



Institut Supérieur d'Informatique, de
Modélisation et de leurs Applications

Campus des Cézeaux
1 rue de la Chébarde
TSA 60125
CS 60026
63 178 Aubière CEDEX

RAPPORT D'INGENIEUR

INTEGRATION D'APPLICATION EN ENTREPRISE

FILIERE : GENIE LOGICIEL ET SYSTEMES INFORMATIQUES

GESTION DE RENDEZ-VOUS DE MEDECINS

Présenté par : Pierre CHEVALIER & Benoît GARÇON

TABLE DES FIGURES ET ILLUSTRATIONS

FIGURE 1 - SOURCE DE HELLO	ERREUR ! SIGNET NON DEFINI.
FIGURE 2 - EXECUTION ET RESULTATS DE HELLO.....	ERREUR ! SIGNET NON DEFINI.
FIGURE 3 - PARALLELISATION DE LA BOUCLE EXTERNE	ERREUR ! SIGNET NON DEFINI.
FIGURE 4 - SECTION CRITIQUE.....	ERREUR ! SIGNET NON DEFINI.
FIGURE 5 - RECAPITULATIF DE LA PREMIERE PARALLELISATION	ERREUR ! SIGNET NON DEFINI.
FIGURE 6 - PERFORMANCES DE LA PREMIERE PARALLELISATION GROUPEES PAR TAILLE DE FENETRE.....	ERREUR ! SIGNET NON DEFINI.
FIGURE 7 - COURBES DE PERFORMANCE DE LA PREMIERE PARALLELISATION	ERREUR ! SIGNET NON DEFINI.
FIGURE 8 - TABLE DES RESULTATS POUR DU SCHEDULING STATIC	ERREUR ! SIGNET NON DEFINI.
FIGURE 9 - RESULTATS AVEC STATIC SCHEDULING POUR UNE IMAGE 1024X1024	ERREUR ! SIGNET NON DEFINI.
FIGURE 10 - RESULTATS POUR UNE IMAGE 1024X1024 PAR NOMBRE MAXIMUM DE THREADS	ERREUR ! SIGNET NON DEFINI.
FIGURE 11 - TABLEAU DES RESULTATS AVEC DU SCHEDULING DYNAMIQUE	ERREUR ! SIGNET NON DEFINI.
FIGURE 12 - COURBE DES RESULTATS POUR LE SCHEDULING DYNAMIQUE ..	ERREUR ! SIGNET NON DEFINI.

TABLE DES MATIERES

Table des figures et illustrations	i
Table des matières.....	ii
Introduction.....	1
Chapitre 1 : Analyse.....	2
Chapitre 2 : Conception.....	3
2.1 Métaprogrammation	3
2.2 Adaptation	3
Chapitre 3 : Résultat	4
Conclusion	5

INTRODUCTION

Dans le cadre du cours d'intégration d'application en entreprise de la filière Génie Logiciel et Systèmes Informatiques de la troisième année d'étude à l'ISIMA, il nous a été demandé de produire une application J2EE fournissant un service web permettant de gérer les rendez-vous d'un cabinet de médecins.

L'objectif est donc de développer une interface de programmation, ou API, sous la forme d'un service web REST consommable par des clients tiers par le protocole HTTP afin de mettre en pratique tout ce qui a été vu en cours.

Ce rapport a pour but de décrire notre projet, notre analyse et les pistes de développement explorées ainsi que le résultat final de notre travail. Nous commencerons donc par une analyse du sujet afin de mettre en évidence les axes principaux du travail. Nous verrons ensuite comment nous avons conçu notre application et quels outils ont eu un rôle majeur dans ce processus. Enfin nous détaillerons le rendu final de cette application, son fonctionnement, son déploiement et ses tests.

Chapitre 1 : ANALYSE

Chapitre 2 : CONCEPTION

2.1 Métaprogrammation

Utilisation de netbeans à partir du sql

2.2 Adaptation

Adaptation du code

Chapitre 3 : RESULTAT

Présentation du script de déploiement + Tests et utilisation du service

CONCLUSION

Nous avons donc réussi à produire un service web répondant aux critères du sujet. Notre application rdvMed conçu en J2EE permet donc de manipuler des entités persistées comme des médecins et des patients, avec toutes les fonctionnalités CRUD essentielles. Il est aussi possible d'organiser des rendez-vous, de les modifier et les annuler, et aussi de lister les créneaux libres pour les médecins.

L'ensemble de ce travail est disponible librement sur GitHub et géré par intégration continue sur Travis CI permettant un développement ouvert et collaboratif. Un simple script disponible dans la release permet de télécharger, builder, déployer et tester simplement l'application avec une seule commande (sur Travis ou sur une machine vierge).

Tout ceci a pu être développé facilement grâce aux outils de génie logiciel à notre disposition : les outils de métaprogrammation de NetBeans pour générer du code basique, l'outil Ant pour gérer la compilation, le packaging, la documentation et les tests automatiquement et la CI pour pouvoir tester notre déploiement automatiquement sans recréer manuellement une nouvelle machine virtuelle pour chaque essai.

Notre application répond au cahier des charges initial mais pourrait fournir encore plus. Par exemple nous avons pensé fournir un client web Angular consommant notre service. Nous l'avons fait grâce à NetBeans qui a généré automatiquement une interface graphique basique à partir de notre service mais nous l'avons supprimé puisqu'il sortait du cadre du projet. Il pourrait être intéressant donc de poursuivre les travaux sur une partie client et ajouter au serveur des éléments lui permettant d'être mis à l'échelle dans le « cloud » (architecture en micro-services, circuit breaker, feature flipping, etc.).