

# Orinoco

## Plan de tests

Projet #5  
Par Olivier Tholy

# Accéder au site Orinoco

Pour afficher l'intégralité du site, il faut au préalable lancer la commande « node server » dans le terminal de votre ordinateur.

Le repository Github est disponible à cette adresse :

<https://github.com/sthylo/orinoco>

Le site Orinoco est disponible à cette adresse :

<https://sthylo.github.io/orinoco/index.html>

# Connexion avec l'API

## Objectif du test :

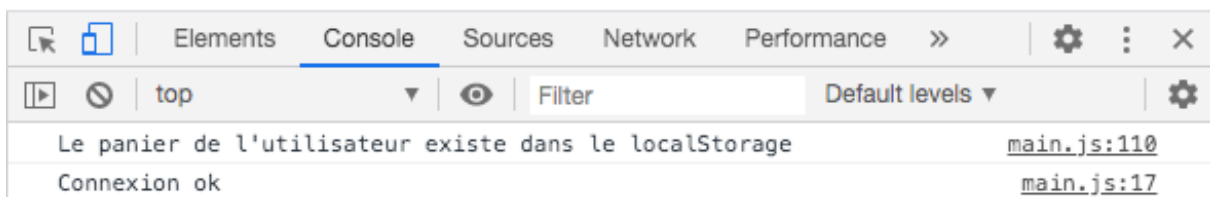
Vérifier que la fonction de connexion avec l'API se fait correctement pour récupérer les données des différents produits.

## Code JavaScript :

```
10 function getProduits(){
11     return new Promise((resolve) => {
12         let request = new XMLHttpRequest();
13         request.onreadystatechange = function() {
14             if(this.readyState == XMLHttpRequest.DONE && this.status == 200)
15             {
16                 resolve(JSON.parse(this.responseText));
17                 console.log("Connexion ok");
18                 error = document.getElementById("error");
19                 if(error){
20                     error.remove();
21                 }
22             }
23         }
24         request.open("GET", APIURL + idProduit);
25         request.send();
26     });
27 }
```

La console doit retourner la réponse « Connexion ok » grâce à la commande `console.log("Connexion ok");`

## Console :



## Résultat du test : Réussi

# Affichage de la liste des produits

## Objectif du test :

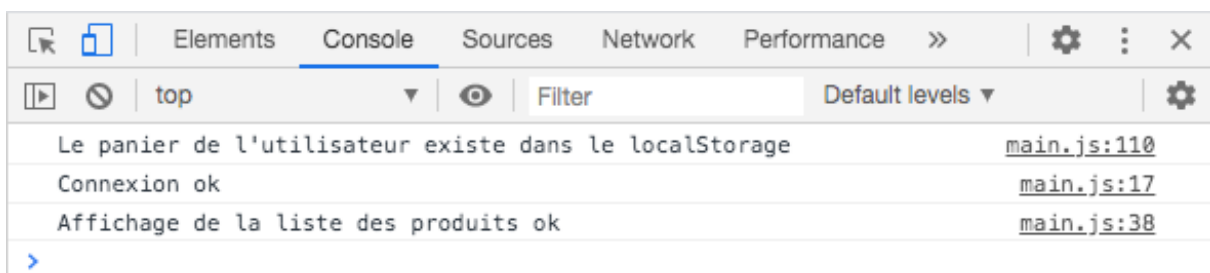
Vérifier que la fonction d'affichage des produits se fait correctement sur la page index.html

## Code JavaScript :

```
30  async function allProductsList(){
31      const produits = await getProduits();
32
33      let listProduct = document.createElement("section");
34      listProduct.setAttribute("id", "list-articles");
35
36      let main = document.getElementById("main");
37      main.appendChild(listProduct);
38      console.log("Affichage de la liste des produits ok");
```

La console doit retourner la réponse « Affichage de la liste des produits ok » grâce à la commande `console.log("Affichage de la liste des produits ok");`

## Console :



Résultat du test : Réussi

# Affichage des détails du produit

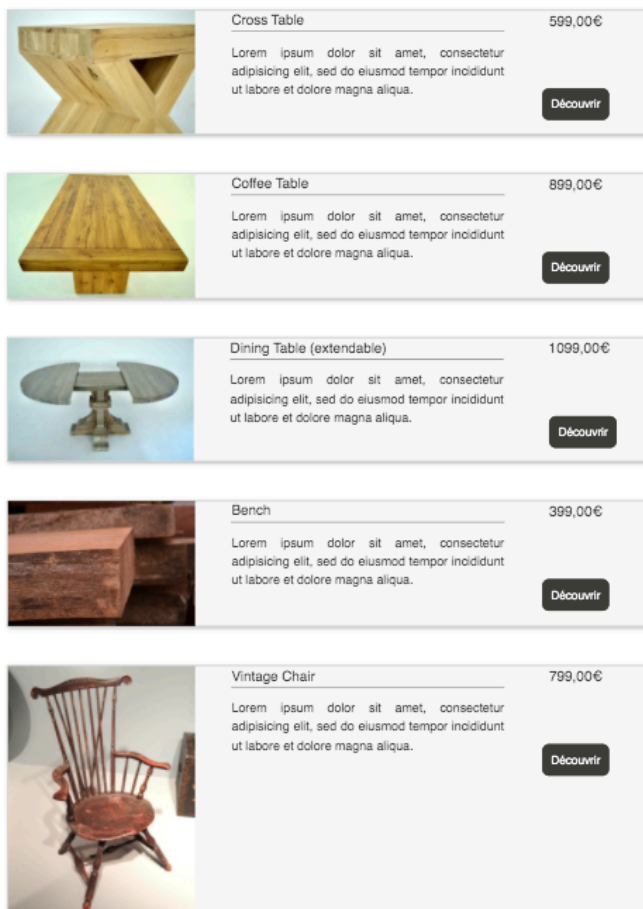
## Objectif du test :

Vérifier que nous récupérons bien les détails des produits présents dans l'API pour les afficher sur la page index.html

## Code JavaScript :

```
78     imageArticle.src = produit.imageUrl;  
79     nomArticle.textContent = produit.name;  
80     description.textContent = produit.description;  
81     prix.textContent = produit.price/100 + ",00€";
```

## Page index.html :



Résultat du test : Réussi

# Redirection vers la page du produit

## Objectif du test :

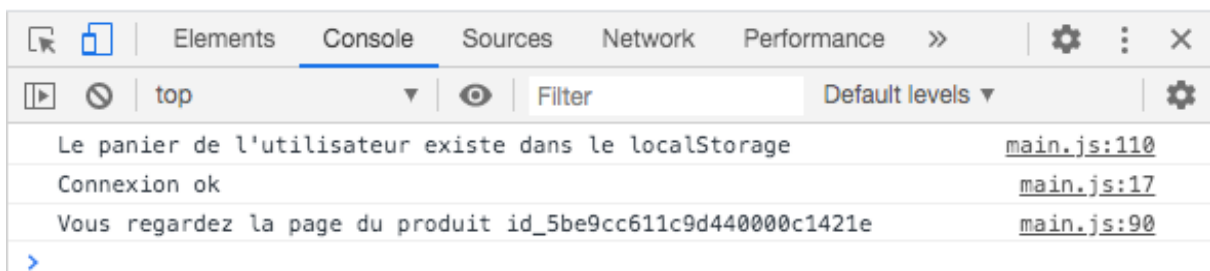
Vérifier que nous atterrissons bien sur la page du produit sélectionné au clic sur le bouton « Découvrir ».

## Code JavaScript :

```
50      let lienArticle = document.createElement("a");
63      lienArticle.setAttribute("href", "produit.html?id=" + produit._id);
82      lienArticle.textContent = "Découvrir";
```

Une fois que nous avons cliqué sur le bouton « découvrir », nous sommes bien redirigés vers la page du produit et la console doit retourner la réponse « Vous regardez la page du produit id\_xxx » grâce à la commande `console.log("Vous regardez la page du produit id_" + produitSelected._id);`

## Console :



**Résultat du test : Réussi**

# Affichage des détails du produit

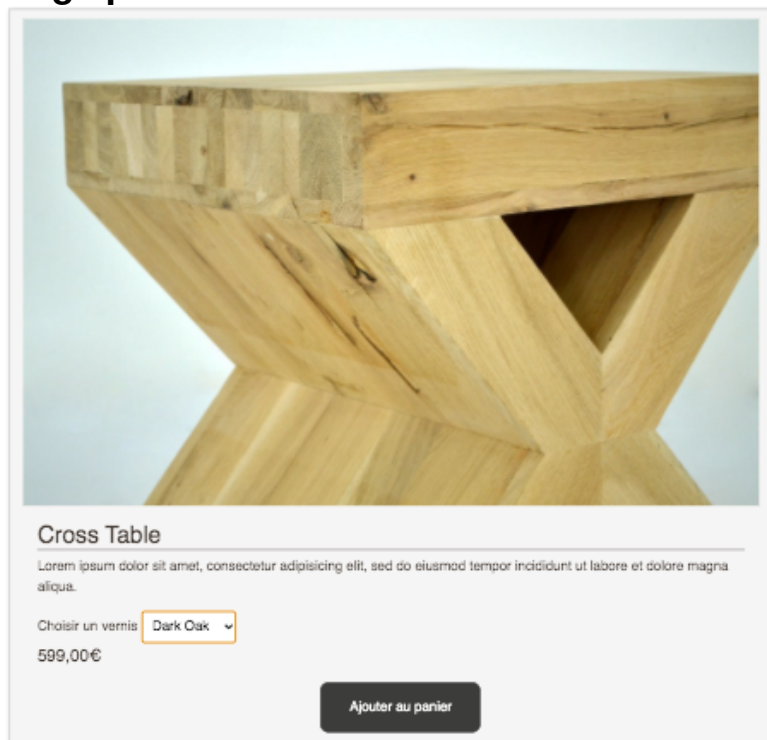
## Objectif du test :

Vérifier que les détails du produit s'affichent bien sur la page « produit.html » et que l'option de personnalisation est présente.

## Code JavaScript :

```
87  async function detailProduit(){
88      idProduit = location.search.substring(4);
89      const produitSelected = await getProduits();
90      console.log("Vous regardez la page du produit id_" + produitSelected._id);
91
92      /* Ajout du contenu des éléments html sur la page produit.html */
93      document.getElementById("img_product").setAttribute("src", produitSelected.imageUrl);
94      document.getElementById("name_product").innerHTML = produitSelected.name;
95      document.getElementById("description_product").innerHTML = produitSelected.description;
96      document.getElementById("price_product").innerHTML = produitSelected.price / 100 + ",00€";
97
98      /* Création de l'option de personnalisation du produit sur la page produit.html */
99      produitSelected.varnish.forEach((produit) => {
100          let optionProduit = document.createElement("option");
101          document.getElementById("option_select").appendChild(optionProduit).innerHTML = produit;
102      });
103  };
```

## Page produit.html :



## Résultat du test : Réussi

# Ajout d'un produit au panier

## Objectif du test :

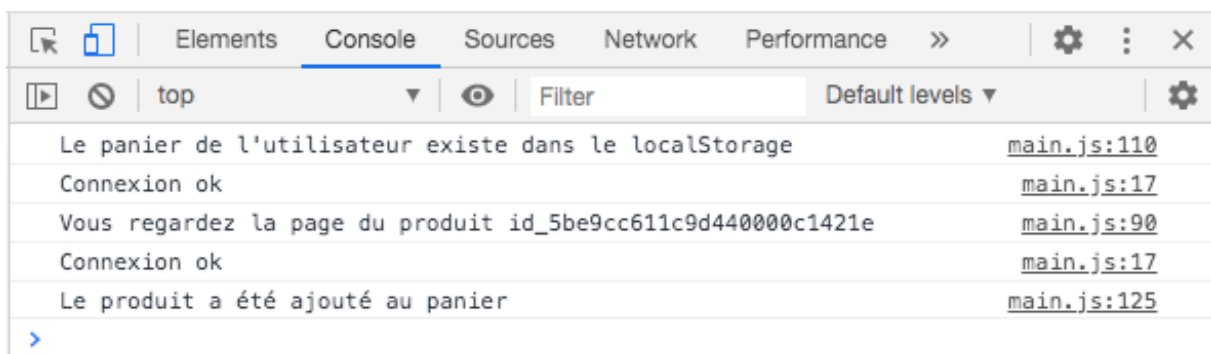
Vérifier que le produit est bien ajouté au panier, dans la page « panier.html », au clic sur le bouton « ajouter au panier ».

## Code JavaScript :

```
119 function addPanier(){
120     let inputBuy = document.getElementById("add_product");
121     inputBuy.addEventListener("click", async function() {
122         const produits = await getProduits();
123         userBasket.push(produits);
124         localStorage.setItem("userBasket", JSON.stringify(userBasket));
125         console.log("Le produit a été ajouté au panier");
```

La console doit retourner « Le produit a été ajouté au panier » grâce à la commande `console.log("Le produit a été ajouté au panier");`

## Console :



**Résultat du test : Réussi**



# Confirmation d'ajout d'un produit au panier

## Objectif du test :

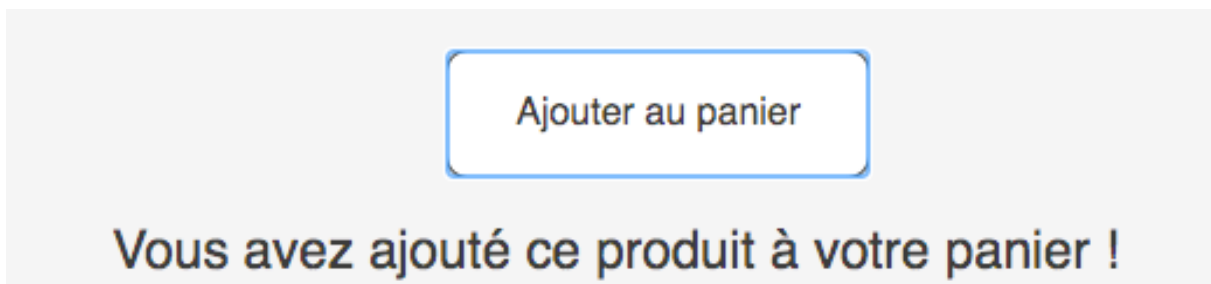
Vérifier que l'utilisateur a bien un message de confirmation quand il ajoute un produit au panier en cliquant sur le bouton « ajouter au panier ».

## Code JavaScript :

```
126     setTimeout(function() {  
127         document.getElementById('add_done').textContent = "Vous avez ajouté ce produit à votre panier !";  
128     }, 500);  
129     function add_done_remove(){  
130         document.getElementById("add_done").textContent="";  
131     }  
132     window.setTimeout(add_done_remove, 4000);  
133 }
```

Au clic sur le bouton « Ajouter au panier » nous avons bien un message de confirmation.

## Page produit.html :



## Résultat du test : Réussi

# Affichage des produits ajoutés au panier

## Objectif du test :

Vérifier que les produits ajoutés au panier sont bien affichés dans la page « panier.html ».

## Code JavaScript :


Structure du panier :

```
142 let facture = document.createElement("table");
143 let ligneTableau = document.createElement("tr");
144 let colonneNom = document.createElement("th");
145 let colonnePrixUnitaire = document.createElement("th");
146 let colonneRemove = document.createElement("th");
147 let ligneTotal = document.createElement("tr");
148 let colonneRefTotal = document.createElement("th");
149 let colonnePrixPaye = document.createElement("td");
```

Ligne produit :

```
162 JSON.parse(localStorage.getItem("userBasket")).forEach((produit)=>{
163   let ligneProduit = document.createElement("tr");
164   let nomProduit = document.createElement("td");
165   let prixUnitProduit = document.createElement("td");
166   let removeProduit = document.createElement("i");
```

## Page panier.html :

Nom du produit	Prix du produit	
Cross Table	599 €	
Coffee Table	899 €	
Dining Table (extendable)	1099 €	
<b>Total à payer</b>	<b>2597,00€</b>	

Résultat du test : Réussi

# Calcul du panier

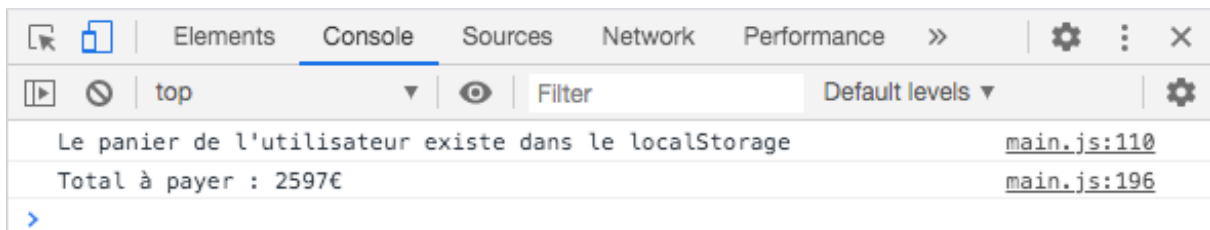
## Objectif du test :

Vérifier que le total du panier s'affiche bien et que la somme est correcte.



## Code JavaScript :

```
187     facture.appendChild(ligneTotal);
188     ligneTotal.appendChild(colonneRefTotal);
189     colonneRefTotal.textContent = "Total à payer";
190     ligneTotal.appendChild(colonnePrixPaye);
191     colonnePrixPaye.setAttribute("id", "total_sum");
192     let totalPaye = 0;
193     JSON.parse(localStorage.getItem("userBasket")).forEach((produit) => {
194         totalPaye += produit.price / 100;
195     });
196     console.log(`Total à payer : ${totalPaye}€`);
197     document.getElementById("total_sum").textContent = `${totalPaye},00€`;
198 }
```

## Console :



## Page panier.html :

Nom du produit	Prix du produit	
Cross Table	599 €	
Coffee Table	899 €	
Dining Table (extendable)	1099 €	
<b>Total à payer</b>	<b>2597,00€</b>	

Résultat du test : Réussi

# Suppression d'un produit du panier

## Objectif du test :

Vérifier que le panier est bien mis à jour lors de la suppression d'un produit.

La fonction est déclenchée par le clic sur la corbeille de la ligne du produit.

## Code JavaScript :

```
166 let removeProduit = document.createElement("i");
```

```
202 function annulerProduit(i){
203     console.log(`Enlever le produit à l'index ${i}`);
204     userBasket.splice(i, 1);
205     console.log(`Admin : ${userBasket}`);
206     localStorage.clear();
207     console.log(`localStorage vidé`);
208     localStorage.setItem('userBasket', JSON.stringify(userBasket));
209     console.log(`localStorage mis à jour`);
210     window.location.reload();
211 };
```

Au clic sur la corbeille, la page est bien rechargée et le panier est bien mis à jour avec le nouveau montant total.

## Page panier.html :

Nom du produit	Prix du produit	
Coffee Table	899 €	
Dining Table (extendable)	1099 €	
<b>Total à payer</b>	<b>1998,00€</b>	

Résultat du test : Réussi

# Vérification des données du formulaire

## Objectif du test :

Vérifier que les données saisies dans le formulaire sont correctes.  
Exemple : pas de chiffre dans le champ « nom ».

## Code JavaScript :

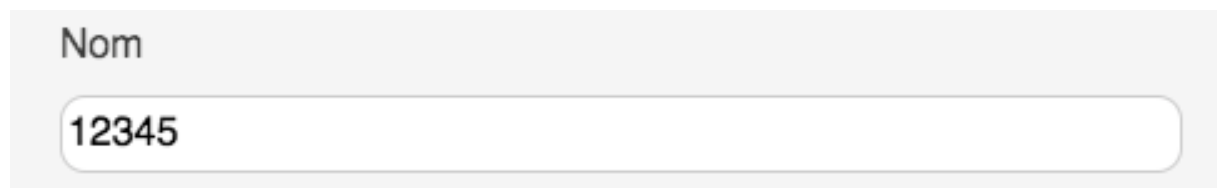
Configuration de la regex pour le champ « nom » :

```
217 let checkString = /^[a-zA-Z ,.'-]/;
```

Vérification du champ :

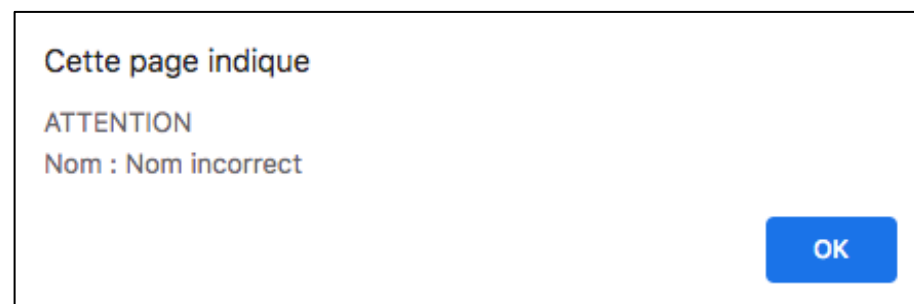
```
230 if(checkString.test(formNom) == false){  
231     checkMessage = "Nom : Nom incorrect";  
232 }else{  
233     console.log("Nom ok");  
234 };
```

Si le champ nom du formulaire contient des chiffres :



A screenshot of a web form. It features a label 'Nom' above a text input field. The input field contains the text '12345'.

Un message d'erreur s'affiche :



A screenshot of an error message dialog box. The text inside reads: 'Cette page indique', 'ATTENTION', and 'Nom : Nom incorrect'. There is a blue 'OK' button in the bottom right corner.

**Résultat du test : Réussi**

# Validation de commande

## Objectif du test :

Vérifier que la requête POST envoie bien les objets « contact » et « products » au serveur et en cliquant sur le bouton « Je valide ma commande » de la page panier.html.

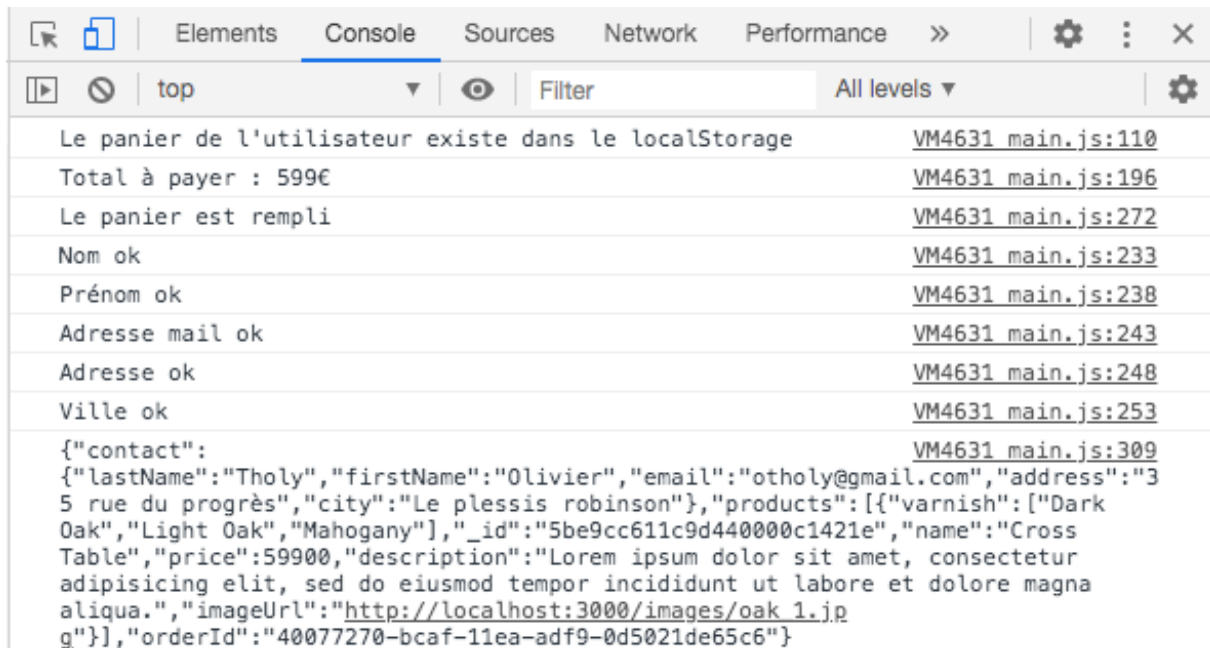
La validation de la commande doit entraîner la redirection vers la page de confirmation de commande.

## Code JavaScript :

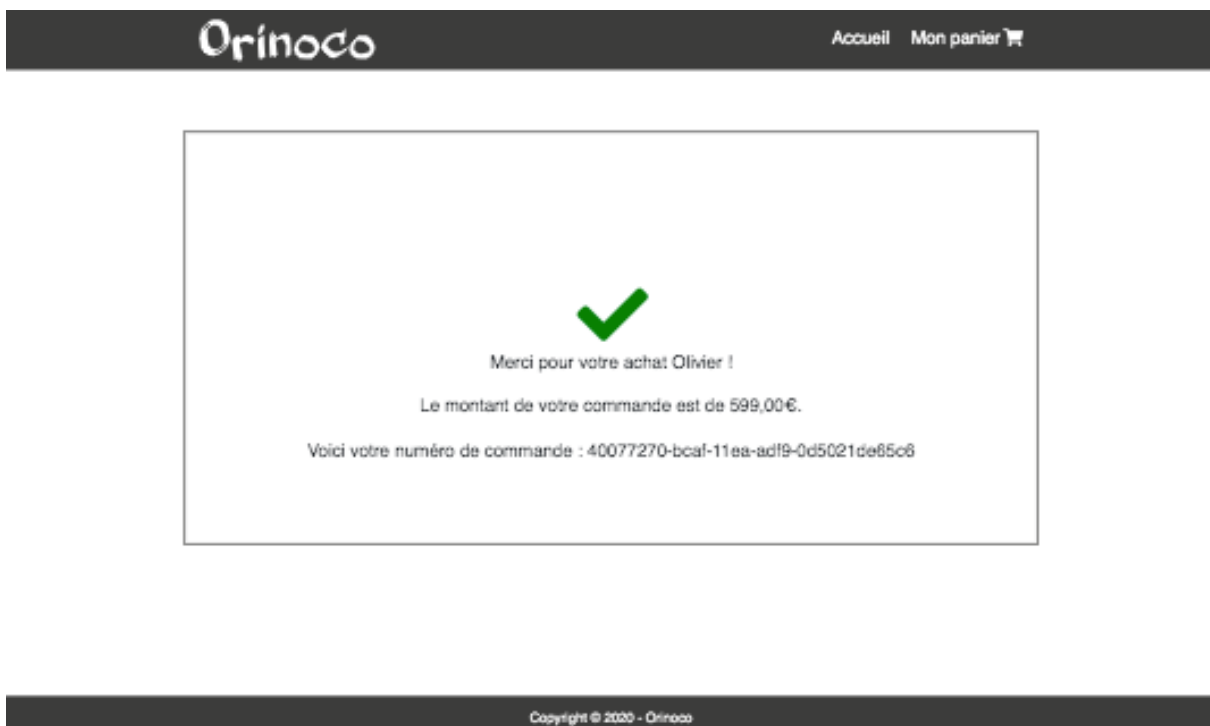
```
285  function validForm(){
286      let envoiPost = document.getElementById("envoiPost");
287      envoiPost.addEventListener("click", function(event){
288          event.preventDefault();
289      /* Si le panier est rempli et que les inputs du formulaire sont correctement remplis
290      if(checkPanier() == true && checkInput() == true){
291          let contact = {
292              lastName: document.getElementById("formNom").value,
293              firstName: document.getElementById("formPrenom").value,
294              email: document.getElementById("formMail").value,
295              address: document.getElementById("formAdresse").value,
296              city: document.getElementById("formVille").value
297          }
298          let objet = {
299              contact,
300              products
301          };
302      /* Envoi des objets vers l'API pour redirection vers la page confirmation.html */
303      let objetRequest = JSON.stringify(objet);
304      var request = new XMLHttpRequest();
305      request.open("POST", "http://localhost:3000/api/furniture/order");
306      request.setRequestHeader("Content-Type", "application/json");
307      request.onreadystatechange = function() {
308          if (this.readyState == XMLHttpRequest.DONE){
309              console.log(this.responseText);
310              localStorage.setItem('order', this.responseText);
311              window.location.href = "./confirmation.html";
312          }
313      }
314      request.send(objetRequest);
315      }else{
316          console.log("Admin : ERROR");
317      };
```

Dans la console, nous devons retrouver nos champs de « contact » du formulaire ainsi que les produits envoyés vers le serveur.

## Console :



Nous sommes bien redirigés vers la page confirmation.html :



Résultat du test : Réussi

# Confirmation de commande

## Objectif du test :

Vérifier que les éléments suivant sont bien présents sur la page de confirmation de commande.

- Remerciement de l'utilisateur
- Montant total de la commande
- ID de la commande

## Code JavaScript :

```
324 ▼ function resultOrder(){
325 ▼   if(localStorage.getItem("order") != null){
326     let order = JSON.parse(localStorage.getItem("order"));
327     document.getElementById("firstName").innerHTML = order.contact.firstName;
328     document.getElementById("orderId").innerHTML = order.orderId;
329     let priceOrder = 0;
330     let displayPrice = order.products;
331     displayPrice.forEach((element) =>{
332       priceOrder += element.price / 100;
333     });
334     document.getElementById("priceOrder").innerHTML = priceOrder;
335     console.log(order);
```

## Page confirmation.html :



Merci pour votre achat Olivier !

Le montant de votre commande est de 599,00€.

Voici votre numéro de commande : 40077270-bcaf-11ea-adf9-0d5021de65c6

## Résultat du test : Réussi