**Web.cpp**

**Web(Application \*c,SensorManager \*s, Thermostat \* t,WifiManager \* lw, IRManager \*ir, TimeNtp \* ti, DomoticzBroadcaster \*dom) : server(80) Demander**

Constructeur de Web:

Initialise les paramètres lconfig, lsensor, lthermos, lwifi, lIR, ltimentp et ldomo avec les attributs passés en paramètre.

**void init()**

Démarrage du serveur web.

Initialiser la page de lecture de la température et de l'humidité.

Reboot.

Initialisation de la page de saisie et lecture de la valeur de l'étalonnage.

Initialisation de la page de saisie et lecture de la valeur de l’étalonnage.

Requête du statut du capteur de présence.

Envoi d'une trame Infrarouge.

On en mode Eco.

Off.

Web pages initialisation :

-Initialiser la page racine du service Web.

-Initialiser la page de lecture et de modification des valeurs de connexion Wifi.

-Log page.

-config page.

-sensor page.

-formulaire config.

-formulaire mode racine.

-Initialiser la page de config planning

-Formulaire Domoticz :

-Dombroadcaster

-Domoticz save

-Check password form

-Formulaire addplanning

-Delete planning

**void handle\_root()**

Page racine.

Gestion de la connexion.

**void sensor\_print()**

Page Capteur.

**void sendir()**

Récupération des valeurs saisies pour l’IR et envoie lorsque l’utilisateur fait appel au service sendir.

**void modee()**

Association des valeurs et des variables pour le mode de fonctionnement en fonction de ce que l’utilisateur a entré sur la page web.

**void wifiCredentials()**

Quand l'utilisateur fait appel au service /wificredentials.

**void calibrateT()**

Quand l'utilisateur fait appel au service /calibrateTemp.

**void calibrateH()**

Quand l'utilisateur fait appel au service /calibrateHum.

**void saveconfig()**

Mise en mémoire de tous les champs saisis pour la configuration.

**void delplanning()**

Modification/Suppression planning.

**void saveplanning()**

Mise en mémoire des champs saisies.

**void savedomo()**

Mise en mémoire des champs saisies.

**void logging()**

Gestion de la connexion.

**Méthodes pour la création du contenue html ci-dessous.**

**bool hasAccess(Application \*c,String & str)**

Gestion de l’absence de mot de passe avec message d’erreur.

**void forehead(String & str)**

**void head(String & str)**

**void end(String & str)**

**void Add\_refresh(String & str)**

**void Add\_table(String & str, const String & title)**

**void Add\_form(String & str, const String & url)**

**void Add\_input(String & str, const String & id, const String & label, const String val)**

**void Add\_button(String & str, const String & type, const String & label)**

**void Add\_hrefbutton(String & str, const String & type, const String & label,const String & url)**

Mise en forme de la page web grâce à des string auxquels on concatène à chaque fois le code en html nécessaire.

**void sensor(Application \*c,SensorManager \*sen,String &str)**

Récupération des infos capteurs.

Mise en forme de l’affichage d’un capteur.

**void root(Application \*c,SensorManager \*sen, DomoticzBroadcaster \*ldomo,String &str)**

Mise en forme de la page racine avec les actions à faire si un champ est remplie ou un choix sélectionné ou une valeur forcée.

**void mode(String &opt,String mode)**

Mise en forme de l’affichage des modes avec les actions à faire si un champ est remplie ou un choix sélectionné ou une valeur forcée.

**void model(String &opt,String mdl)**

Mise en forme de l’affichage du modelé de la pompe avec les actions à faire si un champ est remplie ou un choix sélectionné ou une valeur forcée.

**void gpio(String &opt , uint8\_t gpio)**

Mise en forme de l’affichage des GPIO.

**void formconfig(Application \*c, Thermostat \*t, String &str,String &error)**

Mise en forme de l’affichage du formulaire config.

**void formplanning(Thermostat \*t,String &str,String &error)**

Mise en forme de l’affichage de planning en fonction des valeurs définis pour un thermostat

**void formlog(Application \*lc,String &str)**

Mise en forme de l’affichage des logs de l’application.

**void formlogging(String &str,String &error)**

Mise en forme de l’affichage formulaire de connexion.

**void formdomo(Application \*lc,String &str, String &error)**

Mise en forme de l’affichage du formulaire Domoticz.

|  |
| --- |
| Web |
| -httpUpdater : ESP8266HTTPUpdateServer  -lconfig : Application \*  -lsensor : SensorManager \*  -lthermos : Thermostat \*  -lwifi : WifiManager \*  -lIR : IRManager \*  -ltimentp : TimeNtp \*  -ldomo : DomoticzBroadcaster \*  +server ESP8266WebServer |
| -head(String & str)  -forehead(String & str)  -end(String& str)  -gpio(String &opt,uint8\_t gpio)  -model(String &opt,String mdl)  -mode(String &opt,String mode)  -handle\_root()  -sensor\_print()  -sendir()  -modee()  -wifiCredentials()  -calibrateT()  -calibrateH()  -saveconfig()  -delplanning()  -saveplanning()  -savedomo()  -void logging()  -Add\_refresh(String & str)  -Add\_table(String & str, const String &)  -Add\_form(String & str, const String & url)  -Add\_input(String & str, const String & id, const String & label, const String val)  -Add\_button(String & str, const String & type, const String & label)  -Add\_hrefbutton(String & str, const String & type, const String & label,const String & url)  +Web(Application \*lconfig, SensorManager \*lsensor, Thermostat \* lthermos, WifiManager \* lwifi, IRManager \*ir, TimeNtp \* ti,DomoticzBroadcaster \*dom) : Web  +init()  +root(Application \*c,SensorManager \*sen,DomoticzBroadcaster \*ldomo,String &str)  +formconfig() : String  +formconfig(Application \*c, Thermostat \*t, String &str, String &error)  +sensor(Application \*c,SensorManager \*sen,String &str)  +formplanning(Thermostat \*t,String &str,String &error)  +formlog(Application \*lc,String &str)  +formlogging(String &str,String &error)  +hasAccess (Application \*c,String &str) : bool  +formdomo(Application \*lc,String &str, String &error) |