# 



AppsGate

“Applications Gateway”



Cahier “Comment faire pour …”

Date

Juin 2015

Abstract

Ce cahier est un condensé du mode d’emploi du logiciel développé par UJF/LIG dans le cadre du projet AppsGate. Ce cahier complète les living documents de spécification détaillée de l’implémentation du serveur et client qui eux, s’adressent seulement au développeur. L’objectif de ce cahier est d’abord de faciliter la prise en main du logiciel par un utilisateur (voire développeur) naif : installation, work-arounds, extensions.

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Authors** | **Changes** |
| V0.0 | 15 juin 15 | J. Coutaz | template |
|  |  |  |  |

Disclaimer

The information in this document is subject to change without notice. The Members of the AppsGate Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the AppsGate Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Copyright

The document is proprietary of the AppsGate UJF/LIG members. No copying or distributing, in any form or by any means, is allowed without the prior written agreement of the owner of the property rights.

****Glossary****

|  |  |
| --- | --- |
| **Term** | **Meaning** |
|  |  |

Executive Summary

Ce cahier est un condensé du mode d’emploi du logiciel développé par UJF/LIG dans le cadre du projet AppsGate. Ce cahier complète les living documents de spécification détaillée de l’implémentation du serveur et client qui eux, s’adressent seulement au développeur. L’objectif de ce cahier est d’abord de faciliter la prise en main du logiciel par un utilisateur (voire développeur) naif : installation, work-arounds, extensions.

Table of Contents

[1. Introduction 6](#_Toc342490790)

[2. Development 6](#_Toc342490791)

[3. Conclusion 6](#_Toc342490792)

[4. References 6](#_Toc342490793)

# Où retrouver le code source

# Dépendances : quelles versions de code utilisé

JVM, JDK

OSGi

ApAM

Backbone ?

Navigateurs

Etc.

# Quel environnement de développement et comment l’installer

# Comment récupérer une configuration EnOcean depuis celle d’un miniPC

La configuration du MiniPC est la suivante : le serveur et le client sont installés sous le répertoire /opt/appsgate.

La configuration enocean est dans le répertoire conf, il s'agit du ficher enocean.conf.db.

/opt/appsgate/server/conf/enocean.conf.db

Il suffit de sauvegarder ce fichier une fois que l'appairage est effectué. Si la configuration disparaît ou que l'on souhaite utiliser l'appairage sur un autre pc, il suffit de le recopier.

# Logs de console : comment les effacer et comment les visualiser

Les logs de console sont de simples traces d’excutions. Ils sont situés dans le repertoire :

* /opt/appsgate/server/log

Une fois que l'application est arrêtée, il n'y a aucun souci pour effacer les traces de logs.

Ils sont enregistrés de manière rotative.

Les traces complètes sont dans les fichiers shortterm qui sont stockées de manière rotative, et le fichier mid-term/longterm ne contient que les niveaux d'information.

Les fichiers midterm correspondent aux logs d'une exécution du serveur.

# Comment modifier les terminaux de la grammaire (fichiers JSON) et autres messages à l’adresse du end-user

Tous les terminaux sont definis dans les fichiers json situés sur le serveur du client

/opt/appsgate/client/app/locales sous en et fr

Il y a 2 fichiers dans chaque répertoire :

* translation.json => le fichier principal
* debugger.json => le fichier concernant les messages des timelines

# Comment ajouter un nouvel équipement de bout en bout (du niveau core au niveau langage SPOK language).

La documentation en anglais est disponible dans le document intitulé : AppsGate-D2.4b-HMI Middleware.V1.7.

Au niveau système, il s'agit de faire en sorte qu'il soit reconnu dans le système en tant que CoreObject.

Du côté client, l'objectif est plus complexe, car aucune standardisation n'a été faite. Il faut donc prévoir le cas de la télécommande, ainsi que celui du langage. Ces 2 parties sont décrites dans le document.

Concernant les traces, la procédure n'est pas documentée. Concernant le graphe de dépendance, il faut rajouter les noms dans le fichier debugger.json.

# Comment réutiliser les données de la BD

Les données de la base de données sont stockées dans une base de type noSQL appelée Mongo. Il est possible d'interroger cette base de données via un client : robomongo. Malgré tout, cette utilisation n'est pas très intuitive.

Il a donc été mis en place un script qui permet d'exporter les données au format CSV. Cet export ne concerne que les données correspondant à l'historique.

Pour réaliser cet export, il suffit de faire la commande suivante

cd /opt/appsgate/server

sh ./exec.sh

2 fichiers sont alors générés devices.csv ainsi que programs.csv.

## devices.csv

Ce fichier contient toutes les traces provenant des équipements sous le format csv

**timestamp, id, type, "name", location.id, location.name, event.type, event.state**

timestamp

*id :* l'identifiant de l'équipement

*type* : le type d'équipement

*name* : le nom de l'équipement tel qu'il a été choisi par l'utilisateur et tel qu'il apparaît dans la base au moment de la trace

*location.id* : un identifiant du lieu dans lequel se trouve l'équipement

*location.name* : le nom choisi par l'utilisateur pour ce lieu

*event.type* : suivant le type d'équipement, un nom est donné qui décrit la nature de l'événement

event.state : élément json décrivant l'état de l'équipement

## programs.csv

Ce fichier contient toutes les traces provenant des programmes sous le format csv.

**timestamp, id, "name", state, event, device.id, device.name**

*id :* l'identifiant du programme

*name* : le nom du programme

*state* : l'état du programme ('incomplete', 'processing', 'invalid', 'limping', 'deployed')

*event*: le type de l'événement ('user' pour un changement du programme, 'write' pour une action du programme, 'read' pour une lecture d'une information par le programme)

*device.id* : l'identifiant de l'équipement qui a été lu ou modifié (quand c'est le cas)

*device.name* : le nom de l'équipement qui a été lu ou modifié (quand c'est le cas)

# Autres choses ?