

Smart VMC

1.0

Generated on Sun Jan 18 2026 09:43:30 for Smart VMC by Doxygen 1.15.0

Sun Jan 18 2026 09:43:30

Chapter 1

Directory Hierarchy

1.1 Directories

components	??
dashboard	??
ControlPanel.tsx	??
SensorRow.tsx	??
StatusDisplay.tsx	??
ui	??
AppButton.tsx	??
AppInput.tsx	??
Card.tsx	??
ConfigurationPanel.tsx	??
DashboardPanel.tsx	??
SetupPanel.tsx	??
TabHeader.tsx	??
dashboard	??
ControlPanel.tsx	??
SensorRow.tsx	??
StatusDisplay.tsx	??
firmware	??
src	??
main.hpp	??
hooks	??
useBLE.ts	??
useWindowApi.ts	??
mobile	??
components	??
dashboard	??
ControlPanel.tsx	??
SensorRow.tsx	??
StatusDisplay.tsx	??
ui	??
AppButton.tsx	??
AppInput.tsx	??
Card.tsx	??
ConfigurationPanel.tsx	??
DashboardPanel.tsx	??
SetupPanel.tsx	??

TabHeader.tsx	??
hooks	??
useBLE.ts	??
useWindowApi.ts	??
services	??
NotificationService.ts	??
App.tsx	??
services	??
NotificationService.ts	??
src	??
main.hpp	??
ui	??
AppButton.tsx	??
AppInput.tsx	??
Card.tsx	??

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

AppButtonProps	??
BLECharacteristicCallbacks	
ConfigCallbacks	??
CardProps	??
ConfigurationPanelProps	??
ControlPanelProps	??
DashboardPanelProps	??
NotificationService	??
SensorRowProps	??
SetupPanelProps	??
StatusDisplayProps	??
TabHeaderProps	??
TextInputProps	
AppInputProps	??
WindowState	??

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AppButtonProps	
Propriétés du bouton	??
AppInputProps	
Propriétés du champ de saisie	??
CardProps	
Propriétés du composant carte	??
ConfigCallbacks	
Traite les configurations WiFi et de localisation	??
ConfigurationPanelProps	
Propriétés du panneau de configuration	??
ControlPanelProps	
Propriétés du panneau de contrôle	??
DashboardPanelProps	
Propriétés du tableau de bord	??
NotificationService	
Service pour gérer les notifications locales	??
SensorRowProps	
Propriétés de la ligne de capteurs	??
SetupPanelProps	
Propriétés du panneau de configuration initiale	??
StatusDisplayProps	
Propriétés de l'affichage d'état	??
TabHeaderProps	
Propriétés de l'en-tête d'onglets	??
WindowState	
État de la fenêtre côté mobile	??

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

firmware/src/ main.hpp	
Déclarations principales pour le firmware de contrôle de fenêtre VMC	??
mobile/ App.tsx	
Composant principal de l'application mobile Smart VMC	??
mobile/components/ ConfigurationPanel.tsx	
Panneau de configuration des seuils de température et pollution	??
mobile/components/ DashboardPanel.tsx	
Tableau de bord affichant l'état, les capteurs et les commandes de la VMC	??
mobile/components/ SetupPanel.tsx	
Panneau de configuration initiale avec saisie WiFi et localisation	??
mobile/components/ TabHeader.tsx	
Barre d'onglets principale	??
mobile/components/dashboard/ ControlPanel.tsx	
Panneau de contrôle de la VMC (mode auto, commandes manuelles, slider)	??
mobile/components/dashboard/ SensorRow.tsx	
Ligne d'affichage des valeurs de capteurs (température et AQI)	??
mobile/components/dashboard/ StatusDisplay.tsx	
Affichage de l'état global de la VMC (ouverte/fermée/verrouillée)	??
mobile/components/ui/ AppButton.tsx	
Bouton applicatif stylé (primaire, danger, outline) avec état de chargement	??
mobile/components/ui/ AppInput.tsx	
Champ de saisie stylé avec placeholder grisé	??
mobile/components/ui/ Card.tsx	
Conteneur carte avec ombre et arrondis	??
mobile/hooks/ useBLE.ts	
Hook React pour la communication Bluetooth avec l'ESP32	??
mobile/hooks/ useWindowApi.ts	
Hook React pour communiquer avec l'API smart-vmc	??
mobile/services/ NotificationService.ts	??

Chapter 5

Directory Documentation

5.1 mobile/components Directory Reference

Directories

- directory [dashboard](#)
- directory [ui](#)

Files

- file [ConfigurationPanel.tsx](#)
Panneau de configuration des seuils de température et pollution.
- file [DashboardPanel.tsx](#)
Tableau de bord affichant l'état, les capteurs et les commandes de la VMC.
- file [SetupPanel.tsx](#)
Panneau de configuration initiale avec saisie WiFi et localisation.
- file [TabHeader.tsx](#)
Barre d'onglets principale.

5.2 mobile/components/dashboard Directory Reference

Files

- file [ControlPanel.tsx](#)
Panneau de contrôle de la VMC (mode auto, commandes manuelles, slider).
- file [SensorRow.tsx](#)
Ligne d'affichage des valeurs de capteurs (température et AQI).
- file [StatusDisplay.tsx](#)
Affichage de l'état global de la VMC (ouverte/fermée/verrouillée).

5.3 firmware Directory Reference

Directories

- directory [src](#)

5.4 mobile/hooks Directory Reference

Files

- file [useBLE.ts](#)
Hook React pour la communication Bluetooth avec l'ESP32.
- file [useWindowApi.ts](#)
Hook React pour communiquer avec l'API smart-vmc.

5.5 mobile Directory Reference

Directories

- directory [components](#)
- directory [hooks](#)
- directory [services](#)

Files

- file [App.tsx](#)
Composant principal de l'application mobile Smart VMC.

5.6 mobile/services Directory Reference

Files

- file [NotificationService.ts](#)

5.7 firmware/src Directory Reference

Files

- file [main.hpp](#)
Déclarations principales pour le firmware de contrôle de fenêtre VMC.

5.8 mobile/components/ui Directory Reference

Files

- file [AppButton.tsx](#)
Bouton applicatif stylé (primaire, danger, outline) avec état de chargement.
- file [AppInput.tsx](#)
Champ de saisie stylé avec placeholder grisé
- file [Card.tsx](#)
Conteneur carte avec ombre et arrondis.

Chapter 6

Class Documentation

6.1 AppButtonProps Interface Reference

Propriétés du bouton.

6.1.1 Detailed Description

Propriétés du bouton.

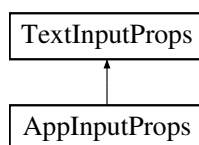
The documentation for this interface was generated from the following file:

- mobile/components/ui/[AppButton.tsx](#)

6.2 AppInputProps Interface Reference

Propriétés du champ de saisie.

Inheritance diagram for AppInputProps:



6.2.1 Detailed Description

Propriétés du champ de saisie.

The documentation for this interface was generated from the following file:

- mobile/components/ui/[AppInput.tsx](#)

6.3 CardProps Interface Reference

Propriétés du composant carte.

6.3.1 Detailed Description

Propriétés du composant carte.

The documentation for this interface was generated from the following file:

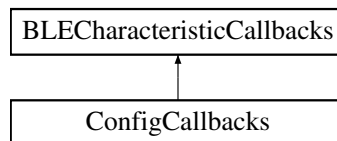
- [mobile/components/ui/Card.tsx](#)

6.4 ConfigCallbacks Class Reference

Traite les configurations WiFi et de localisation.

```
#include <main.hpp>
```

Inheritance diagram for ConfigCallbacks:



6.4.1 Detailed Description

Traite les configurations WiFi et de localisation.

The documentation for this class was generated from the following file:

- [firmware/src/main.hpp](#)

6.5 ConfigurationPanelProps Interface Reference

Propriétés du panneau de configuration.

6.5.1 Detailed Description

Propriétés du panneau de configuration.

The documentation for this interface was generated from the following file:

- [mobile/components/ConfigurationPanel.tsx](#)

6.6 ControlPanelProps Interface Reference

Propriétés du panneau de contrôle.

6.6.1 Detailed Description

Propriétés du panneau de contrôle.

The documentation for this interface was generated from the following file:

- [mobile/components/dashboard/ControlPanel.tsx](#)

6.7 DashboardPanelProps Interface Reference

Propriétés du tableau de bord.

6.7.1 Detailed Description

Propriétés du tableau de bord.

The documentation for this interface was generated from the following file:

- [mobile/components/DashboardPanel.tsx](#)

6.8 NotificationService Class Reference

Service pour gérer les notifications locales.

Public Member Functions

- `async requestPermission ()`
@function requestPermission
- `async createChannel ()`
@function createChannel
- `async displayNotification (title:string, body:string)`
@function displayNotification

6.8.1 Detailed Description

Service pour gérer les notifications locales.

Ce service utilise la bibliothèque Notifee pour :

- Demander la permission d'envoyer des notifications
- Créer un canal de notification pour les alertes VMC
- Afficher des notifications locales avec un titre et un corps

6.8.2 Member Function Documentation

6.8.2.1 createChannel()

```
async NotificationService::createChannel () [inline]
```

```
@function createChannel
```

Crée un canal de notification pour les alertes VMC

Returns

{Promise<string>} L'ID du canal créé

6.8.2.2 displayNotification()

```
async NotificationService::displayNotification (  
    title:string,  
    body:string) [inline]
```

```
@function displayNotification
```

Affiche une notification locale avec le titre et le corps spécifiés

Parameters

{string}	title - Le titre de la notification
{string}	body - Le corps de la notification

6.8.2.3 requestPermission()

```
async NotificationService::requestPermission () [inline]
```

```
@function requestPermission
```

Demande la permission d'envoyer des notifications

Returns

{Promise<boolean>} Vrai si la permission est accordée, sinon faux

The documentation for this class was generated from the following file:

- mobile/services/[NotificationService.ts](#)

6.9 SensorRowProps Interface Reference

Propriétés de la ligne de capteurs.

6.9.1 Detailed Description

Propriétés de la ligne de capteurs.

The documentation for this interface was generated from the following file:

- [mobile/components/dashboard/SensorRow.tsx](#)

6.10 SetupPanelProps Interface Reference

Propriétés du panneau de configuration initiale.

6.10.1 Detailed Description

Propriétés du panneau de configuration initiale.

The documentation for this interface was generated from the following file:

- [mobile/components/SetupPanel.tsx](#)

6.11 StatusDisplayProps Interface Reference

Propriétés de l'affichage d'état.

6.11.1 Detailed Description

Propriétés de l'affichage d'état.

The documentation for this interface was generated from the following file:

- [mobile/components/dashboard/StatusDisplay.tsx](#)

6.12 TabHeaderProps Interface Reference

Propriétés de l'en-tête d'onglets.

6.12.1 Detailed Description

Propriétés de l'en-tête d'onglets.

The documentation for this interface was generated from the following file:

- [mobile/components/TabHeader.tsx](#)

6.13 WindowState Interface Reference

État de la fenêtre côté mobile.

6.13.1 Detailed Description

État de la fenêtre côté mobile.

The documentation for this interface was generated from the following file:

- [mobile/hooks/useWindowApi.ts](#)

Chapter 7

File Documentation

7.1 firmware/src/main.hpp File Reference

Déclarations principales pour le firmware de contrôle de fenêtre VMC.

```
#include <Arduino.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <WebServer.h>
#include <ArduinoJson.h>
#include <ESP32Servo.h>
#include <NimBLEDevice.h>
#include <Preferences.h>
```

Classes

- class [ConfigCallbacks](#)
Traite les configurations WiFi et de localisation.

Macros

- #define [SERVO_PIN](#) 13
Variable pin du servomoteur.
- #define [CURRENT_SENSOR_PIN](#) 34
Variable pin du capteur de courant.
- #define [MAX_CURRENT_THRESHOLD](#) 2500
Seuil de courant maximal en mA.
- #define [SERVICE_UUID](#) "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
UUIDs pour le service et les caractéristiques BLE.
- #define [CHAR_CONFIG_UUID](#) "beb5483e-36e1-4688-b7f5-ea07361b26a8"
UUID de la caractéristique de configuration (WiFi, localisation).
- #define [CHAR_IP_UUID](#) "12345678-1234-1234-1234-1234567890ab"
UUID de la caractéristique d'adresse IP.

Functions

- bool [checkSafety](#) ()
Vérifie l'état de sécurité du système.
- void [setVMC](#) (int angle)
Commande le servomoteur à un angle spécifique.
- void [handleStatus](#) ()
Gestionnaire HTTP GET /api/window/status.
- void [handleControl](#) ()
Gestionnaire HTTP POST /api/window/control.
- void [setup](#) ()
Initialisation de la connexion (WiFi, BLE, serveur).
- void [loop](#) ()
Boucle d'exécution principale.

Variables

- Servo [windowServo](#)
Objet servomoteur pour contrôler l'ouverture de la fenêtre.
- Preferences [preferences](#)
Objet Preferences pour stocker la configuration persistante.
- WebServer [server](#)
Objet WebServer pour gérer les requêtes HTTP.
- BLECharacteristic * [plpChar](#)
Caractéristique BLE pour l'adresse IP.
- String [wifi_ssid](#)
SSID WiFi.
- String [wifi_pass](#)
Mot de passe WiFi.
- float [latitude](#)
Variable de configuration de localisation.
- float [longitude](#)
Variable de configuration de localisation.
- float [tempMax](#)
Variables de configuration de la température maximum.
- float [tempMin](#)
Variables de configuration de la température minimum.
- int [aqiMax](#)
Variables de configuration de la qualité de l'air maximum.
- bool [isOpen](#)
Indique si la ventilation est ouverte.
- bool [autoMode](#)
Indique si le mode automatique est activé
- int [targetAngle](#)
Angle cible pour le servomoteur.
- bool [safetyLockout](#)
Indique si le verrouillage de sécurité est activé
- int [lastCurrentReading](#)
Dernière lecture du courant en mA.
- int [currentAngle](#)

- Angle actuel du servomoteur.*
 - float `lastTemp`
Dernière température lue.
- int `lastAQI`
Dernière qualité de l'air lue.
- unsigned long `lastWeatherCheck`
Timestamp de la dernière vérification météo.
- bool `shouldRestart`
Indique si un redémarrage est nécessaire.

7.1.1 Detailed Description

Déclarations principales pour le firmware de contrôle de fenêtre VMC.

7.1.2 Macro Definition Documentation

7.1.2.1 CHAR_CONFIG_UUID

```
#define CHAR_CONFIG_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"
```

UUID de la caractéristique de configuration (WiFi, localisation).

7.1.2.2 CHAR_IP_UUID

```
#define CHAR_IP_UUID "12345678-1234-1234-1234-1234567890ab"
```

UUID de la caractéristique d'adresse IP.

7.1.2.3 CURRENT_SENSOR_PIN

```
#define CURRENT_SENSOR_PIN 34
```

Variable pin du capteur de courant.

7.1.2.4 MAX_CURRENT_THRESHOLD

```
#define MAX_CURRENT_THRESHOLD 2500
```

Seuil de courant maximal en mA.

7.1.2.5 SERVICE_UUID

```
#define SERVICE_UUID "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
```

UUIDs pour le service et les caractéristiques BLE.

7.1.2.6 SERVO_PIN

```
#define SERVO_PIN 13
```

Variable pin du servomoteur.

7.1.3 Function Documentation

7.1.3.1 checkSafety()

```
bool checkSafety ()
```

Vérifie l'état de sécurité du système.

Lit le capteur de courant et vérifie s'il dépasse le seuil

Returns

true si sécurisé, false si surintensité détectée

7.1.3.2 handleControl()

```
void handleControl ()
```

Gestionnaire HTTP POST /api/window/control.

Reçoit les commandes pour contrôler la fenêtre

7.1.3.3 handleStatus()

```
void handleStatus ()
```

Gestionnaire HTTP GET /api/window/status.

Retourne l'état complet du système en JSON

7.1.3.4 loop()

```
void loop ()
```

Boucle d'exécution principale.

Gère les requêtes HTTP, la météo et les redémarrages

7.1.3.5 setup()

```
void setup ()
```

Initialisation de la connexion (WiFi, BLE, serveur).

7.1.3.6 setVMC()

```
void setVMC (  
    int angle)
```

Commande le servomoteur à un angle spécifique.

Parameters

<i>angle</i>	Angle cible en degrés (0-90)
--------------	------------------------------

Vérifie la sécurité, contraint l'angle et met à jour l'état

7.1.4 Variable Documentation

7.1.4.1 aqiMax

```
int aqiMax [extern]
```

Variables de configuration de la qualité de l'air maximum.

7.1.4.2 autoMode

```
bool autoMode [extern]
```

Indique si le mode automatique est activé

7.1.4.3 currentAngle

```
int currentAngle [extern]
```

Angle actuel du servomoteur.

7.1.4.4 isOpen

```
bool isOpen [extern]
```

Indique si la ventilation est ouverte.

7.1.4.5 lastAQI

```
int lastAQI [extern]
```

Dernière qualité de l'air lue.

7.1.4.6 lastCurrentReading

```
int lastCurrentReading [extern]
```

Dernière lecture du courant en mA.

7.1.4.7 lastTemp

```
float lastTemp [extern]
```

Dernière température lue.

7.1.4.8 lastWeatherCheck

```
unsigned long lastWeatherCheck [extern]
```

Timestamp de la dernière vérification météo.

7.1.4.9 latitude

```
float latitude [extern]
```

Variable de configuration de localisation.

7.1.4.10 longitude

```
float longitude [extern]
```

Variable de configuration de localisation.

7.1.4.11 plpChar

```
BLECharacteristic* pIpChar [extern]
```

Caractéristique BLE pour l'adresse IP.

7.1.4.12 preferences

```
Preferences preferences [extern]
```

Objet Preferences pour stocker la configuration persistante.

7.1.4.13 safetyLockout

```
bool safetyLockout [extern]
```

Indique si le verrouillage de sécurité est activé

7.1.4.14 server

```
WebServer server [extern]
```

Objet WebServer pour gérer les requêtes HTTP.

7.1.4.15 shouldRestart

```
bool shouldRestart [extern]
```

Indique si un redémarrage est nécessaire.

7.1.4.16 targetAngle

```
int targetAngle [extern]
```

Angle cible pour le servomoteur.

7.1.4.17 tempMax

```
float tempMax [extern]
```

Variables de configuration de la température maximum.

7.1.4.18 tempMin

```
float tempMin [extern]
```

Variables de configuration de la température minimum.

7.1.4.19 wifi_pass

```
String wifi_pass [extern]
```

Mot de passe WiFi.

7.1.4.20 wifi_ssid

```
String wifi_ssid [extern]
```

SSID WiFi.

7.1.4.21 windowServo

```
Servo windowServo [extern]
```

Objet servomoteur pour contrôler l'ouverture de la fenêtre.

7.2 main.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef MAIN_HPP
00002 #define MAIN_HPP
00003
00004 #include <Arduino.h>
00005 #include <WiFi.h>
00006 #include <HTTPClient.h>
00007 #include <WebServer.h>
00008 #include <ArduinoJson.h>
00009 #include <ESP32Servo.h>
00010 #include <NimBLEDevice.h>
00011 #include <Preferences.h>
00012
00016
00020 #define SERVO_PIN 13
00024 #define CURRENT_SENSOR_PIN 34
00028 #define MAX_CURRENT_THRESHOLD 2500
00032 #define SERVICE_UUID "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
00036 #define CHAR_CONFIG_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"
00040 #define CHAR_IP_UUID "12345678-1234-1234-1234-1234567890ab"
00041
00045 extern Servo windowServo;
00046
00050 extern Preferences preferences;
00051
00055 extern WebServer server;
00056
00060 extern BLECharacteristic *pIpChar;
00061
00065 extern String wifi_ssid;
00069 extern String wifi_pass;
00070
00074 extern float latitude;
00078 extern float longitude;
00082 extern float tempMax;
00086 extern float tempMin;
00090 extern int aqiMax;
00091
00095 extern bool isOpen;
00096
00100 extern bool autoMode;
00104 extern int targetAngle;
00108 extern bool safetyLockout;
00112 extern int lastCurrentReading;
00116 extern int currentAngle;
00120 extern float lastTemp;
00124 extern int lastAQI;
00128 extern unsigned long lastWeatherCheck;
00132 extern bool shouldRestart;
00133
00139 bool checkSafety();
00140
00146 void setVMC(int angle);
00147
00152 void handleStatus();
00153
00158 void handleControl();
00159
00163 void setup();
00168 void loop();
00169
00171
00176 class ConfigCallbacks : public BLECharacteristicCallbacks {
00181     void onWrite(BLECharacteristic *pCharacteristic);
00182 };
00183
00184 #endif // MAIN_HPP

```

7.3 mobile/App.tsx File Reference

Composant principal de l'application mobile Smart VMC.

Variables

- [import React](#)
 - import [import](#) { SafeAreaView, ScrollView, StatusBar, StyleSheet, Alert, ActivityIndicator, View, Text } from 'react-native'
 - [import](#) AsyncStorage from [react](#) native async storage [async storage](#)
 - [import](#) TabHeader from components [TabHeader](#)
 - [import](#) SetupPanel from components [SetupPanel](#)
 - [import](#) DashboardPanel from components [DashboardPanel](#)
 - [import](#) ConfigurationPanel from components [ConfigurationPanel](#)
 - function App() export default [App](#)
- @function App*

7.3.1 Detailed Description

Composant principal de l'application mobile Smart VMC.

Gère la navigation entre les onglets (Setup, Dashboard, Configuration) et la communication avec l'ESP32 via BLE et API HTTP

7.3.2 Variable Documentation

7.3.2.1 App

```
function App () export default App ()
```

@function App

Composant racine de l'application Smart VMC

Gère :

- La navigation entre 3 onglets (Setup, Dashboard, Configuration)
- La connexion BLE à l'ESP32
- La communication avec l'API REST
- Le stockage persistant de l'adresse IP du serveur

Returns

{React.JSX.Element} Interface utilisateur complète

7.3.2.2 ConfigurationPanel

```
export default ConfigurationPanel
```

7.3.2.3 DashboardPanel

```
export default DashboardPanel
```

7.3.2.4 import

```
import { SafeAreaView, ScrollView, StatusBar, StyleSheet, Alert, ActivityIndicator, View, Text } from 'react-native'
```

7.3.2.5 React

```
import React
```

7.3.2.6 SetupPanel

```
export default SetupPanel
```

7.3.2.7 storage

```
import AsyncStorage from react native async storage async storage
```

7.3.2.8 TabHeader

```
export default TabHeader
```

7.4 mobile/components/ConfigurationPanel.tsx File Reference

Panneau de configuration des seuils de température et pollution.

Variables

- `import React`
- `import import { View, Text, StyleSheet, Alert } from 'react-native'`
- `onUpdateThresholds __pad0__`
- `onUpdateThresholds tMin`
- `onUpdateThresholds aqiMax`

Variables de configuration de la qualité de l'air maximum.

- `isLoading __pad1__`
- `const ConfigurationPanel`
 - @function ConfigurationPanel*
- `const onUpdateThresholds`
- `const isLoading`
- `const styles`

7.4.1 Detailed Description

Panneau de configuration des seuils de température et pollution.

7.4.2 Variable Documentation

7.4.2.1 __pad0__

`onUpdateThresholds __pad0__`

7.4.2.2 __pad1__

`isLoading __pad1__`

7.4.2.3 aqiMax

`onUpdateThresholds aqiMax`

Variables de configuration de la qualité de l'air maximum.

7.4.2.4 ConfigurationPanel

```
export default ConfigurationPanel
```

@function ConfigurationPanel

Formulaire de configuration des seuils (température, AQI) pour le mode Auto

Parameters

<code>{ConfigurationPanelProps}</code>	props - Propriétés du composant
--	---------------------------------

Returns

`{React.JSX.Element}` Carte de configuration

7.4.2.5 import

```
import { View, Text, StyleSheet, Alert } from 'react-native'
```

7.4.2.6 isLoading

```
const isLoading
```

7.4.2.7 onUpdateThresholds

```
const onUpdateThresholds
```

7.4.2.8 React

```
import React
```

7.4.2.9 styles

```
const styles
```

7.4.2.10 tMin

```
onUpdateThresholds tMin
```

7.5 mobile/components/dashboard/ControlPanel.tsx File Reference

Panneau de contrôle de la VMC (mode auto, commandes manuelles, slider).

Variables

- `import React` from `react`
- `import { View, Text, Switch, StyleSheet }` from `'react-native'`
- `import Slider` from `react native community slider`
- `isLocked __pad0__`
- `targetAngle __pad1__`
- `onToggleAuto __pad2__`
- `onCommand __pad3__`
- `onAngleChange __pad4__`
- `onResetSafety __pad5__`
- `export const ControlPanel`
 @function ControlPanel
- `export const isLocked`
- `export const targetAngle`
 Angle cible pour le servomoteur.
- `export const onToggleAuto`
- `export const onCommand`
- `export const onAngleChange`
- `export const onResetSafety`
- `const styles`

7.5.1 Detailed Description

Panneau de contrôle de la VMC (mode auto, commandes manuelles, slider).

7.5.2 Variable Documentation

7.5.2.1 __pad0__

`isLocked __pad0__`

7.5.2.2 __pad1__

`targetAngle __pad1__`

7.5.2.3 __pad2__

`onToggleAuto __pad2__`

7.5.2.4 __pad3__

`onCommand __pad3__`

7.5.2.5 __pad4__

`onAngleChange __pad4__`

7.5.2.6 __pad5__

`onResetSafety __pad5__`

7.5.2.7 ControlPanel

```
export const ControlPanel
```

```
@function ControlPanel
```

Panneau de contrôle principal de la VMC

Parameters

<code>{ControlPanelProps}</code>	props - Propriétés du composant
----------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Panneau de contrôle

Affiche deux modes :

- Mode verrouillé : Bouton de réarmement sécurité
- Mode normal : Switch auto/manuel, boutons ouvrir/fermer, slider d'angle

7.5.2.8 import

```
import { View, Text, Switch, StyleSheet } from 'react-native'
```

7.5.2.9 isLocked

```
export const isLocked
```

7.5.2.10 onAngleChange

```
export const onAngleChange
```

7.5.2.11 onCommand

```
export const onCommand
```

7.5.2.12 onResetSafety

```
export const onResetSafety
```

7.5.2.13 onToggleAuto

```
export const onToggleAuto
```

7.5.2.14 react

```
import React from react
```

7.5.2.15 slider

```
import Slider from react native community slider
```

7.5.2.16 styles

```
const styles
```

7.5.2.17 targetAngle

```
export const targetAngle
```

Angle cible pour le servomoteur.

7.6 mobile/components/dashboard/SensorRow.tsx File Reference

Ligne d'affichage des valeurs de capteurs (température et AQI).

Variables

- `import React` from `react`
- `import { View, Text, StyleSheet }` from `'react-native'`
- `aqi __pad0__`
- `export const SensorRow`
`@function SensorRow`
- `export const aqi`
- `const styles`

7.6.1 Detailed Description

Ligne d'affichage des valeurs de capteurs (température et AQI).

7.6.2 Variable Documentation

7.6.2.1 `__pad0__`

`aqi __pad0__`

7.6.2.2 `aqi`

`export const aqi`

Initial value:

```
=> {
  return (
    <View style={styles.row}>
      <Text style={styles.value}> {temp !== undefined && temp !== null ? temp.toFixed(1) : '--'}°C</Text>
      <Text style={styles.value}> AQI {aqi ?? '--'}</Text>
    </View>
  );
}
```

7.6.2.3 `import`

`import { View, Text, StyleSheet } from 'react-native'`

7.6.2.4 `react`

`import React` from `react`

7.6.2.5 `SensorRow`

`export const SensorRow`

`@function SensorRow`

Affiche la température et l'AQI sous forme de badges

Parameters

<code>{SensorRowProps}</code>	props - Propriétés du composant
-------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Ligne de capteurs

7.6.2.6 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({
  row: {
    flexDirection: 'row',
    justifyContent: 'center',
    gap: 20,
    marginBottom: 25,
  },
  value: {
    fontSize: 18,
    fontWeight: '600',
    color: '#374151',
  },
})
```

7.7 mobile/components/dashboard/StatusDisplay.tsx File Reference

Affichage de l'état global de la VMC (ouverte/fermée/verrouillée).

Variables

- `import React` from `react`
- `import { View, Text, StyleSheet }` from `'react-native'`
- `isLocked __pad0__`
- `export const StatusDisplay`
- `@function StatusDisplay`
- `export const isLocked`
- `const styles`

7.7.1 Detailed Description

Affichage de l'état global de la VMC (ouverte/fermée/verrouillée).

7.7.2 Variable Documentation

7.7.2.1 __pad0__

`isLocked __pad0__`

7.7.2.2 import

```
import { View, Text, StyleSheet } from 'react-native'
```

7.7.2.3 isLocked

```
export const isLocked
```

Initial value:

```
=> {  
  
  if (isLocked) {  
    return (  
      <View style={styles.container}>  
        <Text style={styles.emoji}></Text>  
        <Text style={[styles.text, { color: '#DC2626' }]}>  
          SÉCURITÉ ACTIVE  
        </Text>  
        <Text style={styles.subtext}>Moteur bloqué</Text>  
      </View>  
    );  
  }  
  
  return (  
    <View style={styles.container}>  
      <Text style={styles.emoji}>  
        {isOpen ? '' : ''}  
      </Text>  
      <Text style={ [  
        styles.text,  
        { color: isOpen ? '#007A5E' : '#EF4444' }  
      ]}>  
        {isOpen ? 'OUVERTE' : 'FERMÉE'}  
      </Text>  
    </View>  
  );  
}
```

7.7.2.4 react

```
import React from react
```

7.7.2.5 StatusDisplay

```
export const StatusDisplay
```

```
@function StatusDisplay
```

Affiche l'état visuel de la VMC avec emoji et texte coloré

Parameters

<code>{StatusDisplayProps}</code>	props - Propriétés du composant
-----------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Vue d'état

7.7.2.6 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({
  container: {
    alignItems: 'center',
    marginBottom: 20,
  },
  emoji: {
    fontSize: 80,
    marginBottom: 10,
  },
  text: {
    fontSize: 28,
    fontWeight: '800',
    letterSpacing: 0.5,
    textAlign: 'center',
  },
  subtext: {
    fontSize: 16,
    color: '#666',
    marginTop: 5,
  }
})
```

7.8 mobile/components/DashboardPanel.tsx File Reference

Tableau de bord affichant l'état, les capteurs et les commandes de la VMC.

Variables

- [import React](#)
- [import import { StyleSheet, View } from 'react-native'](#)
- [import NotificationService](#) from [services NotificationService](#)
- [onToggleAuto __pad0__](#)
- [onCommand __pad1__](#)
- [onAngleChange __pad2__](#)
- [onResetSafety __pad3__](#)
- [const DashboardPanel](#)
 @function DashboardPanel
- [const onToggleAuto](#)
- [const onCommand](#)
- [const onAngleChange](#)
- [const onResetSafety](#)
- [const styles](#)

7.8.1 Detailed Description

Tableau de bord affichant l'état, les capteurs et les commandes de la VMC.

7.8.2 Variable Documentation

7.8.2.1 __pad0__

[onToggleAuto __pad0__](#)

7.8.2.2 __pad1__

```
onCommand __pad1__
```

7.8.2.3 __pad2__

```
onAngleChange __pad2__
```

7.8.2.4 __pad3__

```
onResetSafety __pad3__
```

7.8.2.5 DashboardPanel

```
export default DashboardPanel
```

```
@function DashboardPanel
```

Affiche l'état, les capteurs et les commandes de la VMC

Parameters

<code>{DashboardPanelProps}</code>	props - Propriétés du composant
------------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Vue tableau de bord

7.8.2.6 import

```
import { StyleSheet, View } from 'react-native'
```

7.8.2.7 NotificationService

```
import NotificationService from services NotificationService
```

7.8.2.8 onAngleChange

```
const onAngleChange
```

7.8.2.9 onCommand

```
const onCommand
```

7.8.2.10 onResetSafety

```
const onResetSafety
```

7.8.2.11 onToggleAuto

```
const onToggleAuto
```

7.8.2.12 React

```
import React
```

7.8.2.13 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({
  container: {
    alignItems: 'center',
  },
})
```

7.9 mobile/components/SetupPanel.tsx File Reference

Panneau de configuration initiale avec saisie WiFi et localisation.

Variables

- `import React`
- `import import { View, Text, StyleSheet, Platform, Alert } from 'react-native'`
- `import Geolocation from react native geolocation service`
- `isScanning __pad0__`
- `onConnect __pad1__`
- `onConnect pass`
- `onConnect lat`
- `onConnect lon`
- `onConnect ip`
- `savedIp __pad2__`
- `const SetupPanel`
 - *@function SetupPanel*
- `const isScanning`
- `const onConnect`
- `const savedIp`
- `const styles`

7.9.1 Detailed Description

Panneau de configuration initiale avec saisie WiFi et localisation.

7.9.2 Variable Documentation

7.9.2.1 __pad0__

`isScanning` __pad0__

7.9.2.2 __pad1__

`onConnect` __pad1__

7.9.2.3 __pad2__

`savedIp` __pad2__

7.9.2.4 import

```
import { View, Text, StyleSheet, Platform, Alert } from 'react-native'
```

7.9.2.5 ip

`onConnect` ip

7.9.2.6 isScanning

```
const isScanning
```

7.9.2.7 lat

`onConnect` lat

7.9.2.8 lon

`onConnect` lon

7.9.2.9 onConnect

```
const onConnect
```

7.9.2.10 pass

```
onConnect pass
```

7.9.2.11 React

```
import React
```

7.9.2.12 savedIp

```
const savedIp
```

7.9.2.13 service

```
import Geolocation from react native geolocation service
```

7.9.2.14 SetupPanel

```
export default SetupPanel
```

```
@function SetupPanel
```

Formulaire d'envoi des identifiants WiFi + localisation via BLE

Parameters

<code>{SetupPanelProps}</code>	props - Propriétés du composant
--------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Carte de configuration initiale

7.9.2.15 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({
  sectionTitle: { fontSize: 20, fontWeight: '700', marginBottom: 15, color: '#007A5E' },
  statusText: { marginBottom: 15, textAlign: 'center', color: '#666', fontSize: 14 },
})
```

7.10 mobile/components/TabHeader.tsx File Reference

Barre d'onglets principale.

Variables

- `import React` from `react`
- `import { View, Text, TouchableOpacity, StyleSheet }` from `'react-native'`
- `onTabChange __pad0__`
- `const TabHeader`
`@function TabHeader`
- `const onTabChange`
- `const styles`

7.10.1 Detailed Description

Barre d'onglets principale.

7.10.2 Variable Documentation

7.10.2.1 `__pad0__`

`onTabChange __pad0__`

7.10.2.2 `import`

`import { View, Text, TouchableOpacity, StyleSheet }` from `'react-native'`

7.10.2.3 `onTabChange`

`const onTabChange`

Initial value:

```
=> {
  return (
    <View style={styles.header}>
      <Text style={styles.headerTitle}> Smart VMC</Text>
      <View style={styles.tabs}>
        <TouchableOpacity
          onPress={() => onTabChange('SETUP')}
          style={[styles.tab, activeTab === 'SETUP' && styles.activeTab]}
        >
          <Text style={styles.tabText}> Wifi Setup</Text>
        </TouchableOpacity>
        <TouchableOpacity
          onPress={() => onTabChange('DASHBOARD')}
          style={[styles.tab, activeTab === 'DASHBOARD' && styles.activeTab]}
        >
          <Text style={styles.tabText}> Contrôle</Text>
        </TouchableOpacity>
        <TouchableOpacity
          onPress={() => onTabChange('CONFIGURATION')}
          style={[styles.tab, activeTab === 'CONFIGURATION' && styles.activeTab]}
        >
          <Text style={styles.tabText}> Réglages</Text>
        </TouchableOpacity>
      </View>
    </View>
  );
}
```

7.10.2.4 react

```
import React from react
```

7.10.2.5 styles

```
const styles
```

7.10.2.6 TabHeader

```
export default TabHeader
```

```
@function TabHeader
```

Barre d'onglets principale

Parameters

<code>{TabHeaderProps}</code>	props - Propriétés du composant
-------------------------------	---------------------------------

Returns

`{React.JSX.Element}` En-tête avec trois onglets

7.11 mobile/components/ui/AppButton.tsx File Reference

Bouton applicatif stylé (primaire, danger, outline) avec état de chargement.

Variables

- `import React` from `react`
- `import { TouchableOpacity, Text, StyleSheet, TouchableOpacityProps, ActivityIndicator }` from `'react-native'`
- interface `AppButtonProps` extends `TouchableOpacityProps`
- `variant __pad0__`
- `isLoading __pad1__`
- export const `AppButton`
- `@function AppButton`
- export const `variant` = 'primary'
- export const `isLoading` = false
- export const `style`
- export const `disabled`
- export const `props`
- const `styles`

7.11.1 Detailed Description

Bouton applicatif stylé (primaire, danger, outline) avec état de chargement.

7.11.2 Variable Documentation

7.11.2.1 __pad0__

`variant __pad0__`

7.11.2.2 __pad1__

`isLoading __pad1__`

7.11.2.3 AppButton

```
export const AppButton
```

```
@function AppButton
```

Bouton réutilisable avec variantes et état de chargement

Parameters

<code>{AppButtonProps}</code>	props - Propriétés du composant
-------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Bouton stylé

7.11.2.4 disabled

```
export const disabled
```

7.11.2.5 import

```
import { TouchableOpacity, Text, StyleSheet, TouchableOpacityProps, ActivityIndicator } from  
'react-native'
```

7.11.2.6 isLoading

```
export const isLoading = false
```

7.11.2.7 props

```
export const props
```

7.11.2.8 react

```
import React from react
```

7.11.2.9 style

```
export const style
```

7.11.2.10 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({
  base: {
    paddingVertical: 14,
    borderRadius: 12,
    alignItems: 'center',
    justifyContent: 'center',
    shadowColor: '#000',
    shadowOffset: { width: 0, height: 2 },
    shadowOpacity: 0.1,
    shadowRadius: 3,
    elevation: 2,
  },
  text: {
    fontWeight: '700',
    fontSize: 16,
    letterSpacing: 0.5,
  },
  disabled: {
    opacity: 0.6,
  }
})
```

7.11.2.11 TouchableOpacityProps

```
interface AppButtonProps extends TouchableOpacityProps
```

Initial value:

```
{
  title: string
```

7.11.2.12 variant

```
export const variant = 'primary'
```

7.12 mobile/components/ui/AppInput.tsx File Reference

Champ de saisie stylé avec placeholder grisé

Variables

- `import React` from `react`
- `import { TextInput, StyleSheet, TextInputProps }` from `'react-native'`
- interface `AppInputProps` extends `TextInputProps`
- `const styles`

7.12.1 Detailed Description

Champ de saisie stylé avec placeholder grisé

7.12.2 Variable Documentation

7.12.2.1 import

```
import { TextInput, StyleSheet, TextInputProps } from 'react-native'
```

7.12.2.2 react

```
import React from react
```

7.12.2.3 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({
  input: {
    borderWidth: 1,
    borderColor: '#E0E0E0',
    borderRadius: 12,
    paddingVertical: 14,
    paddingHorizontal: 16,
    marginBottom: 15,
    fontSize: 16,
    backgroundColor: '#F9FAFB',
    color: '#1F2937',
  },
})
```

7.12.2.4 TextInputProps

```
interface AppInputProps extends TextInputProps
```

Initial value:

```
{}
```

```
export const AppInput: React.FC<AppInputProps> = ({ style, ...props }) => {
  return (
    <TextInput
      style={ [styles.input, style] }
      placeholderTextColor="#9CA3AF"
      {...props}
    />
  );
}
```

7.13 mobile/components/ui/Card.tsx File Reference

Conteneur carte avec ombre et arrondis.

Variables

- `import React` from `react`
- `import { View, StyleSheet, ViewStyle, ViewProps }` from `'react-native'`
- interface `CardProps` extends `ViewProps`
- `style __pad0__`
- `export const Card`
- `@function Card`
- `export const style`
- `export const props`
- `const styles`

7.13.1 Detailed Description

Conteneur carte avec ombre et arrondis.

7.13.2 Variable Documentation

7.13.2.1 `__pad0__`

`style __pad0__`

7.13.2.2 `Card`

`export const Card`

`@function Card`

Conteneur stylé type carte

Parameters

<code>{CardProps}</code>	props - Propriétés du composant
--------------------------	---------------------------------

Returns

`{React.JSX.Element}` Carte

7.13.2.3 `import`

`import { View, StyleSheet, ViewStyle, ViewProps }` from `'react-native'`

7.13.2.4 props

```
export const props
```

Initial value:

```
=> {  
  return <View style={[styles.card, style]} {...props}>{children}</View>;  
}
```

7.13.2.5 react

```
import React from react
```

7.13.2.6 style

```
export const style
```

7.13.2.7 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({  
  card: {  
    backgroundColor: 'white',  
    padding: 24,  
    borderRadius: 16,  
    marginBottom: 20,  
    shadowColor: '#000',  
    shadowOffset: { width: 0, height: 2 },  
    shadowOpacity: 0.1,  
    shadowRadius: 4,  
    elevation: 4,  
  },  
})
```

7.13.2.8 ViewProps

```
interface CardProps extends ViewProps
```

Initial value:

```
{  
  children: React.ReactNode
```

7.14 mobile/hooks/useBLE.ts File Reference

Hook React pour la communication Bluetooth avec l'ESP32.

Variables

- `import { useState, useEffect } from 'react'`
- `const SERVICE_UUID = "4fafc201-1fb5-459e-8fcc-c5c9c331914b"`
@constant {string} SERVICE_UUID
- `const CHAR_UUID = "beb5483e-36e1-4688-b7f5-ea07361b26a8"`
@constant {string} CHAR_UUID
- `const CHAR_IP_UUID = "12345678-1234-1234-1234-1234567890ab"`
@constant {string} CHAR_IP_UUID
- `const bleManager = new BleManager()`
@constant {BleManager} bleManager
- `export const useBLE`
@function useBLE

7.14.1 Detailed Description

Hook React pour la communication Bluetooth avec l'ESP32.

Gère le scan BLE, la connexion à l'ESP32, et l'échange de données

7.14.2 Variable Documentation

7.14.2.1 bleManager

```
const bleManager = new BleManager()
```

@constant {BleManager} bleManager

Instance globale du gestionnaire BLE

7.14.2.2 CHAR_IP_UUID

```
const CHAR_IP_UUID = "12345678-1234-1234-1234-1234567890ab"
```

@constant {string} CHAR_IP_UUID

UUID de la caractéristique BLE pour lire l'adresse IP de l'appareil

7.14.2.3 CHAR_UUID

```
const CHAR_UUID = "beb5483e-36e1-4688-b7f5-ea07361b26a8"
```

@constant {string} CHAR_UUID

UUID de la caractéristique BLE pour la configuration WiFi et localisation

7.14.2.4 import

```
import { useState, useEffect } from 'react'
```

7.14.2.5 SERVICE_UUID

```
const SERVICE_UUID = "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
```

@constant {string} SERVICE_UUID

UUID du service BLE principal

7.14.2.6 useBLE

```
export const useBLE
```

@function useBLE

Hook pour gérer les opérations Bluetooth

Fournit :

- Le statut de la connexion BLE
- L'état du scan
- Une fonction pour configurer l'ESP32 via BLE
- Une fonction pour récupérer l'IP de l'ESP32 via BLE

Returns

{Object} Objet contenant les états et fonctions BLE
{string} bleStatus - Message de statut de la connexion BLE
{boolean} isScanning - Indique si un scan est en cours
{Function} scanAndConfigure - Configure l'ESP32 avec WiFi/GPS
{Function} scanAndGetIp - Récupère l'adresse IP de l'ESP32

7.15 mobile/hooks/useWindowApi.ts File Reference

Hook React pour communiquer avec l'API smart-vmc.

Variables

- `import { useState, useCallback } from 'react'`
- `export interface WindowState`
- `temp __pad0__`
- `aqi __pad1__`
- `targetAngle __pad2__`
- `autoMode __pad3__`
- `safetyLockout __pad4__`
- `currentLoad __pad5__`
- `tMax __pad6__`
- `tMin __pad7__`
- `aqiMax __pad8__`
- `export const useWindowApi`
 @function useWindowApi
- `const sendAngle`
 @function sendAngle
- `const toggleAutoMode`
 @function toggleAutoMode
- `const resetSafety`
 @function resetSafety
- `const updateThresholds`
 @function updateThresholds
- `return`

7.15.1 Detailed Description

Hook React pour communiquer avec l'API smart-vmc.

Fournit les appels REST pour lire l'état et piloter la VMC (ouverture, mode auto, seuils, sécurité).

7.15.2 Variable Documentation

7.15.2.1 __pad0__

`temp __pad0__`

7.15.2.2 __pad1__

`aqi __pad1__`

7.15.2.3 __pad2__

`targetAngle __pad2__`

7.15.2.4 __pad3__

`autoMode` __pad3__

7.15.2.5 __pad4__

`safetyLockout` __pad4__

7.15.2.6 __pad5__

`currentLoad` __pad5__

7.15.2.7 __pad6__

`tMax` __pad6__

7.15.2.8 __pad7__

`tMin` __pad7__

7.15.2.9 __pad8__

`aqiMax` __pad8__

7.15.2.10 import

```
import { useState, useCallback } from 'react'
```

7.15.2.11 resetSafety

```
const resetSafety
```

Initial value:

```
= async () => {  
  try {  
    const response = await fetch(`http:  
      method: 'POST',  
      body: JSON.stringify({ resetSafety: true })  
    });  
    const json = await response.json();  
    setWindowState(json.state || json);  
    Alert.alert("Info", "Système réarmé. Vérifiez qu'il n'y a plus d'obstruction.");  
  } catch (e) {  
    Alert.alert("Erreur", "Impossible de réarmer");  
  }  
}
```

@function resetSafety

Réarme le verrouillage de sécurité

7.15.2.12 return

return

Initial value:

```
{
  windowState,
  fetchStatus,
  sendCommand,
  sendAngle,
  toggleAutoMode,
  resetSafety,
  updateThresholds,
  isLoading
}
```

7.15.2.13 sendAngle

const sendAngle

Initial value:

```
= async (angle: number) => {
  try {
    const response = await fetch(`http:
      method: 'POST',
      body: JSON.stringify({ angle })
    });
    const json = await response.json();
    setWindowState(json.state || json);
  } catch (e) {
    console.log(e);
  }
}
```

@function sendAngle

Positionne la fenêtre à un angle précis

Parameters

<i>{number}</i>	angle - Angle cible (0-90)
-----------------	----------------------------

7.15.2.14 toggleAutoMode

const toggleAutoMode

Initial value:

```
= async (value: boolean) => {
  try {
    const response = await fetch(`http:
      method: 'POST',
      body: JSON.stringify({ autoMode: value })
    });
    const json = await response.json();
    setWindowState(json.state || json);
  } catch (e) {
    Alert.alert("Erreur", "Impossible de changer le mode");
  }
}
```

@function toggleAutoMode

Active/désactive le mode automatique

Parameters

<code>{boolean}</code>	value - true pour activer
------------------------	---------------------------

7.15.2.15 updateThresholds

const updateThresholds

Initial value:

```
= async (tMax: number, tMin: number, aqiMax: number) => {
  setIsLoading(true);
  try {
    const response = await fetch(`http:
      method: 'POST',
      body: JSON.stringify({ tMax, tMin, aqiMax })
    });
    const json = await response.json();
    setWindowState(json.state || json);
    Alert.alert("Succès", "Paramètres sauvegardés !");
  } catch (e) {
    Alert.alert("Erreur", "Echec sauvegarde");
  } finally {
    setIsLoading(false);
  }
}
```

@function updateThresholds

Met à jour les seuils température/AQI

Parameters

<code>{number}</code>	tMax - Seuil max
<code>{number}</code>	tMin - Seuil min
<code>{number}</code>	aqiMax - Seuil AQI max

7.15.2.16 useWindowApi

export const useWindowApi

@function useWindowApi

Hook pour appeler l'API HTTP de l'ESP32

Parameters

<code>{string}</code>	serverIp - Adresse IP de l'ESP32 (ex: "192.168.1.50")
-----------------------	---

Returns

`{object}` API du hook

`{WindowState|null}` return.windowState - Dernier état connu

`() => Promise<void>` return.fetchStatus - Récupère l'état courant

`{(action:'open'|'close') => Promise<void>}` return.sendCommand - Ouvre/ferme

`{(angle:number) => Promise<void>}` return.sendAngle - Définit un angle précis

`{(value:boolean) => Promise<void>}` return.toggleAutoMode - Active/désactive le mode auto

`() => Promise<void>` return.resetSafety - Réarme la sécurité

`{(tMax:number,tMin:number,aqiMax:number)=>Promise<void>}` return.updateThresholds - Met à jour les seuils

`{boolean}` return.isLoading - Indique si une requête de seuils est en cours

7.15.2.17 WindowState

```
export interface WindowState
```

Initial value:

```
{  
  isOpen: boolean
```

7.16 mobile/services/NotificationService.ts File Reference

Classes

- class [NotificationService](#)
Service pour gérer les notifications locales.

Variables

- [import notifee](#)

7.16.1 Variable Documentation

7.16.1.1 notifee

```
import notifee
```