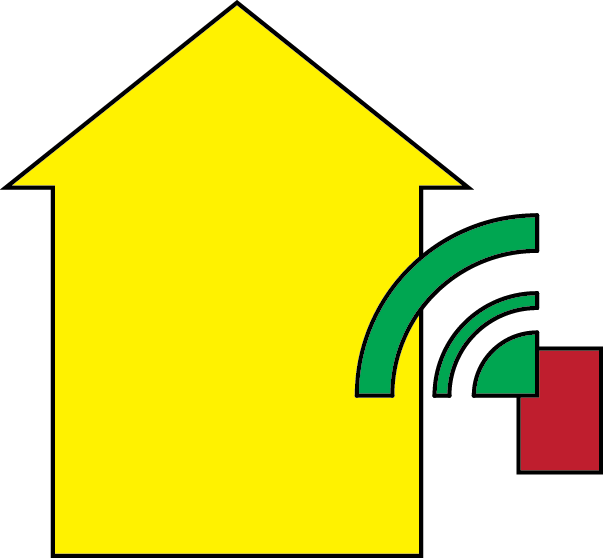
Karen LIENEMANN

David ANATON



DomoHome

Manuel développeur



# Table des matières

[1. Table des matières 2](#_Toc464917128)

[2. Historique des modifications 2](#_Toc464917129)

[3. Introduction 4](#_Toc464917130)

[4. Serveur 4](#_Toc464917131)

[4.1. Installation de Node.js 4](#_Toc464917132)

[4.2. S’assurer que l’installation a fonctionné 4](#_Toc464917133)

[4.3. Lancer le serveur 5](#_Toc464917134)

[4.4. Travail effectué 5](#_Toc464917135)

[5. Application mobile 6](#_Toc464917136)

[5.1. Installation d’Ionic 6](#_Toc464917137)

[5.2. Commencer l’émulation de l’application 7](#_Toc464917138)

[5.3. Exporter une application 8](#_Toc464917139)

[5.3.1. Sous Android 8](#_Toc464917140)

[5.3.2. Sous iOS 8](#_Toc464917141)

[5.3.3. Sous Windows Phone 8](#_Toc464917142)

[5.4. Travail effectué 8](#_Toc464917143)

[5.4.1. js/app.js 8](#_Toc464917144)

[5.4.2. js/service.js 8](#_Toc464917145)

[5.4.3. js/controllers.js 9](#_Toc464917146)

[5.4.4. templates/tab-capteurList.html 9](#_Toc464917147)

[5.5. Manuel Utilisateur 10](#_Toc464917148)

[5.5.1. Onglet réglages 10](#_Toc464917149)

[5.5.2. Onglet favoris 11](#_Toc464917150)

[5.5.3. Onglet capteurs 12](#_Toc464917151)

# Historique des modifications

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name*** | ***Date (dd/mm/yy)*** | ***Reason For Changes*** | ***Version*** |
| Karen LIENEMANN | 14/10/16 | Charte graphique | 0.1.0 |
| Karen LIENEMANN | 17/10/16 | Ajout du tutoriels Node & Ionic | 1.0.0 |
| David ANATON | 19/10/16 | Ajout de capture d’écran pour les tutoriels | 1.1.0 |
| Karen LIENEMANN | 20/10/16 | Description de l’application Node | 1.2.0 |
| David ANATON | 22/10/16 | Description de l’application Ionic | 1.3.0 |
| Karen LIENEMANN | 24/10/16 | Relecture et corrections mineures | 2.0.0 |

# Introduction

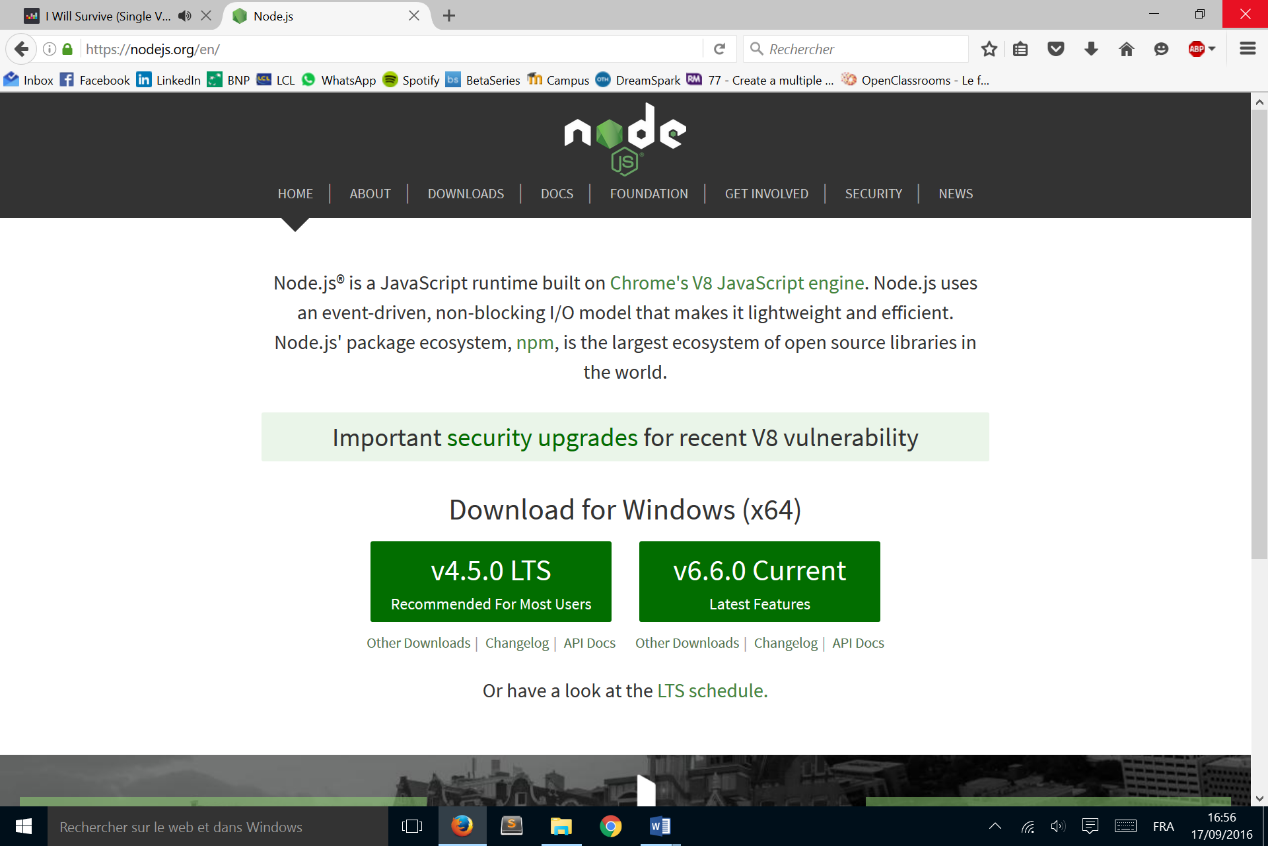
Ce manuel est là pour expliquer l’approche utilisée pour développer l’application DomoHome permettant une gestion de capteurs de températures.

# Serveur

Le serveur a été écrit en utilisant Node.js.

## Installation de Node.js

Tout d’abord, aller sur <http://nodejs.org/en/> et télécharger une des versions proposées.

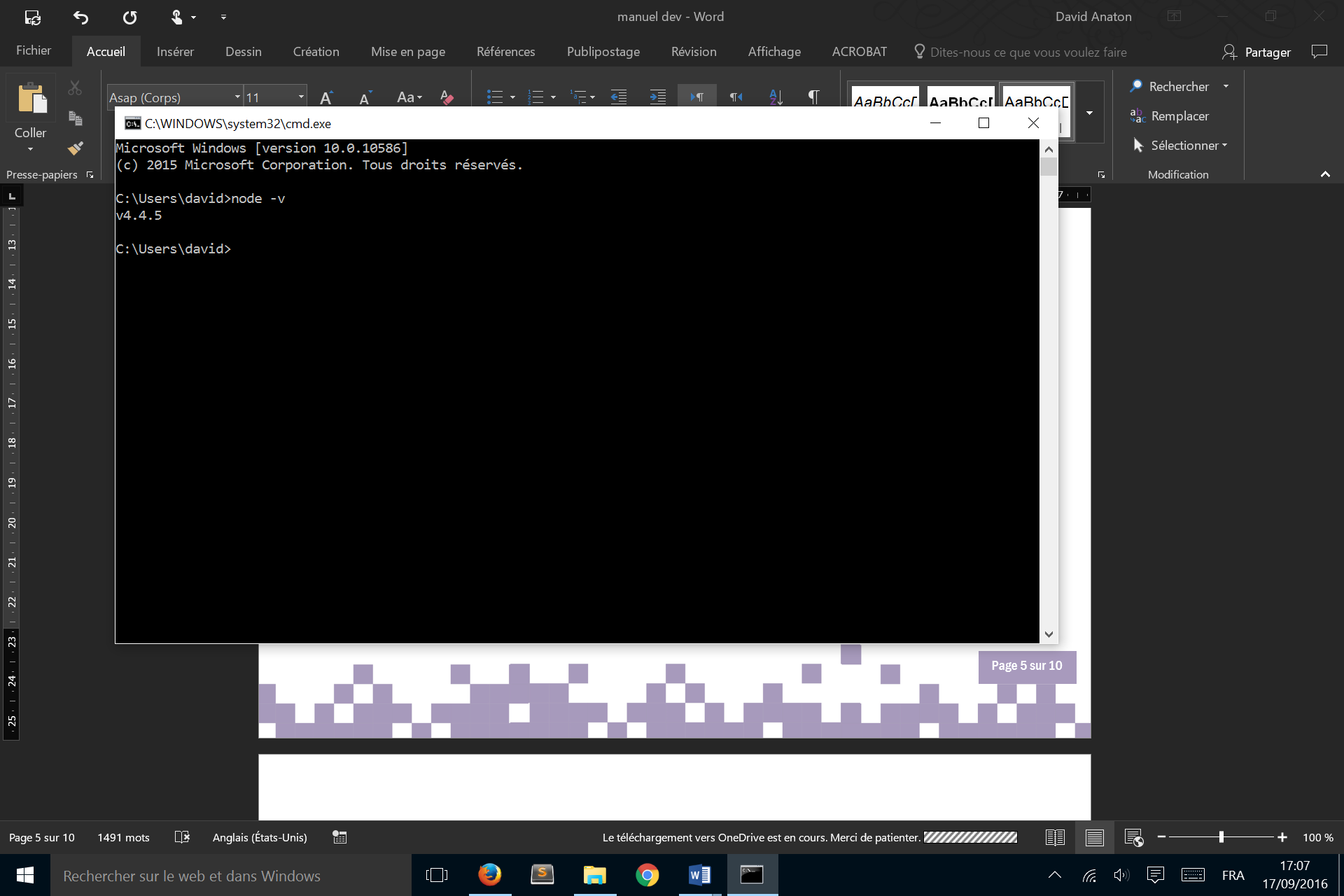


Lancer l’exécutable. Toujours cliquer sur “Next” et finir par “Finish” ; aucun logiciel tiers n’est installé par défaut.

## S’assurer que l’installation a fonctionné

Sur Windows, taper Windows+R et lancer “CMD” pour ouvrir une nouvelle invite de commande.

Vous pouvez ensuite lancer “node -v”. Vous devriez voir apparaitre le numéro de version de votre installation.

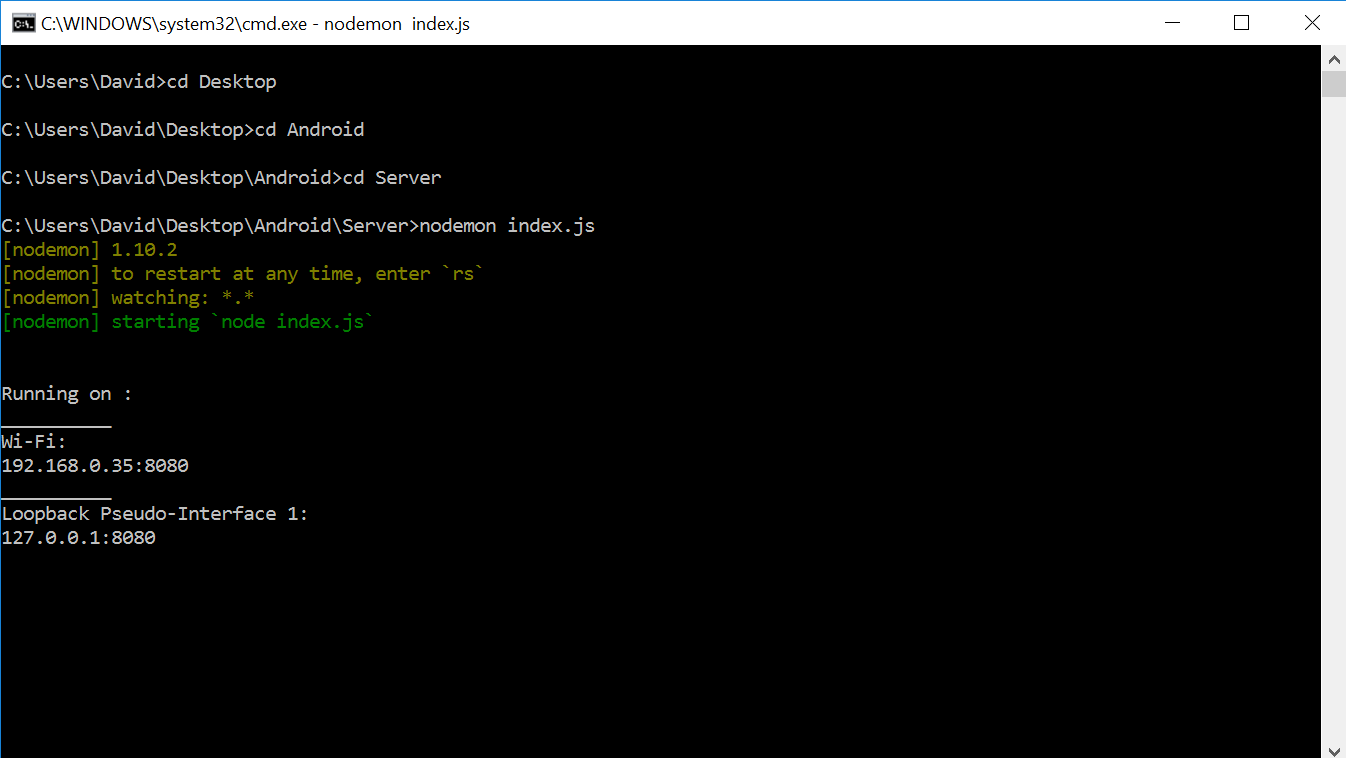


Sinon, vous devez ajouter node à votre variable d’environnement PATH (suivre [ce tutoriel](https://github.com/nodejs/node-v0.x-archive/issues/4356))

## Lancer le serveur

Ouvrir une nouvelle invite de commande, naviguer jusqu’au dossier de l’application correspondant au serveur (appelé « Server »). Commencer par installer les dépendances nécessaires pour le serveur : lancer « npm install ». Cela va installer les dépendances déclarées dans le package.json.

Lancer « node index.js ». Vous devriez voir apparaitre les IP disponibles pour se connecter au serveur.



## Travail effectué

Le serveur ne se constitue que d’un seul fichier : le index.js.

Ce dernier contient toute la logique du serveur.

Pour effectuer le travail demandé, nous avons utilisé express. Express est un framework qui permet de s’affranchir des requêtes de bas niveau pour se concentrer uniquement sur le cœur applicatif.

Ainsi, nous avons créé une application qui, en fonction des requêtes effectuées sur le serveur, de renvoyer des capteurs avec leurs températures, filtrés en fonction des paramètres de la requête.

# Application mobile

L’application mobile a été codée en Cordova avec le Framework Ionic. Ce dernier permet de développer très rapidement des applications mobiles multiplateformes en utilisant des langages web tels que HTML, CSS et Angular pour la partie JavaScript.

## Installation d’Ionic

L’installation d’Ionic requiert celle de Node. Si ce dernier n’est pas installé, merci de suivre la procédure expliquée ici : [4.1 Installation de Node.js](#_Installation_de_Node.js)

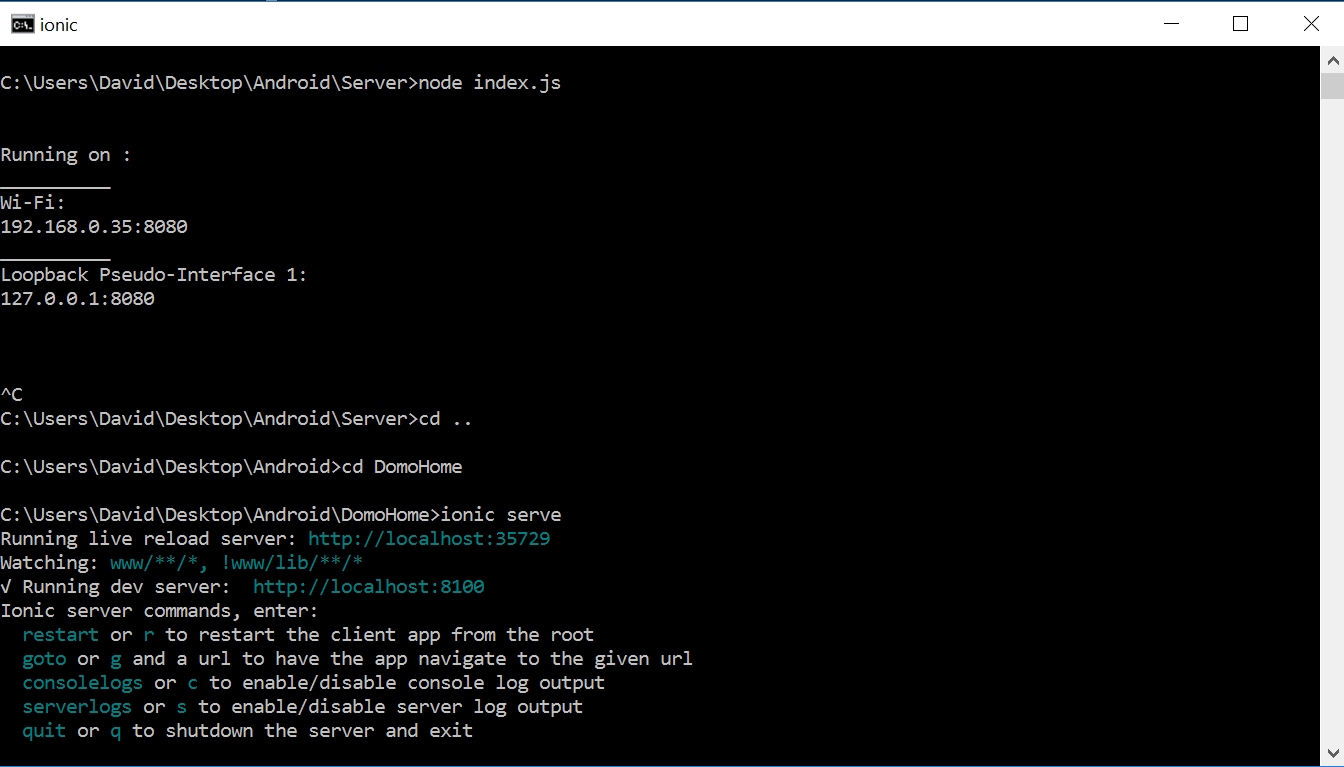
Node.js est installé avec un gestionnaire de paquet : npm.

Pour installer Ionic, nous allons l’utiliser. Ouvrir une invite de commande et taper « npm install -g cordova ionic ».

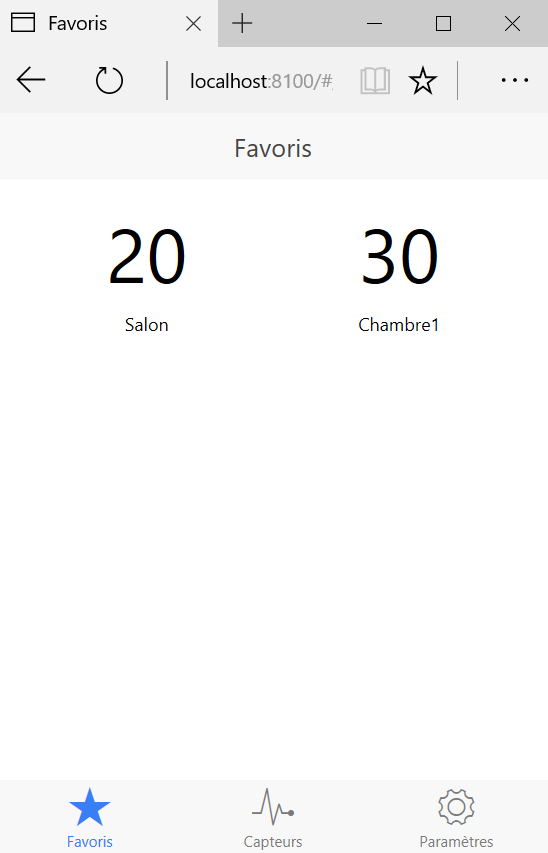


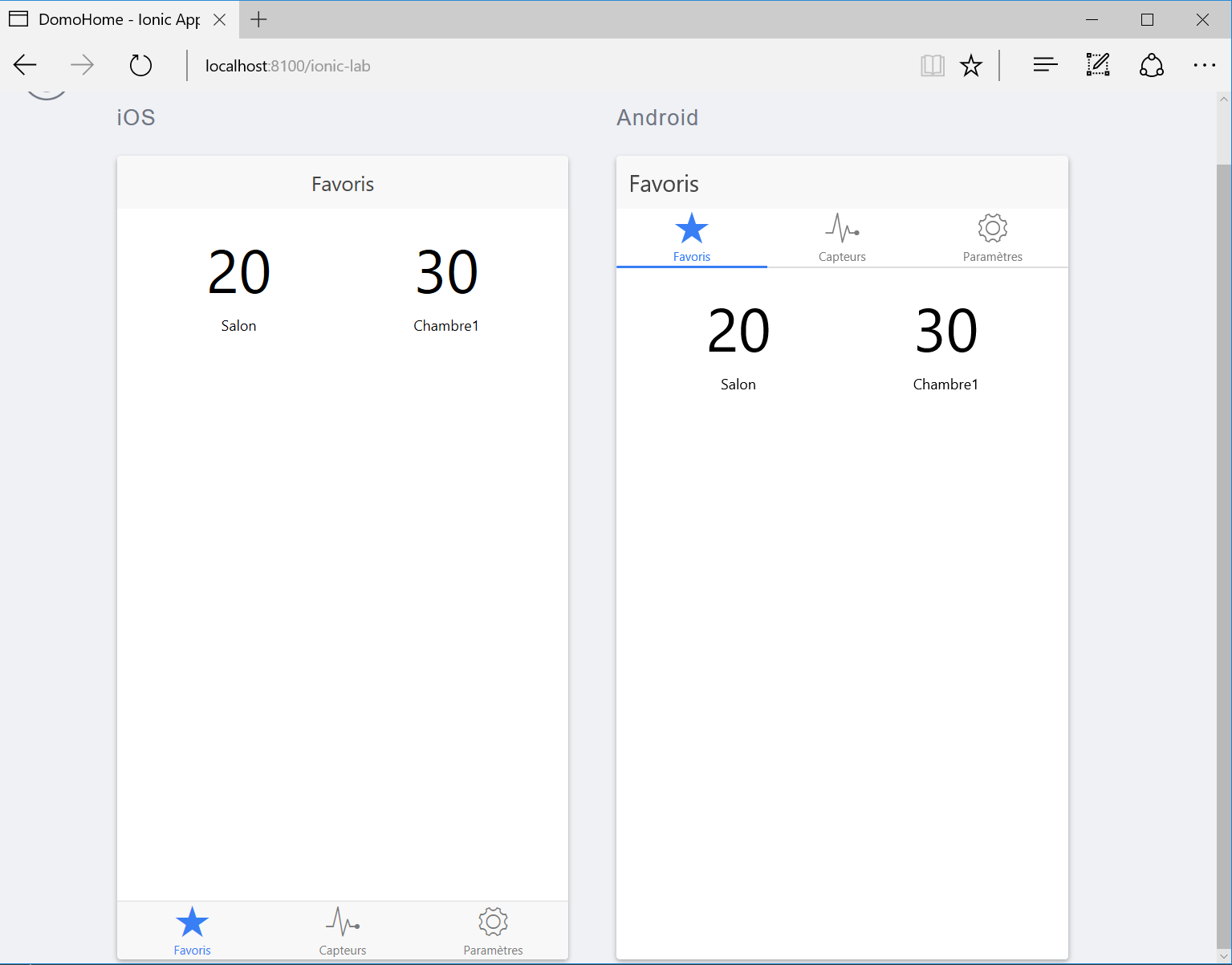
## Commencer l’émulation de l’application

Dans l’invite de commande, se déplacer jusqu’à arriver dans le dossier de l’application (nommé DomoHome), puis taper « ionic serve ».



Une fois l’invite de commande ionic en place, un navigateur contenant l’application devrait s’ouvrir.



Pour émuler iOS et Android simultanément, changer la commande en « ionic serve –lab ».

N.B. : L’émulation Android n’est pas aussi fonctionnelle que celle pour iOS. Elle sert uniquement pour avoir un visuel du rendu.

## Exporter une application

### Sous Android

Toujours en étant place dans le dossier de l’application, lancer « cordova build --release android ». Le fichier APK sera placé sous « .\platforms\android\build\outputs\apk ».

### Sous iOS et Windows Phone

Suivre le tutoriel décrit sur <https://ionicframework.com/docs/guide/publishing.html>.

## Travail effectué

Nous avons créé un projet Ionic en utilisant la commande « ionic start DomoHome tabs » qui génère une application de base avec un système d’onglet.

Le seul dossier à modifier est celui qui contient les fichiers correspondant aux éléments web, c’est-à-dire dans le dossier www.

### js/app.js

Ce fichier lie les différentes vues avec les onglets et les contrôleurs qui leur correspondent.

### js/service.js

Ce fichier créé un service « Capteurs » qui sera accessible dans tous les contrôleurs. Ce dernier créé des fonctions qui se connectent au serveur et renvoient les valeurs que ce dernier aura rendues.

Ce fichier contient toute la logique applicative.

### js/controllers.js

Chaque contrôleur n’est utilisé que dans une seule vue. Le lien avec la vue est fait dans le fichier js/app.js.

Pour expliquer le fonctionnement d’un contrôleur, prenons le premier, qui sert aux favoris.

* L’objet $scope est celui qui sera accessible dans le template HTML. On s’en sert pour stocker toutes les informations qui seront appelées de là-bas.
* L’évènement $ionicView.enter est celui qui est appelé quand on rentre sur la vue qui appelle ce contrôleur. On se sert de ce dernier pour recharger les valeurs des capteurs à chaque changement de vue.
* Important à savoir : on peut également partager une variable dans tout le programme avec le $rootScope. C’est ce qu’on fait notamment avec l’adresse IP du serveur distant.

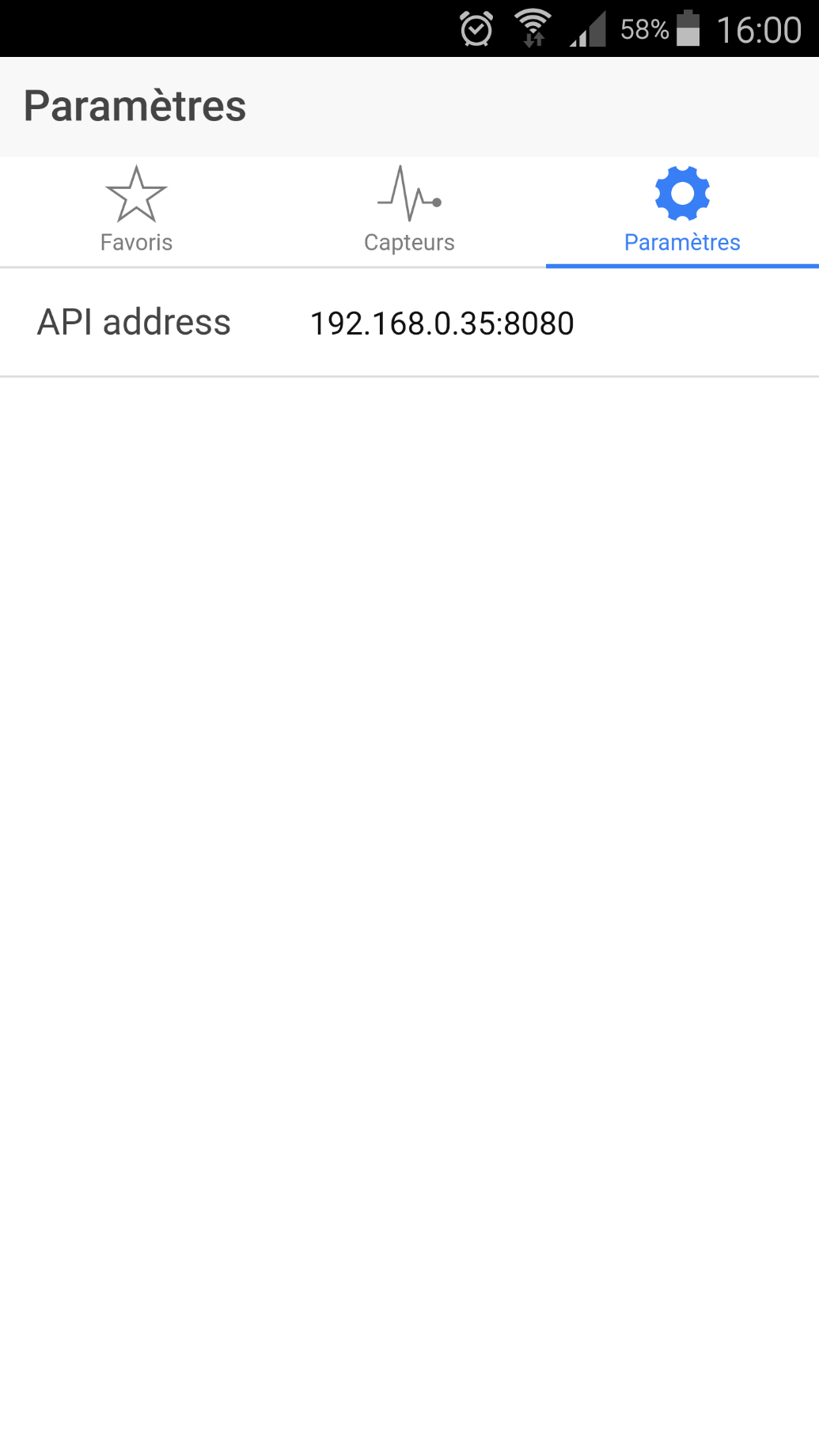
### templates/tab-capteurList.html

Ce fichier résume à peu près toutes les fonctionnalités utilisées dans l’application.

* view-title dans le ion-view permet de changer le nom qui apparait dans la barre de navigation.
* ng-click permet de déclencher une action dans le contrôleur au click de l’élément sur lequel il est placé. Le texte dans le ng-click doit être une fonction contenue dans l’objet $scope du contrôleur de la page où l’on est.
* ion-refresher permet d’automatiquement rafraichir la page lorsque l’on fait défiler la page vers le bas.
* ng-repeat="capteur in capteurs" :
* On boucle sur la variable $scope.capteurs.
* Pour chaque élément rencontré, on crée une variable capteur qui ne sera accessible que dans l’espace des balises entourant le ng-repeat.
* Les doubles accolades servent à afficher une valeur de variables javascript contenue dans le contrôleur.
* ion-option-button permet de placer un bouton contextuel comme celui permettant de supprimer un capteur.

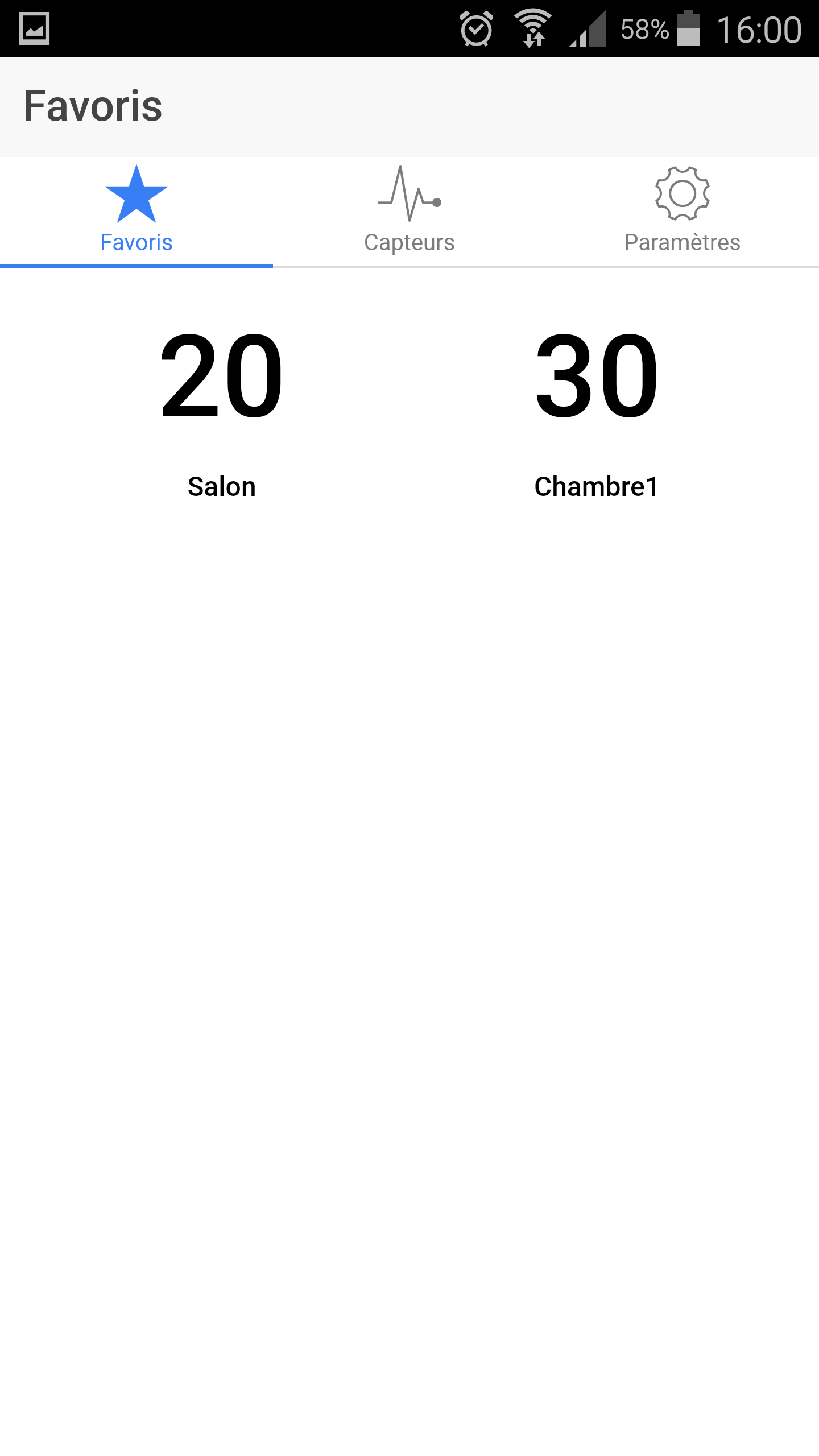
## Manuel Utilisateur

### Onglet réglages

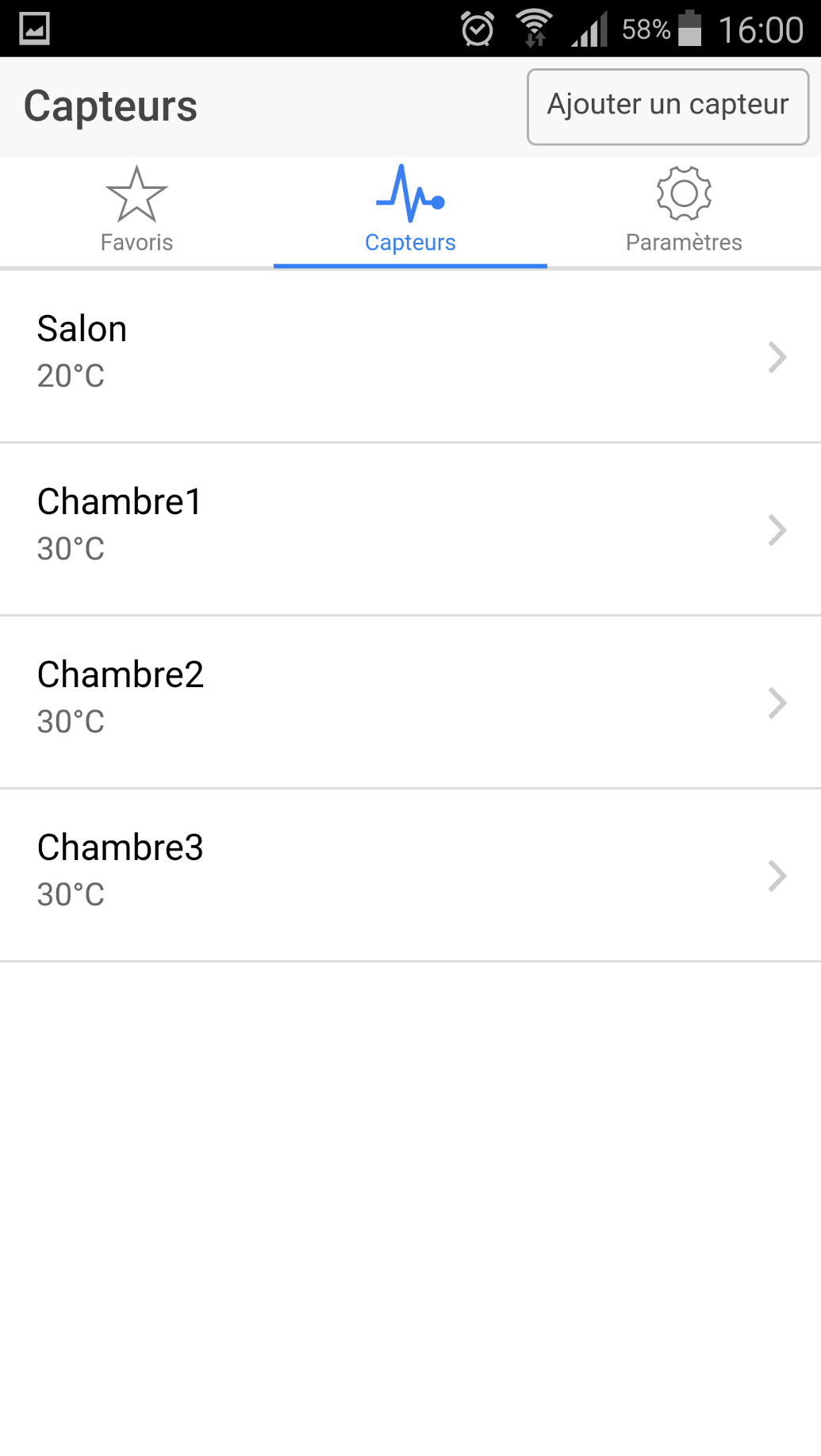
La première chose à faire en allumant l’application est de configurer l’adresse du serveur distant. Cette dernière peut être lue au lancement de ce dernier dans la ligne de commande.

### Onglet favoris

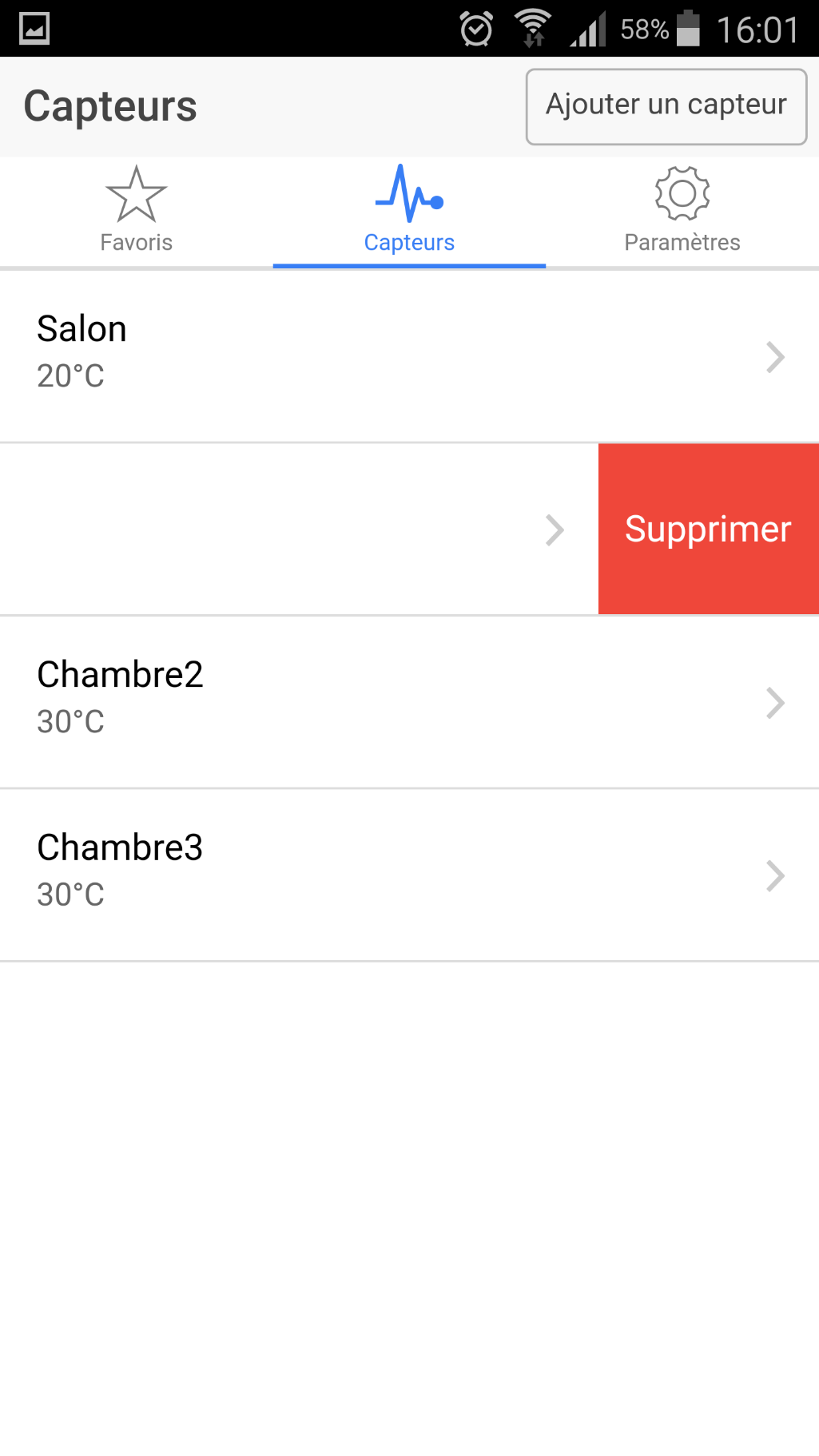
C’est l’onglet sur lequel l’application s’ouvre. Il n’affiche que les températures des capteurs favoris. Pour rafraichir les températures, tirer l’écran vers le bas.



### Onglet capteurs

Cet onglet regroupe tous les capteurs de l’utilisateurs. Pour rafraichir, vous pouvez encore glisser vers le bas. Vous pouvez également cliquer sur le bouton « Ajouter un capteur » pour créer un nouveau capteur.

Vous pouvez également faire glisser un élément vers la gauche pour le supprimer.



Enfin, en cliquant sur un élément, on accède à ses caractéristiques. On peut les modifier au même endroit.

