avoir un nombre en 2ème paramètre
  - Si on appelle la méthode de la manière suivante : IncrementCount("5be1ed3f1c9d44000030b061"), elle ajoutera automatiquement + 1 au produit.

Celle-ci fera passer le produit dans le panier à 4 éléments



- ➢ Si on lui passe un nombre, celui-ci changera le résultat de “count” contenu dans l’objet du produit dans le panier.
- ➢



❖ Retirer des produits dans le panier

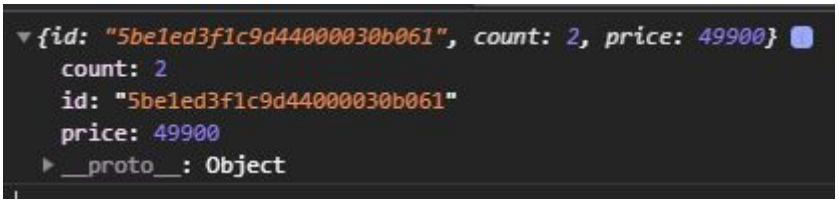
- ➢ Utilisation de la méthode decrementCount("5be1ed3f1c9d44000030b061")
- ➢ même fonctionnement que IncrementCount() mais à l’inverse au lieu d’ajouter, cela retire.

❖ Supprimer un article du panier

- ➢ On appelle la méthode remove(“5be1ed3f1c9d44000030b061”)
- ➢ L’ensemble de produit ayant cet ID sera supprimé du panier

❖ Récupérer un produit du panier

- ➢ On utilise la méthode get(“5be1ed3f1c9d44000030b061”)
- ➢ la méthode retourne :



❖ Récupérer le nombre total de produit dans le panier

- ➢ On appelle la méthode number Product()
- ➢ La méthode nous retournera alors le nombre de produits dans le panier en additionnant le count de chaque produit.

❖ Récupérer le montant total de l’ensemble des produits du paniers cumulé

- ➢ On appelle la méthode getTotalPrice()
- ➢ La méthode retourne le montant totale de du panier

❖ Savoir si un article est déjà dans le panier

- ➢ On appelle la méthode has(“5be1ed3f1c9d44000030b061”)
- ➢ Si ce produit est déjà dans le panier, alors la méthode retourne ”true” sinon elle retournera “false” si aucun produit comportant l’id n’est trouvé

- ❖ Réinitialiser le panier
  - ➢ On appelle la méthode reset()
  - ➢ Renvoie un tableau vide dans le localStorage à la clé “panier” afin de supprimer l’ensemble des éléments du panier

- ❖ Afficher un produit dans le panier
  - ➢ On appelle la méthode show(“5be1ed3f1c9d44000030b061”) de manière asynchrone
  - ➢ Celle-ci nous renvois le HTML suivant :

```
<img src="http://localhost:3000/images/vcam_1.jpg" alt="Franck JS 105">
<div class="cart-item_infos">
    <p class="cart-item_price">49900 €</p>
    <a class="cart-item_name" href="product.html?id=5be1ed3f1c9d44000030b061">Zurss 50S</a>
    <label class="mt-2" data-children-count="1">
        Qté:
        <input type="number" min="1" value="2">
    </label>
</div>
<span class="btn-delete-item" data-remove-product="">
    <svg width="1rem" height="1rem" viewBox="0 0 16 16" class="bi bi-x-circle" fill="currentColor" xmlns="http://www.w3.org/2000/svg">
        <path fill-rule="evenodd" d="M8 15A7 7 0 1 0 8 1a7 7 0 0 0 0 14zm0 1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"></path>
        <path fill-rule="evenodd" d="M4.646 4.646a.5.5 0 0 1 .708 0L8 7.293l2.646-2.647a.5.5 0 0 1 .708.708L8.707 8l2.647 2.646a.5.5 0
 .293 8 4.646 5.354a.5.5 0 0 1 0-.708z"></path>
    </svg>
</span>
```

Class ValidationForm.js :

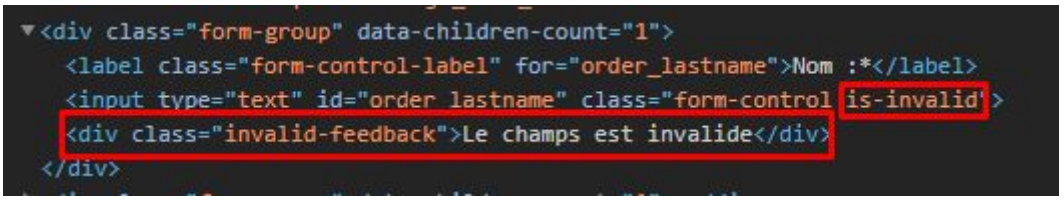
- ❖ Vérifier si un email est valide
  - ➢ On appelle la méthode emailIsValid(“alex@test.fr”)
  - ➢ Si l’email passé en paramètre correspond au critère d’un email la fonction renverra “true”, sinon “false”

- ❖ Vérifier si le champ est vide ou null
  - ➢ On appelle la méthode fieldIsInvalid(NodeElement)
  - ➢ Si le champs est vide ou null alors la méthode renverra “true” et dans le cas contraire “false”

- ❖ Rendre un champ invalide
  - ➢ On appelle la méthode renderInvalidField(NodeElement, “Message d’erreur à afficher à l’utilisateur”(optionnel))
  - ➢ La méthode changera l’input comme dans les exemples suivant :
    - ■ Avant vérification

```
<div class="form-group" data-children-count="1">
    <label class="form-control-label" for="order_lastname">Nom :*</label>
    <input type="text" id="order_lastname" class="form-control">
</div>
```

- Après vérification



❖ Rendre un champs valide

➢ On appelle la méthode renderValidField(NodeElement)

➢ la méthode supprimera la class “is-invalid” et la div avec la class “invalid-feedback” si l’élément était invalide avant et y ajoutera la class “is-valid” dans l’input