

# Développement de serveur web nodeJS (DRAFT)

*TD - "lib-express"*



## Description du projet

lib-express est une plateforme de blog.

Sur cette plateforme un minimum de fonctionnalités d'un blog classique vous sera demandé:

- Il est possible d'ajouter des articles depuis un backend
- Un système de catégories permet de classifier les articles
- L'utilisateur pourra consulter les articles depuis le frontend
- Une système de login mot de passe permet de se connecter au backend avec un utilisateur.

## Layouts frontend et backend

Pour le frontend et le backend il vous sera demandé d'utiliser le système de layout de PUG.

Deux layouts seront donc créées affichant un titre: lib-express pour le frontend, Dashboard - lib-express pour le backend.

## Fonctionnalités BackEnd

Il est possible d'accéder au backend de votre application depuis l'adresse **/dashboard**.

### Liste des articles

Sur l'adresse **/dashboard**, la liste des articles est affichée. Il sera aussi possible d'accéder à l'édition, la création ou la suppression d'article depuis cette page.

DASHBOARD - lib-express		
Création	Catégories	
Titre de l'article	Editer	Supprimer
Titre de l'article	Editer	Supprimer
Titre de l'article	Editer	Supprimer
Titre de l'article	Editer	Supprimer
Titre de l'article	Editer	Supprimer
Titre de l'article	Editer	Supprimer

←

→

## Pagination

Un maximum de 25 articles sera affiché. Au delà, un système de pagination permettra d'afficher les 25 suivants, etc.

Afin de naviguer les liens suivants auront une URL avec un paramètre page indiquant la page à atteindre, exemple: **/dashboard?page=4** ⇒ affichera les articles à partir du 100ème article.

## Accès à la création d'articles

Un bouton "Créer" en haut de la page de création d'article qui redirigera vers l'URL /article.

Pour plus d'information sur cette page, consulter le chapitre [Création, modification d'un article](#)

## Edition d'article

Quand l'utilisateur clique sur le bouton d'édition de l'article. Il est renvoyé vers la même template qui sera utilisé lors de la création à partir de l'url **/dashboard/article/:id**. Un bouton sauvegarder permet d'enregistrer les modifications. Un bouton Annuler permet de revenir vers le dashboard.

Pour éditer et créer le texte d'une page on utilisera le plugin Tiny MCE

<https://www.tiny.cloud/docs/>. Pour installation, veuillez vous référer à la page suivante:

<https://www.tiny.cloud/docs/general-configuration-guide/advanced-install/#sdkinstall>

Lors de l'édition d'une page nous appelons la méthode **PUT** sur le service

**/dashboard/article/:id**

Une fois la mise à jour effectuée nous retournons vers le dashboard.

Il est possible de sélectionner des catégories à partir d'une liste avec multi sélection.

Titre de l'article

File Edit View Format

↶ ↷

Paragraph ▾

**B** *I*

≡ ≡ ≡ ≡

≡ ≡

Hello, World!

POWERED BY TINY

Catégorie 1

Catégorie 2

Catégorie 3

Catégorie 4

Annuler

Sauvegarder

Sur la saisie d'écran ci-dessus on peut voir que l'utilisateur peut modifier le titre, utiliser le plugin TinyMCE pour éditer son article.

Les catégories 1 et 2 sont sélectionnées. Il suffit à l'utilisateur de cliquer sur une catégorie pour la supprimer ou l'ajouter à cet article.

### Suppression d'article

Quand l'utilisateur clique sur le bouton de suppression de l'article, une boîte de dialogue ou popup permet de confirmer la suppression de l'article. Si l'utilisateur confirme la suppression, le service **/dashboard/article/:id** sera appelé avec la méthode **DELETE**. Une fois la ressource supprimée la liste devra se rafraîchir.

### Création d'un article

Nous utiliserons ici la même page que pour l'édition et les mêmes fonctionnalités.

La seule différence sera sur le mode d'édition, si nous créons une page alors:

- Le bouton sauvegarder devient créer
- On utilisera la méthode **POST** sur le service **/dashboard/article/**

### Gestion des catégories

Cette page accessible depuis la liste des articles permet d'ajouter, éditer et supprimer les catégories.

Pour cela nous allons utiliser le même type de tableau que nous avons utiliser sur la liste des articles.

Sur chaque ligne est affiché le titre de la catégorie, un bouton supprimer et un bouton éditer.

Cette page sera disponible depuis l'url **/dashboard/categories**

Pour la suppression, l'édition et création l'API REST fonctionnera suivant le même principe que pour les articles. Il est possible de faire une autre vue que celle utilisée pour la liste des articles.

### Connexion

Un système de login permet de vous connecter sur l'URL **/dashboard/login**.

Un seul utilisateur sera disponible pour vous connecter: admin Il sera possible de modifier le mot de passe du compte admin (**procédure à définir**).

Un service **REST** sur l'adresse **/dashboard/login** permettra de vérifier le couple login mot de passe grâce à une requête **POST**. Si le mot de passe est correct, la réponse 200 sera renvoyé et un token sera donné au navigateur. Une redirection vers **/dashboard** est effectuée.

Si le mot de passe est incorrect, le code 401 sera retourné. un message d'erreur sera affiché sur la page. Et aucune redirection sera effectué.

Si le client n'a pas token dans son navigateur aucune adresse ayant le préfixe **/dashboard** ne sera disponible. Si le token ne peut être vérifier, l'utilisateur sera ramené sur la page de login.

## Fonctionnalités FrontEnd

Deux vues seront disponibles sur le front-end.

### Liste des articles

Une première permettra d'afficher la liste des articles sous forme résumés. Cette vue sera disponible sur la racine du serveur: <http://localhost>

Chaque article sera présenté dans un bloc contenant: un titre, les 200 premiers caractères de l'article. Un lien est disponible permettant de consulter le détail de l'article.

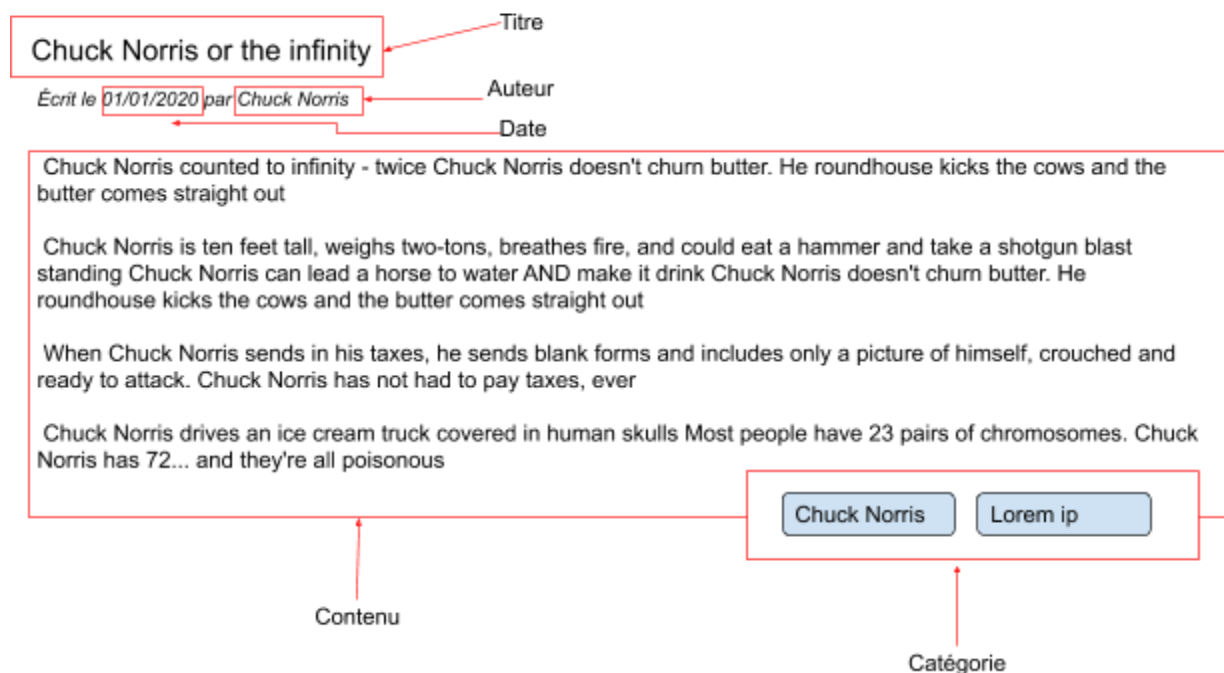
Ce lien amènera sur l'URL suivante: **/article/:id**

### Détail d'un article

En cliquant sur le lien d'un article il vous sera possible d'afficher la totalité de l'article.

Toutes les informations concernant l'article sont affichées: titre, contenu, date, auteur, et catégories

## lib-express



## Barème

- Qualité du code 4 points:
  - Gestion d'erreur
  - Nommage variables, fonctions, coding convention
  - DRY: do not repeat yourself
- Fonctionnalités de base 16 points:
  - Backend: 12 points
    - Gestion des articles 6
      - Création
      - Edition
      - Suppression
    - Gestion des catégories: 2
      - Création
      - Edition
      - Suppression
    - Authentification: 2
  - Frontend: 4 points
    - Liste des articles
      - Affichage des articles 1
      - Pagination 1
    - Détail de l'article: 2
      - Affichage du titre
      - Affichage de l'article
      - Date
      - Auteur
      - Catégories
- Fonctionnalités bonus 2 points pour les commentaires:
  - Backend: 1 point
    - Gestion des articles
    - Gestion des catégories
  - Frontend: 1 point
    - Liste des commentaires
    - Suppression de commentaire

Pour chaque fonctionnalité, il vous sera demandé de faire attention au rendu graphique de l'application et de la bonne utilisation de bootstrap. Il est donc nécessaire de faire un rendu compréhensible et agréable pour l'utilisateur.