**Documentation du client Web**

# Généralités

**Construction du site :** Thomas Prak

**Contact :** thomas.prak@imerir .com

Le client web a pour objectifs de permettre aux utilisateurs de consulter les données mesurées par la station météo Arduino.

Construction du site : Thomas Prak

**Développement du client web :**



Boostrap : Framework Html Css et JavaScript pour l’architecture et le design du client

Highcharts : API pour la construction des graphes de données

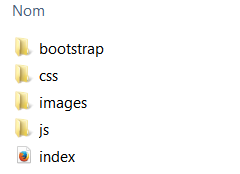
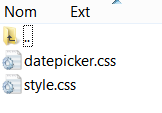
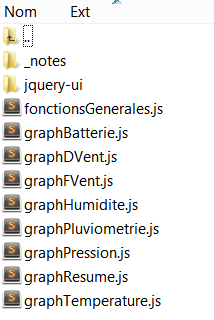
jQuery : Pour la gestion des évènements

Ajax & Json : Pour l’envoi et la réception des données des requêtes http

# Architecture

## Organisation des dossiers

Le client web se compose de plusieurs dossiers et fichiers :



**Bootstrap** : contient les fichiers du framework Bootstrap

**Css**: contient les fichiers de style pour les datepicker (datepicker.css) ainsi que pour le footer (style.css)

**Images** : dossier contenant tous les logos et autres images du client

**Js**: contient la librairie jQuery ainsi que les différents scripts JavaScript auxquels sont associés les fonctionnalités et les graphiques du client web.

**Index.html** : fichier html du client, page de navigation.

## Fichier «  index.html »

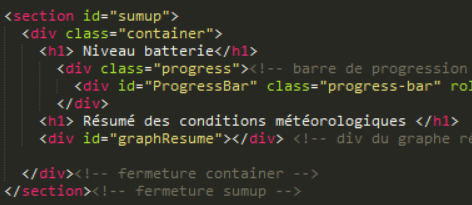
Le fichier index.html se découpe en plusieurs parties. Les parties classiques tel que le <<head>> contenant les importations des différentes librairies /API / Frameworks.

Le <<body>> se découpe en plusieurs sections :

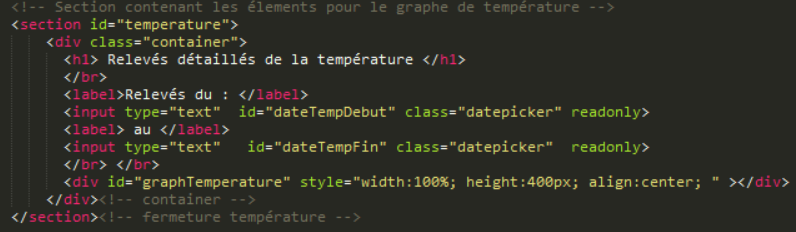
* Section info : bannière et résumé du projet



* Section sumup : barre de progression de la batterie + graphe résumé



* Section « capteur » : section contenant un graphe associé à un capteur



On retrouve également dans le body la construction de la barre de navigation.

Pour finir, le fichier contient également une partie pour la création du «footer »  du client.

## Fichier « fonctionsGenerales.js »

Ce fichier contient les différentes fonctions suivantes :

* Fonction pour le déplacement de la barre de navigation
* Initialisation du format et de la date des datepicker au chargement de la page
* Importation des scripts JS et exécution des fonctions pour le chargement des graphiques

## Fichier « graphXXX .js »

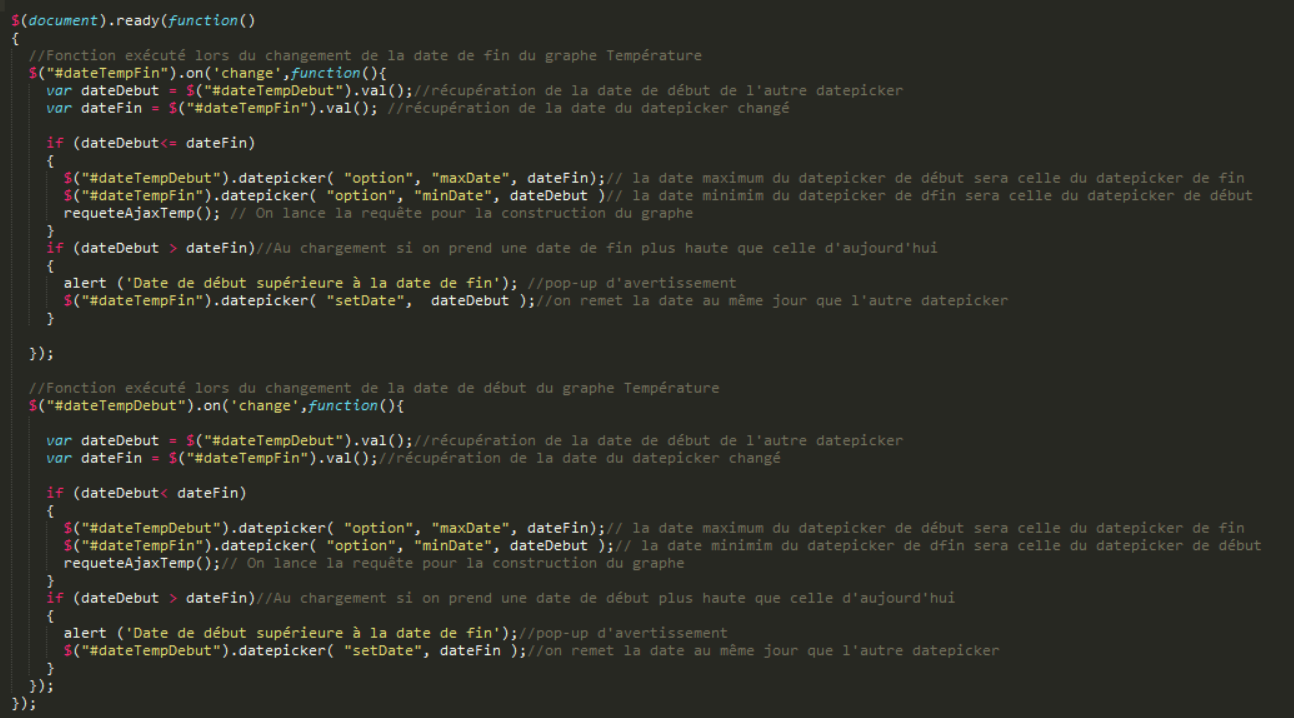
Ces différents fichiers permettent la construction de chacun des graphes du client. Le capteur auquel chaque fichier est associés est facilement identifiable avec le nom du script « graphTemperature.js ; graphPluviometrie.js…)

L’architecture des fichiers est identique mis à part celui du graphResume.js :

La première partie permet de gérer les évènements lors d’un changement de date. Lorsque la date change, une vérification est effectué pour savoir si le datepicker dit de « début » est bien inférieur au datepicker de « fin », si c’est le cas, une fonction sera exécuté afin d’actualiser le graphe.

Dans le cas contraire on met la date du datepicker concerné au même jour que l’autre datepicker.

Chaque fichier gère différents datepicker, par exemple, le fichier « graphTemp.js » sera associé aux datepicker avec l’id « dateTempDebut / dateTempFin »…

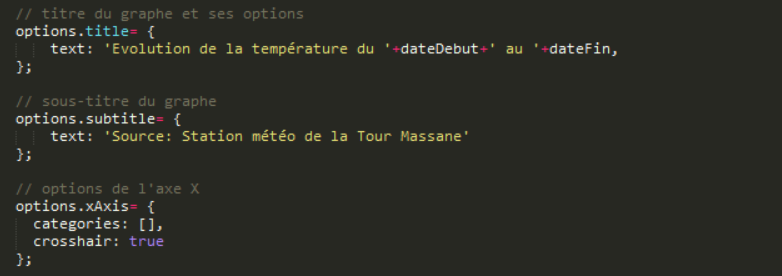


La seconde partie est une fonction qui va effectuer plusieurs actions :

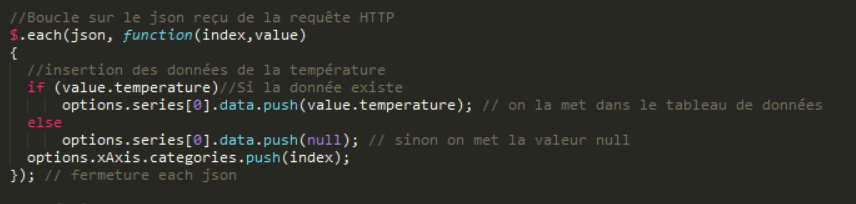
1. Lancement d’une requête ajax pour récupérer les données du capteur en question suivant les dates sélectionnées.



1. Paramétrage des options du graphique HighCharts ( capture d’écran partielle)



1. Parcage du Json reçu



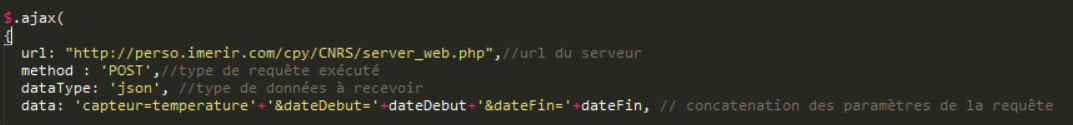
1. Initialisation / création du graphe



# Intégration

## Changement de serveur

Dans le cas d’un changement de serveur, il faut modifier l’URL de la requête Ajax dans chaque fichier Javascript intitulé « graphXXX.js » :



## Intégration d’un nouveau capteur

Pour ajouter un nouveau capteur, il faut au préalable avoir fait toutes les modification au niveau de la station météo, du serveur et de la base de données.

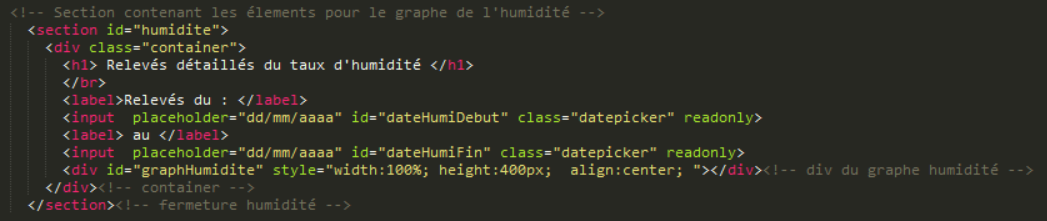
Cette partie ne traitera donc que l’intégration d’un nouveau capteur au niveau du client web.

### Intégration dans le fichier « index.html »

Pour intégrer le graphique du nouveau capteur, il suffit de copier une section existante d’un capteur et de le coller dans le « body ».

Il faut ensuite modifier plusieurs choses :

* L’id de la section
* L’id des datepickers
* L’id de la div qui contiendra le graph

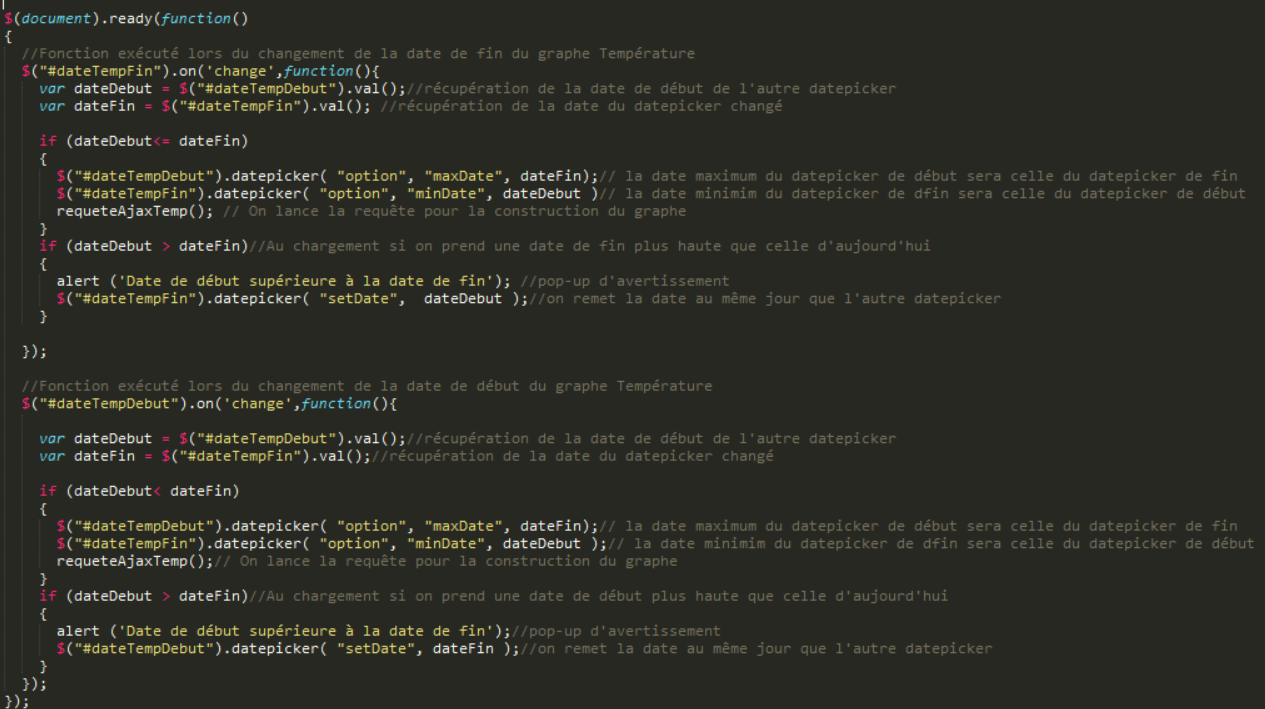


### Creation d’un nouveau fichier script

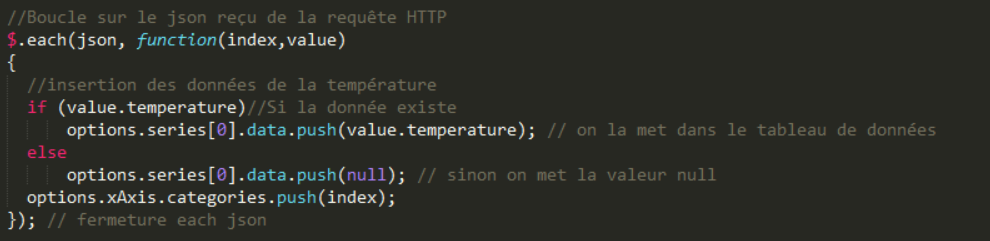
Il faut ensuite créer un nouveau fichier script qui sera dédié au graphique du nouveau capteur.

Pour cela il suffit de dupliquer un script « graphXXX.js » ( sauf graphResume.js) . Il faut ensuite modifier plusieurs éléments dans le nouveau fichier :

* Remplacer les anciens id par les nouveaux ( datepicker + div container du graphe)
* Renommer la fonction de construction du graphe
* Modifier le premier paramètre de la requête ajax (en fonction du nom attribué dans le serveur)
* Il faut également attribuer ce nom au parçage du Json
* Modifier les titres et les unités pour la construction du graphe
* Renommer la variable du chart (graphique).

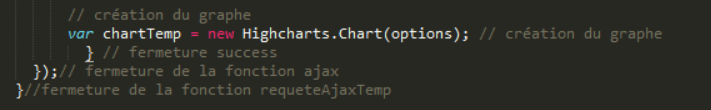




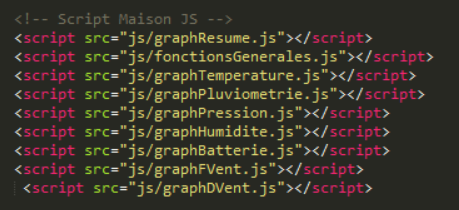






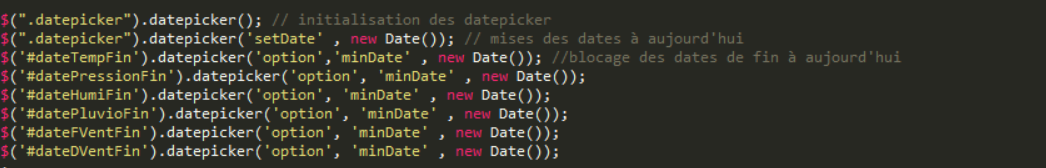


Il faudra par la suite importer ce fichier dans le « fichier index.html »:



Ainsi que dans le fichier « fonctionsGénérales.js » :

* Ajout du datepicker de fin dans cette partie :



* Ajout de l’import en copiant collant un import existant et en modifiant le chemin ainsi que le nom de la reqûete

