

MASTER 2 SIGMA

Prototypage d'un site internet pour visualiser des récits d'esclaves américains



Guide de développement de l'application web : étude des possibilités techniques pour la suite du projet

Florent LAIDIN

Table des matières

Propos introductif	1
1. Page d'accueil (/map).....	2
2. Panneau latéral	2
1.1. Filtres	2
1.2. Légende	2
3. Page « visualisation des récits » (/map/recits)	3
3.1. Génération de routes par récit	3
3.2. Modification des pop-ups	3
3.3. Gestion des styles	3
4. Recherche par type de lieu (/map/places)	4
4.1. Génération de routes par récit	4
4.2. Ajout des géométries surfaciques.....	4
4.3. Style des clusters.....	4
5. Page liste des récits (/recits)	5
6. Fiche récit (/recits/id_recit)	5
7. Page contact (/contact).....	5
8. Choix de la langue	6
9. Gestion des authentifications	6
10. Commentaires généraux sur le prototype	6

Propos introductif

Ce guide de développement est à destination des futurs développeurs chargés de poursuivre ce projet. De nombreux éléments sont en effet incomplets voire manquants sur le prototype. Il s'agit donc ici de préciser ce qui peut être fait dans le futur et avec quels moyens, outils.

Des idées et conseils de développement sont donc présents selon chaque page en fonction de leur avancement. Enfin, des commentaires généraux, notamment sur les aspects de programmation et de clarté du code, clôtureront ce guide.

1. Page d'accueil (/map)

La page d'accueil dispose d'un certain nombre d'éléments (points de publication, barre latéral) mais peut être approfondie.

Tout d'abord, les pop-ups lorsque l'on clique sur chaque récit sont à personnaliser. On peut par exemple imaginer avoir en plus de la date de publication d'autres éléments comme l'affichage du début du titre. Cela serait possible en appelant le titre dans la table des récits et faisant un substring des 30 premiers caractères, par exemple. D'une manière générale, le style des pop-ups peut être repris en soignant un peu plus certains éléments tels que le bouton pour visualiser la carte.

2. Panneau latéral

1.1. Filtres

Pour ce qui est du panneau latéral, plusieurs filtres sont encore manquants par rapport à la commande initiale. Il s'agit des filtres sur le type du récit (écrit ou dicté), sur le continent (Afrique, Amérique, Europe) avec l'affichage de certaines couches selon le continent sélectionné. Ici, plusieurs choix sont possibles.

Les filtres peuvent être ajoutés avec la même méthode en créant des formulaires qui sont appelés dans le contrôleur avec ensuite des conditions selon le choix du formulaire. Mais cela présente une grosse limite qui est de ne pas avoir la possibilité de choisir plusieurs filtres en même temps tels que les lieux de naissance et les récits dictés.

Le second choix est donc de créer un grand formulaire qui prend en compte tous les filtres (menus déroulants). Ainsi on peut sélectionner différents types de filtres et les exécuter en même temps. Mais cette méthode est plus complexe à mettre en place. Il faut alors récupérer dans le contrôleur les valeurs des différents selects pour aller chercher respectivement les données nécessaires avec les modèles.

Une barre de recherche au sein du panneau latéral serait également appréciable, avec un affichage des différents choix possibles au sein de la table des récits. L'exécution de la recherche se ferait dans le même grand formulaire cité précédemment. Cette barre de recherche serait un input de type « search », le nom de l'esclave serait récupéré avec une méthode « post ».

1.2. Légende

Pour ce qui est de la légende, elle est actuellement statique au sein du panneau latéral. Il manque certains éléments à ajouter avec notamment les royaumes africains et aires autochtones amérindiennes.

Sur ce point, le problème de la place prise par la légende pourrait poser problème à l'avenir. Il serait donc préférable pour la suite de l'insérer en dessous de la carte ou de la masquer en dessous de la page et de l'afficher seulement si l'utilisateur clique sur un bouton « voir la légende ». Cela pourrait faire gagner de la place en rendant le site plus aéré.

Une autre possibilité serait de générer une légende dynamique avec, par exemple, l'affichage du nom des royaumes africains au survol avec la souris. Ceci est possible avec le développement de fonctions utilisant des « layer.on » et l'obtention des propriétés de la couche, par exemple.

3. Page « visualisation des récits » (/map/recits)

La page « visualisation des récits » est accessible lorsque l'on sélectionne un récit pour accéder à la carte de celui-ci.

3.1. Génération de routes par récit

Tout d'abord, un premier point peut être fait sur la création de la page et de sa route. En effet, pour chaque sélection de récit, l'utilisateur est renvoyé sur « /map/recits » peu importe le récit. Cela est fonctionnel mais pas optimal. Le plus intéressant serait d'avoir une route propre à chaque récit avec à la fin son identifiant, comme pour la génération des fiches récits.

3.2. Modification des pop-ups

Sur cette page, la personnalisation des pop-ups est à reprendre à la fois sur les points mais également sur les géométries surfaciques. Tout d'abord, le pop-up lorsqu'on clique sur un point pourrait contenir comme titre le nom de l'esclave avec en dessous le nom du lieu et le type de lieu.

Si ces informations sont déjà connues avec la légende et les tooltips, il est toujours appréciable de les retrouver en cliquant dessus. J'ajouterais qu'un lien sur le type de lieu pourrait être mis en place dans le pop-up, cela permettrait de rediriger l'utilisateur sur tous les lieux similaires à celui-ci.

De même pour les surfaciques (pays, états, régions), ces informations sont à retrouver sur un pop-up. Pour les surfaciques, les noms de lieux ne sont pas tous contenus dans le même champ, il va donc falloir les regrouper pour connaître tous les lieux.

3.3. Gestion des styles

Toujours concernant les surfaciques, la gestion des styles est à l'heure actuelle complexe pour les pays représentant plus de trois types de lieux. L'ajout d'un double contour est difficile à mettre en place. Une solution serait de créer un pattern au sein d'un fichier SVG et de l'appeler lors du style de la couche. Cela permettrait d'avoir des styles plus complexes.

La superposition avec les couches pérennes (royaume africain, aires amérindiennes) pose aussi question. Un système de hachure sur les surfaciques quand ils se trouvent superposés pourrait y pallier.

Enfin, l'affichage du nom des royaumes africains pourrait s'afficher sous forme de label ou étiquette sur les polygones. Cela pourrait se faire sous forme de tooltips permanents.

4. Recherche par type de lieu (/map/places)

4.1. Génération de routes par récit

Tout comme la page précédente, une génération de route personnalisé selon le type de lieu serait à privilégier. Cela est alors beaucoup plus simple par la suite lorsque l'on souhaite créer des liens vers ces pages, il suffit juste d'ajouter une variable à la fin du « `base_url()` ».

4.2. Ajout des géométries surfaciques

La sélection par type de lieu n'affiche actuellement que les géométries de type « point ». Il faut donc réfléchir à une méthode permettant d'afficher pour chaque type de lieu les surfaciques correspondant à des récits d'esclaves. Étant donné que le clustering est ici impossible, d'autres options peuvent être envisagées :

- On peut imaginer une variation de l'intensité de la couleur pour les pays les plus représentés dans les récits allant du plus clair (un pays qui est présent que pour 1 ou 2 récits) au plus foncé (un pays présent dans une dizaine de récit). Ici aussi, un pattern pourrait être utilisé pour créer cette variation de couleur.
- Une autre option serait d'afficher sous forme de label ou de tooltip (stylisé) le nombre de récits où le pays est présent.

Le nombre de récits présents dans chaque pays pourrait se faire dans le modèle avec un compte des `id_récits` avec une condition sur le type de lieu.

Pour ces différentes options, un pop-up au clic avec l'affichage de l'ensemble des récits sur ce surfacique est à mon avis nécessaire pour avoir des informations plus précises. Ce pourrait être le nom des narrateurs avec un lien sur leur fiche récit.

4.3. Style des clusters

Concernant les clusters qui apparaissent pour chaque type de lieu, on dispose actuellement d'un style différent par lieu avec une modification des couleurs. Mais l'idéal serait également une variation de taille en fonction du nombre de récit dans le cluster.

Pour cela, on peut utiliser la fonction `getChildcount()` qui permet de récupérer la valeur des clusters. Ensuite des classes peuvent être en fonction de valeurs récupérées. Ces classes peuvent ensuite être insérées avec les classes des clusters actuels par concaténation. On dispose ainsi de clusters qui varient selon le nombre de récits.

5. Page liste des récits (/recits)

Concernant la page liste des récits, la poursuite du développement se situera au niveau des filtres de sélection qui sont à l'heure actuelle absents du prototype.

De ce fait, l'ajout de plusieurs filtres au-dessus du tableau des récits pourrait être intéressant. Ils pourraient permettre de filtrer par type de récit, mais aussi le lieu de publication, le mode de publication...

Ces différents filtres et barres de recherche pourrait être au sein d'un même formulaire. Une fois celui-ci exécuté les informations seront récupérées grâce au modèle qui ira chercher les différents éléments correspondants par une requête SQL.

Enfin, on peut imaginer également avoir un tableau encore plus dynamique avec la possibilité d'ajouter d'autres champs (colonnes) au tableau en cochant et décochant des informations.

6. Fiche récit (/recits/id_recit)

La fiche récit reste pour le moment très sommaire. Si les informations sont modifiées pour chaque sélection de récit, les styles de la fiche sont très simplistes et pourraient être retravaillés.

Dans ce sens, il pourrait être intéressant d'avoir pour chaque récit un affichage de la première de couverture du récit sous forme d'image. Ainsi, lorsque l'utilisateur va sur la fiche une image est appelée dans la table des récits. Celle-ci peut être stockée au format BLOB dans un enregistrement. Il faut tout de même faire attention au volume de données.

De plus, un bouton supplémentaire au sein de la fiche renvoyant vers la carte du récit concerné pourrait être intéressant et utile pour l'utilisateur. Celui-ci serait réalisable si la visualisation des récits se fait avec des routes propres à chaque récit. Il suffirait juste de générer un lien avec à la fin du lien la variable de l'identifiant récit.

7. Page contact (/contact)

La page contact n'as pas du tout été développée. Elle est pourtant essentielle pour les utilisateurs souhaitant être en relation avec Mme Baduel.

Un formulaire html peut être créé avec plusieurs rubriques :

- L'email de l'utilisateur s'il n'est pas connecté.
- Le contenu de son message avec une limite de texte pour ne pas surcharger le site ou quelconque problème technique.
- Et enfin un bouton de type submit pour envoyer le message à l'adresse mail enregistrée sur le site.

8. Choix de la langue

Le site web doit également pouvoir être lu en anglais. Cela pourrait être effectué sur CodeIgniter dans les fichiers de configurations de l'application et notamment dans le répertoire 'App/language'. Certains fichiers de langues peuvent être récupérés et modifiés. La langue peut ensuite être appelée en en-tête de chaque vue.

9. Gestion des authentifications

Enfin, l'objectif final de cette application web va être de pouvoir s'authentifier et de d'avoir la possibilité d'ajouter des récits.

La gestion des authentifications est également possible dans CodeIgniter. On peut récupérer un login et un mot de passe au travers d'un formulaire. Celui-ci va envoyer les informations dans un contrôleur. Dans ce dernier, les deux informations rentrées par l'utilisateur seront comparées avec une table MySQL. Si les données correspondent, alors l'utilisateur est connecté au site et peut ajouter des informations, de manière contrôlée.

Cela nous amène à la possibilité de développer une nouvelle page avec l'ajout des récits, ici on peut avoir plusieurs formulaires et certaines règles pour que les bonnes données soient en accord avec les champs existants dans la table.

10. Commentaires généraux sur le prototype

L'application web dont il est question ici est encore à l'état de prototype et doit donc être développée et améliorée sur de très nombreux points.

D'une manière globale, le code de ce prototype pourrait être clarifié. En effet, la partie javascript pourrait être chargée séparément du code php pour ainsi permettre de mieux distinguer la partie cartographie et développement front-end de la partie plus reliée au serveur et aux données.

Concernant la page d'accueil, bien que certains filtres soient fonctionnels et appellent les bonnes données sur la carte, le contrôleur principal n'est pas optimisée et fonctionne actuellement sur un système de condition.

Pour poursuivre le développement de cette application, il est donc essentiel de reprendre et clarifier les différents éléments du framework (Modèle, Vue, Contrôleur) avec une attention particulière aux différentes tables de la base de données MySQL.