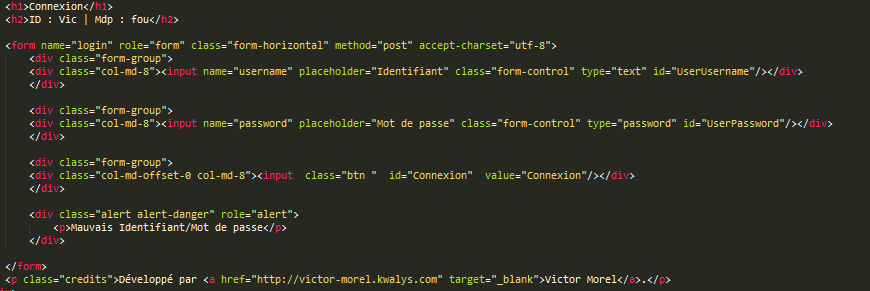
Fonctionnalités techniques

Rappel : L’application est une application mobile hybride html/css

1° Connexion



On commence par définir nos dépendances : Bootstrap et JQuery ainsi que nos polices d’écritures



On créer notre formulaire de connexion avec un champ Identifiant et Mot de passe.

On ajoute une div alerte qui s’affiche lorsque les identifiants sont erronés

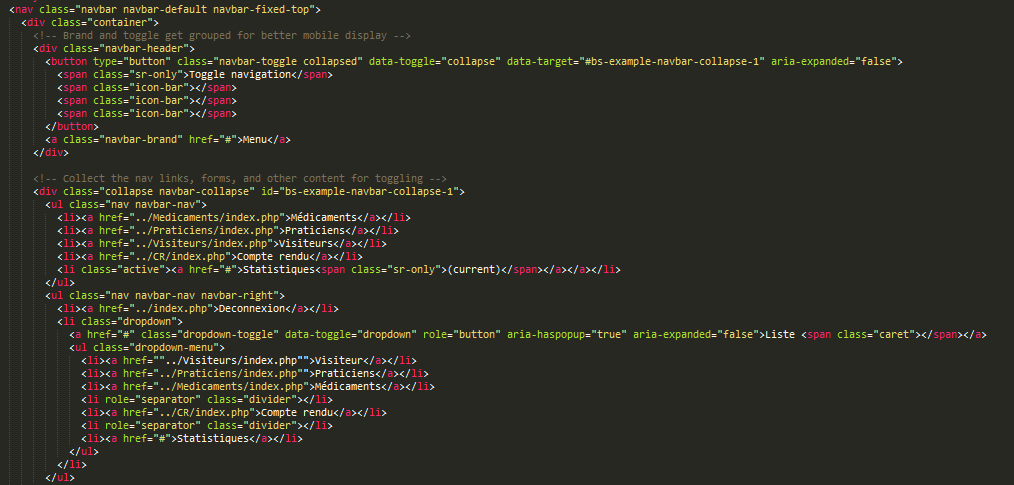


Lorsque l’utilisateur click sur le bouton envoyer on appelle notre web service web.php et lui faire passer en paramètre « connexion », le type « responsable » ainsi que l’identifiant et mot de passe rentré par l’utilisateur.

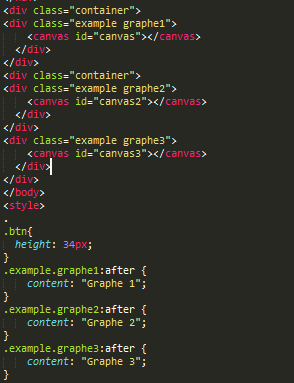
Si une réponse positive est renvoyée on effectue une redirection vers le premier écran de l’application sinon on affiche une erreur à l’utilisateur.



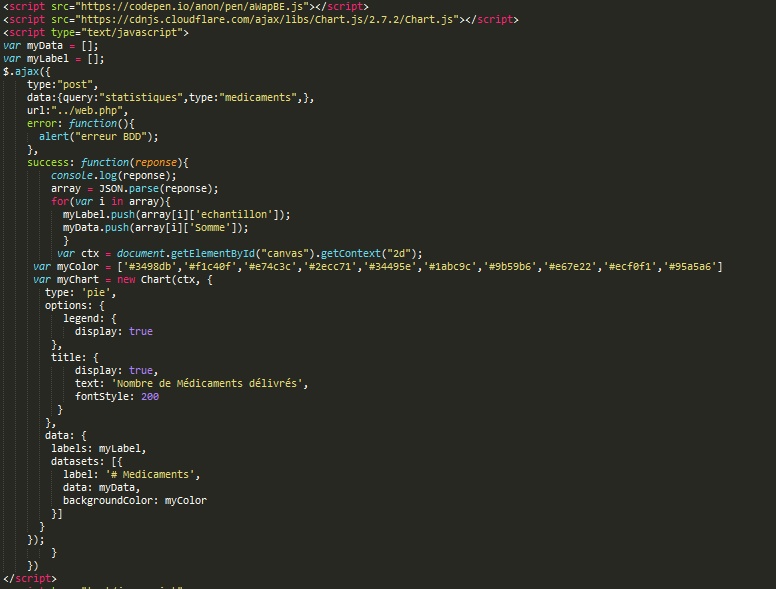
2° Statistiques



On créer notre menu de navigation

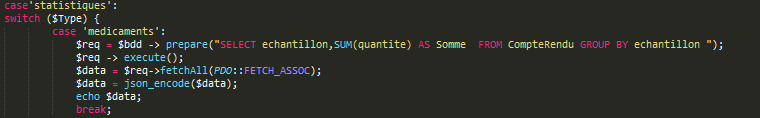


On initialise nos Canvas qui serviront à afficher nos graphes ainsi que leurs noms



On appelle le Framework js chart pour afficher nos graphiques.

On effectue ensuite un appel ajax à notre web service pour récupère le nom de chaque échantillon ainsi que la somme de chacun qu’on ajoute respectivement un tableau MyLabel et MyData



On initialise ensuite les différents couleurs que prendra notre graphique puis on créer ce dernier

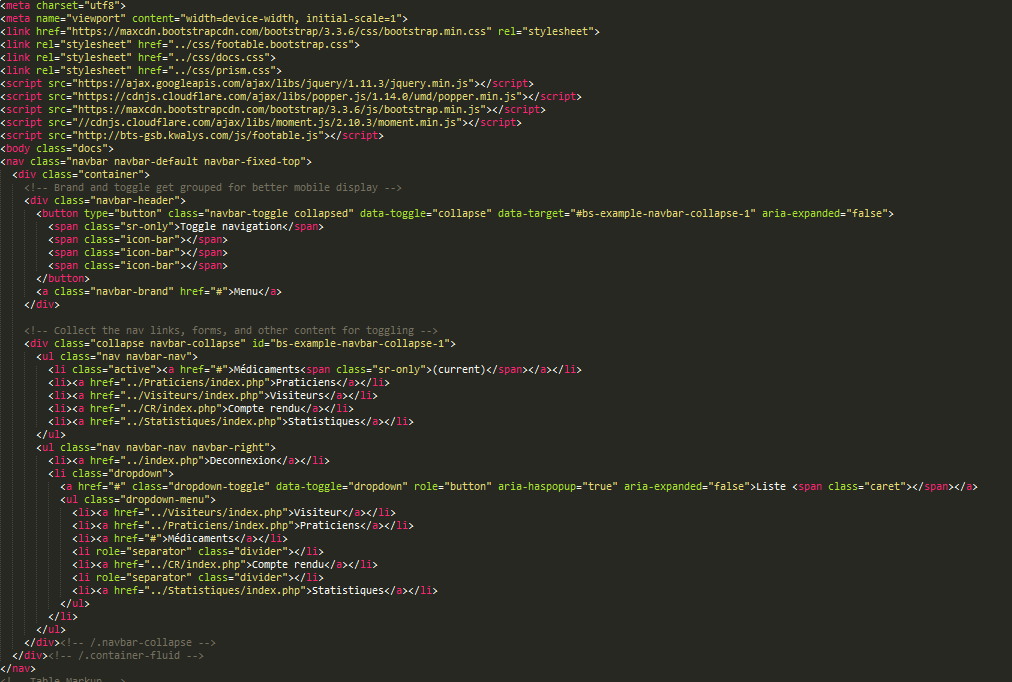
On initialise son type (ici pie pour faire un camembert) , on choisit son titre dans l’array title->text

Puis on initialise les données qui le constitue avec myLabel et myData

On réitère le même processus pour tous les autres graphiques en changeant juste les paramètres d’appel de notre web service ainsi que le type de graph à afficher

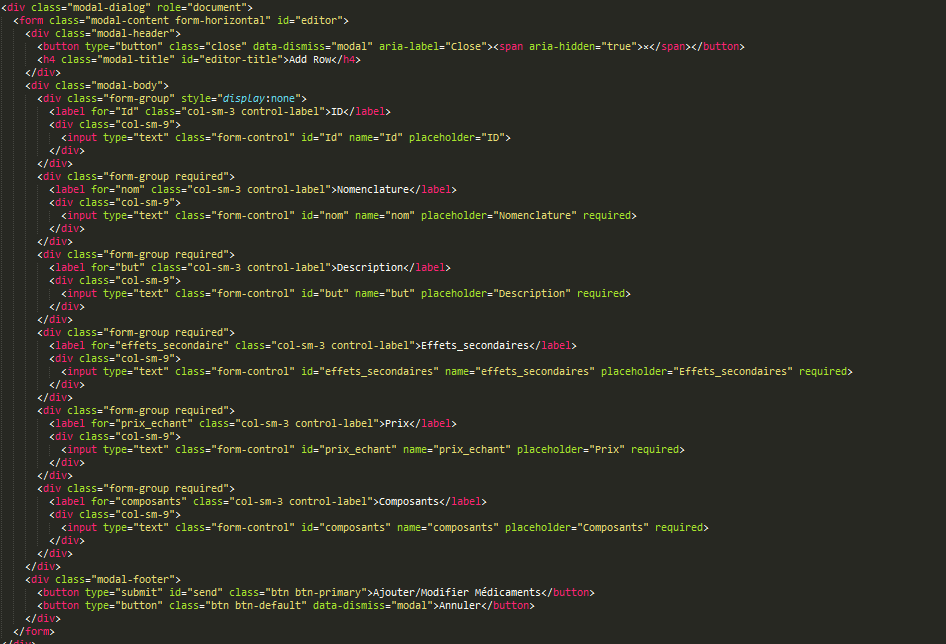
3° Médicaments

Il faut savoir que Médicaments, Visiteurs, Praticiens et compte rendu fonctionne de la même façon, je me baserais donc juste sur médicaments pour que vous compreniez le concept global.

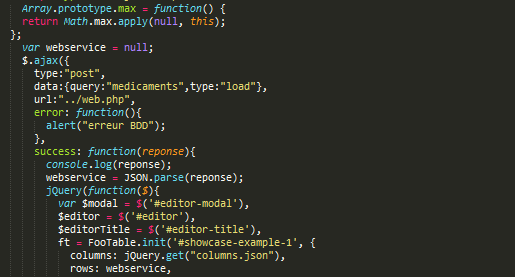


On appelle toutes nos dépendances : Bootstrap, JQuery, Ajax et Footable

Comme pour les statistiques on créer notre navbar

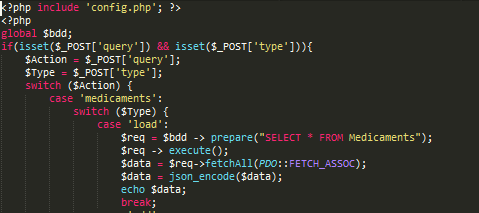


On crée ensuite notre modal qui s’affichera à chaque fois que l’on voudra ajouter/modifier un médicament



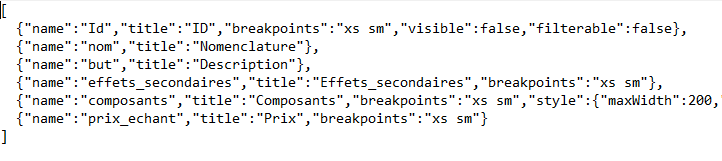
Ici on crée une fonction qui nous permet de connaître le plus grand élément d’un tableau qu’on utilisera pour connaître l’ID le plus grand de notre tableau

On appelle ensuite notre web service pour qu’il nous retourne la liste des médicaments



Si l’appelle est bien effectuer on commence à créer notre tableau html grâce à Footable :

On initialise la modal à utiliser, le tableau ainsi que son titre, on remplit ce dernier en fonction du retour de notre web service et d’un autre fichier json préconfiguré





Ici on définit l’ajout, la suppression et l’Edition d’une ligne :

AddRow :

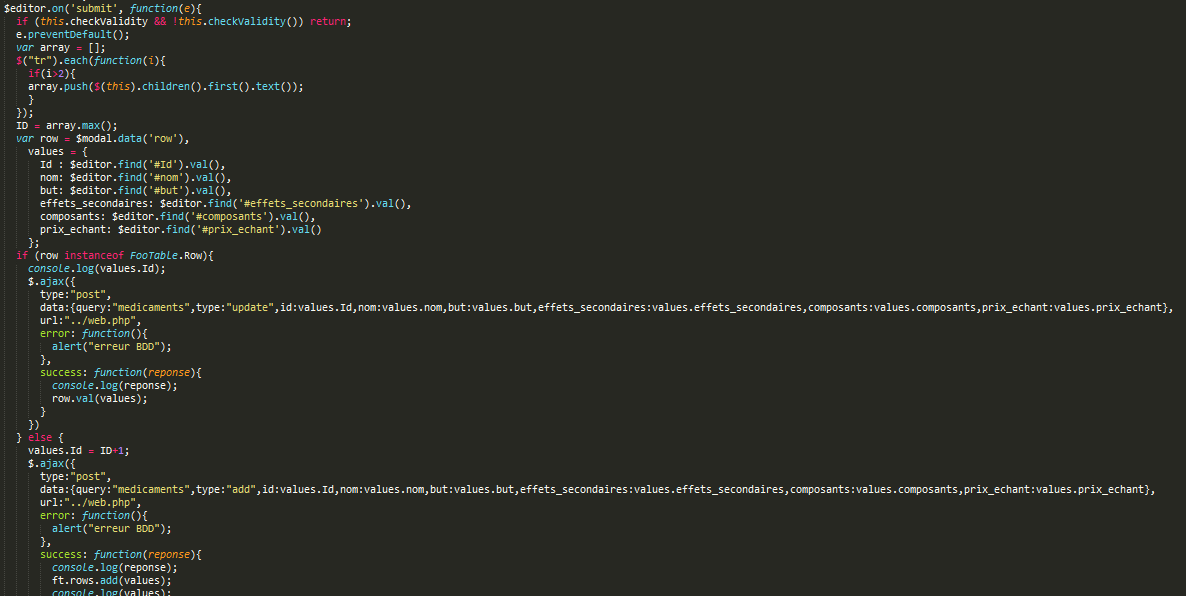
Tout d’abord on vide notre modal avec un removeData(row), au cas ou nous sommes dans le cas d’un 2ème ajout de médicaments puis on change le texte de notre modal « nouveau médicaments » pour indiquer un ajout et ensuite on affiche notre modal

EditRow :

On récupère la du tableau concerné et prérempli notre modal puis on change le nom de notre modal par le nom du médicament à modifier puis on affiche la modale

DeleRow :

On fait un appel à notre web service pour indiquer qu’on supprimer une ligne : si cette suppression est bien effectuée en BDD alors on fait un row.delete()



La partie qui suit concerne l’ajout/la modification au moment d’envoyer le formulaire contenue dans la modale : En premier lieu on récupère tout les <tr> de notre tableau pour trouver l’id le plus grand qui nous sera utile lors de l’insert (en effet même si en base de donnée notre ID est en auto incrémente et ne nécessite donc pas en théorie qu’on lui rappelle l’ID du tableau, Footable lui en aura besoin pour créer la ligne avec le bon ID vue que le web service ne renvoit pas l’id du last insert

On va donc récupérer toutes les valeurs de nos champ input de notre modal

Puis le cas row instanceof Footable.row signifie renvoit true si notre row existe déjà et donc qu’il s’agit d’une modification dans ce cas-là on appelle notre web en lui faisant passé nos paramètres à modifier. SI cette dernière est bien modifiée on fait un row.val(values) pour update notre row côté client. On fait la même chose pour l’ajout mais on utilise la fonction ft.row.add(values)

On répètre cette même logique dans tous nos fichiers sauf pour CR ou on enlève la suppression, l’ajout et la modification pour garder seulement l’affichage mais le fond reste le même