

**Formation React**

**Evaluation**



**zenika**  
ARCHITECTURE INFORMATIQUE

# Evaluation React

---

## Objectif: créer une application de gestion de séries

Vous venez d'être embauché dans une toute nouvelle startup « Netfli ». En tant que tout premier développeur de cette startup, on vous charge de créer l'application qui va révolutionner le monde des séries.

## Besoins

Voici les besoins qu'on vous exprime:

- Afficher les données de séries
- Ajouter/retirer une série de favoris
- Rechercher une série par texte
- Récupérer les séries depuis un serveur
- Bonus (optionels)

### Afficher les données des séries

Les données des séries vous sont fournis au format json. Vous devez récupérer ces données et les afficher dans votre application, sous la forme d'une liste de cartes par exemple (la quantité de données à afficher par série est laissée à votre jugement).

### Ajouter/retirer une série de favoris

Votre utilisateur doit pouvoir ajouter et retirer une série de ses favoris. Une série ajoutée en favori doit posséder un élément visuel permettant de la distinguer des autres.

### Rechercher une série par texte

Un champ doit être présent dans votre application. Votre utilisateur peut y taper du texte et filtrer les séries affichées dans votre application en fonction du texte renseigné. Ce filtre s'applique sur le titre des séries uniquement.

### Récupérer les séries depuis un serveur

Les données des séries doivent être récupérées depuis un serveur REST au lieu d'utiliser un fichier json dans l'application. Une fois que c'est le cas, le système de favoris doit lui aussi modifier les données des séries sur le serveur.

## Bonus

En bonus, vous pouvez si vous le souhaitez ajouter encore plus de fonctionnalités dans votre application. Ces fonctionnalités sont OPTIONNELLES, vous ne pouvez pas perdre de points si vous ne les réalisez pas. En réaliser quelques unes pourra uniquement augmenter votre note.

Voici une liste non exhaustive de fonctionnalités qui peuvent être ajoutées dans votre application.

### **Pouvoir n'afficher que les séries mises en favori**

Votre utilisateur doit pouvoir visualiser uniquement ses séries mises en favoris s'il le souhaite.

### **Permettre la suppression d'une série**

Votre utilisateur doit pouvoir supprimer une série de la liste de votre application s'il le souhaite.

### **Une autre page pour l'ajout de série**

Votre utilisateur doit pouvoir se rendre sur une autre page de votre application pour ajouter une nouvelle série (plutôt que de rester sur la même page). (voir la librairie react-router)

### **Une autre page pour visualiser une série**

Votre utilisateur doit pouvoir se rendre sur une autre page de votre application pour visualiser les données d'une série en particulier. (voir la librairie react-router)

### **Avoir un light/dark mode**

Votre utilisateur doit pouvoir basculer d'un thème light de votre application à un thème dark (et inversement) s'il le souhaite. (voir la notion de context)

## **Critères d'évaluations**

- Fonctionnalités implémentées (nombre, absence de bug). Les fonctionnalités "bonus" ne peuvent pas enlever de points, uniquement en rajouter. Implémenter une de ces fonctionnalités vous imposera de rechercher des sujets React qui n'ont pas été directement abordés pendant les cours. Donnez la priorité aux fonctionnalités "non bonus" demandées.
- Qualité technique de votre code
  - Lisibilité (nom des variables, indentation, etc)
  - Absence de warnings/Errors dans la console
  - Prop-types
  - Découpage en composants
- UI/UX de votre application

## **Contraintes techniques**

- Vous devez utiliser React pour construire votre application
- Vous êtes libres d'utiliser le CLI `create-react-app` pour générer la structure de base de votre application
- Vous êtes libres d'utiliser le framework css que vous souhaitez (ou même aucun).
- Vous êtes libres d'utiliser toute librairie js que vous trouverez pertinente d'inclure.

- Vous êtes libres d'utiliser toutes les ressources que vous trouverez sur le web ou ailleurs.
- Le code source de votre application doit se trouver sur votre compte Github.
- Votre application doit pouvoir être lancé avec un:

```
git clone lenomdevotreapplication.git
cd lenomdevotreapplication
npm i
npm start
```

## Ressources

Vous ai fourni un fichier json `shows.json` contenant une liste de séries. Servez vous de ce fichier tant que vous ne récupérez pas les séries depuis un serveur.

Vous ai fourni un serveur REST fournissant un certain nombre de fonctionnalités:

- Récupération des données d'une liste de séries

```
GET /rest/shows
```

- Récupération des données d'une série

```
GET /rest/shows/:id
```

- Mise à jour des données d'une série

```
PUT /rest/shows/:id
```

- Ajout d'une série

```
POST /rest/shows
```

- Suppression d'une série

```
DELETE /rest/shows
```

Pour démarrer le serveur

```
cd ledossierduserveur
npm i
npm start
```

Attention: arrêter le serveur reset la liste des règles qu'il fournit.

## Conseils

- Lorsque vous attaquerez la partie communication avec un serveur, prenez le temps de vous familiarisez avec l'api.

- Une fonctionnalité à la fois. Une application avec uniquement deux fonctionnalités mais correctement implémentées sera mieux valorisée qu'une application avec 4 fonctionnalités à moitié terminée.
- Servez vous de tous ce que vous avez déjà fait dans les travaux pratiques des cours.
- Votre code sera relu. Assurez vous qu'il soit le plus lisible possible. Si vous avez un doute, demandez à un de vos collègues de faire une relecture pour vous. Egalement, n'hésitez pas à ajouter des commentaires pour expliquer le pourquoi de votre code.