

# Travail écrit Web Mob. UI 2022 - M49/1 - 28.03.2022

- Tout document autorisé.
- Utilisation de l'ordinateur restreinte à l'utilisation d'un lecteur de PDF, un éditeur de code, un browser limité au site Internet du cours (support de cours, liens de premier niveau, et exercices compris) et au localhost.
- Pas d'autre communication réseau (ou directe) autorisée pendant le travail.
- **Toute non-observation des règles ci-dessus entraîne la note 1 et supprime toute possibilité de remédiation.**

Vous rendrez **le dossier complet** de votre projet par **email**, via un **ZIP** portant votre nom, à l'adresse [loris.gavillet@heig-vd.ch](mailto:loris.gavillet@heig-vd.ch). Vous êtes **responsable** de son contenu, et devez vous **assurer** de la bonne réception du dossier avant de quitter la salle. Le travail et la feuille de donnée sont à rendre **au plus tard à 16h15**.

## MISE EN PLACE ET STRUCTURE (15 MINUTES)

Vous allez réaliser une petite application Web contenant un shop en ligne, appelée "Zalansol". Il est possible d'y afficher une liste de produits, voir la vue détaillée de chaque produit et afficher une liste de produits pour une catégorie donnée. Une gestion simplifiée de caddie permet d'ajouter ou non un article dans le panier et un onglet affiche son contenu.

Une structure de base vous est donnée, contenant un projet webpack, le markup HTML/CSS, des fichiers JS et autres assets nécessaires au développement. Vous pouvez la télécharger via ce lien <https://github.com/lgavillet/webmobui-22-exa-a>. Une fois téléchargé, effectuez un "npm install" et vous pourrez ensuite démarrer le projet avec "npm run start", comme vu en cours.

Le fichier src/index.js contient la logique de base pour naviguer entre les sections, selon le concept du hashchange vu ensemble. Pour la structure des hashes, vous devez utiliser les URLs suivantes:

- #products - Liste de tous les produits
- #products-[id] - Vue détaillée d'un produit (où [id] est à remplacer par l'id du produit. Ex.: #products-1234)
- #categories-[id] - Liste des produits d'une catégorie (où [id] est à remplacer par l'id de la catégorie. Ex.: #categories-1234)
- #cart - Vue du panier

Une section Produits est disponible avec un markup pour le listing de produits que vous pouvez réutiliser pour les URLs ci-dessus.

## 1 AFFICHER LA LISTE DE TOUS LES PRODUITS (30 MINUTES)

La liste de tous les produits présents sur le serveur est obtenue grâce à l'URL <https://webmobui-22-exa-backend.herokuapp.com/api/products> et devra se trouver sur le hash "#products".

Vous pouvez créer une fonction d'aide qui fait le fetch et renvoie un tableau de produits dans le fichier src/api.js et placer la logique d'affichage dans un autre fichier, comme par exemple src/sections/products.js. Servez-vous ensuite du markup HTML existant de la section "products-section" (products au pluriel) pour en extraire un template pour chaque élément de la liste, comme vu en cours. Il vous servira alors à construire la liste des produits à afficher et remplacer leur contenu.

Le titre de la section doit être "Produits".

**NB:** Il y a deux liens dans chaque produit: celui qui entoure l'image sert à afficher le détail du produit et celui qui contient le nom de la catégorie, sert à afficher les produits qu'elle contient.

## 2 AFFICHER LA LISTE DE PRODUITS D'UNE CATÉGORIE (20 MINUTES)

En réutilisant une bonne partie du code précédent, vous devez à présent gérer l'affichage des produits d'une catégorie. Cela arrive lorsque l'on clique sur le nom d'une catégorie, avec le lien "#categories-[id]" ou depuis la page home. L'endpoint à utiliser est le suivant: [https://webmobui-22-exa-backend.herokuapp.com/api/categories/\[id\]/products](https://webmobui-22-exa-backend.herokuapp.com/api/categories/[id]/products) où [id] est l'id de la catégorie souhaitée.

Le titre de la section doit être "Produits > [nom de la catégorie]".

## 3 AFFICHER LES DÉTAILS D'UN PRODUIT (20 MINUTES)

L'affichage des détails d'un produit se fait sur le hash '#products-[id]' et vous pouvez récupérer les détails d'un produit via le endpoint [https://webmobui-22-exa-backend.herokuapp.com/api/products/\[id\]](https://webmobui-22-exa-backend.herokuapp.com/api/products/[id]).

Pour les détails d'un produit, vous pouvez utiliser le markup de la section product-section (product au singulier).

Le titre de la section doit être "Produits > [nom du produit]".

## 4 AFFICHER LES CATÉGORIES DE PRODUIT SUR LA SECTION HOME (20 MINUTES)

Sur la section home, se trouve une liste de catégories fictives, à remplacer par la liste des catégories retournée par le serveur. L'endpoint à utiliser est : <https://webmobui-22-exa-backend.herokuapp.com/api/categories>.

Servez-vous ensuite du markup HTML existant de la section "home-section" pour en extraire un template pour chaque élément de la liste, comme vu en cours. Il vous servira alors à construire la liste des catégories à afficher et remplacer leur contenu.

## 5 GESTION DE L'AJOUT AU PANIER (20 MINUTES)

Pour permettre d'ajouter un article au panier, servez-vous de la librairie JsonStorage que nous avons vu ensemble, déjà présente dans "src/lib/JsonStorage.js". Il s'agit d'une gestion de panier on/off, un article y est présent ou non. Pas de quantité à gérer, ni possibilité d'ajouter plusieurs fois le même item.

Pour cela, créer un event listener au clic sur le bouton panier de chaque élément de la liste qui va gérer son ajout/suppression du panier, ainsi que changer l'icône du panier.

## 6 AFFICHAGE DU PANIER (20 MINUTES)

L'affichage du panier se fait sur le hash '#cart' et doit simplement lister les produits choisis.

Lorsque l'on est dans la vue panier et que l'on enlève un élément du panier, on s'attend à ce que celui-ci soit directement enlevé de la liste (uniquement vue panier...).

Le titre de la section doit être "Panier ([taille])" où taille représente le nombre d'éléments dans le panier. Exemple "Panier (2)".

## 7 AFFICHAGE MOBILE (15 MINUTES)

En version mobile, la liste des produits avec trois éléments par ligne est beaucoup trop petite. À l'aide d'une media query assurez-vous que lorsque l'écran fait moins de 767px de large, les éléments s'affichent simplement en colonne les uns après les autres.

## APPLICATION PWA (5 MINUTES)

À faire en dernier pour éviter des problèmes lors du développement. Une fois votre application prête, transformez-la en PWA en liant un fichier manifest et en enregistrant un service worker pour faire du caching. Vous pouvez vous servir des fichiers manifest.webmanifest et workerCacheFetched.js présents dans public, comme vu en cours. Vérifiez votre intégration grâce à l'inspecteur.

## RÉSERVE ET RELECTURE (15 MINUTES)