

## programmation web en js

### projet Photobox : partie 1

#### **Photobox : parcourez et visualisez des galeries de photos**

Photobox est une application de parcours et de visualisation de photos, permettant d'afficher et de faire défiler des galeries de photos paginées. Un mode "lightbox", où chaque photo occupe l'ensemble de la fenêtre navigateur est également disponible et permet de parcourir les galeries page par page.

#### **Préliminaires : Accéder aux données Photobox**

Les données (principalement : les catégories, les photos et leur description) photobox sont accessibles au travers d'une interface de programmation (API) qui retourne les données en format json.

Une documentation de l'API est disponible à l'url :

<https://webetu.iutnc.univ-lorraine.fr/www/canals5/photobox/doc>

Pour tester cette api avant de programmer son utilisation en javascript, vous devez installer une extension dans votre navigateur :

- Postman sur chrome
- RestClient sur Mozilla

Vous pouvez également, sur votre machine personnelle, installer une des 2 applications suivantes :

- Postman
- insomnia

A l'aide de cette extension, accédez aux données décrivant :

- une liste de photos : envoyer une requête de type GET sur l'url : <https://webetu.iutnc.univ-lorraine.fr/www/canals5/photobox/photos>
- une photo : envoyer une requête de type GET sur l'url <https://webetu.iutnc.univ-lorraine.fr/www/canals5/photobox/photos/12>

#### **Exercice 1 : afficher 1 liste de photos**

Créer un programme js/jquery qui affiche une liste de photos sous forme de vignettes. La liste de photos est obtenue grâce à une requête ajax auprès de l'api photobox.

Procédez par étapes :

1. en vous basant sur le document html fourni, qui contient une liste de photos en html, mettez en forme (css) votre galerie de photos (voir exemple en annexe),
2. créez un module nommé `photoLoader` pour gérer les requêtes vers l'api. Ce module doit exposer une interface comprenant 2 méthodes :
  - une méthode d'initialisation, qui reçoit en paramètre l'adresse du serveur hébergeant l'api ( "server\_url", cad : <https://webetu.iutnc.univ-lorraine.fr>),
  - une méthode de chargement d'une liste d'objets, qui reçoit en paramètre une uri (par exemple `"/www/canals5/photobox/photos/?offset=8&size=12"`), effectue la requête ajax, traite les erreurs éventuelles et retourne une promesse,
3. ajoutez un bouton pour charger une galerie dans votre document html, dans la partie "nav",

4. créez un module nommé `gallery` pour charger et construire une galerie de photos. Ce module propose l'interface suivante :
  1. une méthode d'initialisation, qui stocke les paramètres de configuration : notamment identifiant de la galerie ...,
  2. une méthode de chargement des données et insertion de la galerie dans le dom ; cette méthode utilise le module `photoLoader` pour charger les données, et transmet ces données à une fonction privée qui génère et insère le dom correspondant à la galerie.
5. Créer le module principal de l'application qui, lorsque le DOM est chargé, réalise les différentes initialisations, puis associe le "click" sur le bouton de chargement d'une galerie à l'action correspondante.

## **Exercice 2 : lightbox : affichage d'une photo**

Lorsque l'utilisateur clique sur une photo de la galerie, on souhaite afficher cette photo en format original dans une "lightbox", c'est à dire dans un cadre venant recouvrir la galerie (voir exemple en annexe). La galerie doit rester présente, pour pouvoir réapparaître lorsque l'utilisateur quitte la lightbox. Pour réaliser cette fonctionnalité, aucune requête à l'api n'est nécessaire puisque la galerie contient, pour chaque vignette, l'url de la photo originale comme valeur de l'attribut `data-img`.

1. Mettez au point l'aspect visuel de la lightbox en vous basant sur le code html fournit dans le squelette de démarrage. Ajoutez les règles css qui vous permettent de réaliser l'affichage qui vous convient. Une fois l'affichage obtenu, masquez la lightbox.
2. Créez un module `lightbox` qui propose une méthode pour créer et afficher une lightbox correspondant à une vignette de la galerie. Vous pouvez prévoir des méthodes privées, par exemple une méthode privée pour créer la lightbox, une autre pour l'afficher.
3. Enregistrez les handlers permettant l'activation de la lightbox lors du click sur une vignette de la galerie.

## **Exercice 3 : naviguer dans les galeries**

On souhaite compléter la navigation dans les galerie de photos en ajoutant la possibilité de parcourir les pages de photos.

1. examiner les données retournées par l'api pour la liste de photos et identifier comment, à partir de ces données, on peut obtenir la page suivante et la page précédente.
2. compléter le module `gallery` pour y ajouter les méthodes permettant de charger et afficher la page suivante/précédente de la galerie,
3. ajouter les boutons de navigation "next" et "prev" dans la zone de navigation de votre interface html.

