**[Installation de l'application web](https://florian-lepretre.herokuapp.com/teaching/web/tp6" \l "toc5)**

Extraire l'archive tp6.zip fournie.

Le répertoire lego-webstore/ contient les fichiers nécessaires à l'application web.

Installer les dépendances avec les commandes suivantes.

cd lego-webstore/

npm install

Démarrer l'application web en mode DEBUG.

DEBUG=lego-webstore:\* npm start

Dans un navigateur, accéder à l'url localhost:3000. Vérifiez que tout fonctionne correctement. Vous devriez obtenir l'affichage suivant.

Page d'accueil

[**Installation de *nodemon***](https://florian-lepretre.herokuapp.com/teaching/web/tp6#toc6)

Chaque changement que vous effectuerez dans l'application *Express* ne sera pas visible tant que le serveur n'aura pas redémarré. Nous allons automatiser ce redémarrage dès qu'une modification sera effectuée grâce à nodemon.

Installer nodemon en spécifiant que c'est une dépendance *développeur*.

npm install nodemon --save-dev

Dans package.json, modifiez la section script comme ci-dessous :

"scripts": {

"start": "node ./bin/www",

"devstart": "nodemon ./bin/www"

}

Couper le serveur (Ctrl+C) s'il est toujours en cours d'exécution, puis le relancer avec la commande suivante :

DEBUG=lego-webstore:\* npm run devstart

[**Routage**](https://florian-lepretre.herokuapp.com/teaching/web/tp6#toc7)

Les diverses routes de l'application seront définies dans le script routes/router.js. Actuellement, seule la route principale / est gérée.

*/\* GET home page. \*/*

router.get('/', indexController.index);

Lorsqu'un utilisateur visite localhost:3000/, la fonction associée à la route / est invoquée (*cf.* indexController.index).

Les contrôleurs de l'application web sont implémentés dans le répertoire controllers/. Pour l'instant, le contrôleur indexController.js implémente une fonction index qui renvoie du texte brut au navigateur.

exports.index = function(req, res) {

res.send('Page d\'accueil');

};

Vous pouvez essayer de changer le texte qui est renvoyé par ce contrôleur, puis vérifier que les modifications sont correctement prises en compte.

[**Question 1**](https://florian-lepretre.herokuapp.com/teaching/web/tp6#toc8)

Créer un nouveau contrôleur controllers/productsController.js. Ce contrôleur implémentera des fonctions relatives à la gestion des produits de la boutique.

Dans ce contrôleur, exporter une fonction products\_list. Cette fonction doit renvoyer au navigateur le texte brut "Liste des produits".

Dans le routeur routes/router.js, gérer la route /produits en mode get. Associer à cette route la fonction products\_list du contrôleur précédemment développé.

Dans un navigateur, l'adresse localhost:3000/produits doit maintenant afficher :

Liste des produits

Inspirez vous des implémentations déjà fournies pour la gestion de la route principale pour réussir cette question.

[**Question 2**](https://florian-lepretre.herokuapp.com/teaching/web/tp6#toc9)

Dans le contrôleur controllers/productsController.js, exporter une fonction product\_infos.

Cette fonction doit renvoyer au navigateur le texte brut "Fiche produit : [idproduct]", où [idproduct] sera un paramètre passé dans l'URL.

Dans le routeur routes/router.js, gérer la route /produits/:idproduct en mode get. Associer à cette route la fonction product\_infos précédemment développée.

Dans un navigateur, l'adresse localhost:3000/produits/42 doit maintenant afficher :

Fiche produit : 42

[**Question 3**](https://florian-lepretre.herokuapp.com/teaching/web/tp6#toc10)

Créer un nouveau contrôleur controllers/userController.js. Ce contrôleur implémentera des fonctions relatives à la gestion des comptes utilisateurs.

Dans ce contrôleur, exporter une fonction sign\_up. Cette fonction doit renvoyer au navigateur le texte brut "Formulaire d'inscription".

Dans le routeur routes/router.js, gérer la route /inscription en mode get. Associer à cette route la fonction sign\_up du contrôleur précédemment développé.

Dans un navigateur, l'adresse localhost:3000/inscription doit maintenant afficher :

Formulaire d'inscription

[**Question 4**](https://florian-lepretre.herokuapp.com/teaching/web/tp6#toc11)

Vérifier que la visite d'une URL incorrecte renvoie une erreur 404.

**Partie 2**

[**Question 1**](https://florian-lepretre.herokuapp.com/teaching/web/tp4#toc5)

Lier le script js/produit.js à la page html/produit.html. Si la liaison est correctement effectuée, vous devriez observer un Hello world! dans la console.

Ouvrez les outils de développement avec le raccourci Ctrl+Shift+I ou la touche F12 sur la plupart des navigateurs.

[**Question 2**](https://florian-lepretre.herokuapp.com/teaching/web/tp4#toc6)

Sur la page produit.html, on peut modifier la quantité commandée. Pour cela, on peut cliquer sur le bouton + pour augmenter la quantité, ou sur le bouton - pour la diminuer.

Attention, la quantité ne peut pas être nulle ou négative !

[**Question 3**](https://florian-lepretre.herokuapp.com/teaching/web/tp4#toc7)

La page produit.html affiche une image du produit. Des miniatures sont également disponibles. Si l’utilisateur clique sur l’une des miniatures, alors l’image principale est changée au profit de l’image correspondante à la miniature.

[**Question 4**](https://florian-lepretre.herokuapp.com/teaching/web/tp4#toc8)

Créer un script js/panier.js et le lier à la page panier.html.

On veut calculer automatiquement le prix du panier, en fonction des prix des articles qu’il contient, de leur quantité, et du taux de TVA (fixé à 15%).

[**Question 5**](https://florian-lepretre.herokuapp.com/teaching/web/tp4#toc9)

On peut diminuer la quantité d'un article du panier lorsque l’on clique sur le bouton *Supprimer* correspondant. La quantité diminue de 1 à chaque clic, jusqu'à atteindre 0, auquel cas l'article est supprimé du panier.

Le prix du panier doit être actualisé à chaque suppression.

[**Question 6**](https://florian-lepretre.herokuapp.com/teaching/web/tp4#toc10)

Dans produit.html, un utilisateur peut écrire un avis dans la zone de saisie. Ajouter le code HTML suivant sous la section des *Avis*.

<div class="input-group">

<div class="input-group-addon">

<span class="oi oi-pencil"></span>

</div>

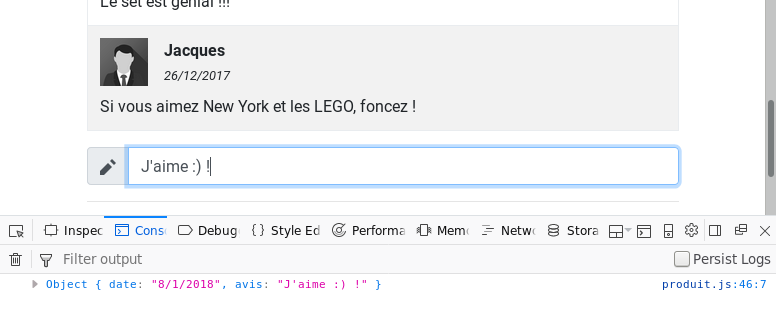
<input type="text" id="avis" class="form-control" placeholder="Rédiger un avis" />

</div>

Lorsqu’on appuie sur la touche Entrée :

* Créer un objet contenant la date et le contenu de l’avis, puis l'afficher dans la console,
* Vider la zone de saisie.

Code de la touche Entrée : 13



[**Q**](https://florian-lepretre.herokuapp.com/teaching/web/tp4#toc11)