

## Rapport Projet RICM5 Contribution JHipster UGA

COURTIAL Julien, GROS-DAILLON Hugo, LAFRASSE Cédric, TERRIER Bastien



Enseignants responsables : DONSEZ Didier, GEOURJON Anthony, TOURANCHEAU Bernard

Printemps 2019

# Sommaire

<b>Introduction</b>	<b>2</b>
<b>Contributions open-source</b>	<b>3</b>
Rappel du sujet	3
Technologies employés	3
<b>Réalisations techniques</b>	<b>4</b>
Sauvegarde automatique de bases de données	4
Page d'administration des bases de données	4
Traduction automatique des entités et des messages	4
Gestion de quotas de création d'entités	4
Intégration d'Angular Material	4
Intégration de la plateforme de paiement Stripe	5
Intégration de la plateforme de paiement Paypal	5
Intégration d'un chatbot Rasa	5
Les étapes dans la réalisation d'un générateur	5
<b>Gestion de projet</b>	<b>6</b>
Récapitulatif	7
<b>Métriques logicielles</b>	<b>8</b>
<b>Conclusion</b>	<b>8</b>
<b>Glossaire</b>	<b>9</b>
<b>Bibliographie</b>	<b>9</b>

# Introduction

Tous les membres de notre équipe de projet avaient déjà utilisé le générateur open-source de code JHipster. Par nos expériences passées, nous avons vu les nombreux avantages de l'utilisation de générateurs pour le développement.

La programmation générative permet notamment de simplifier et d'accélérer le processus de création de site internet. L'exécution d'une simple commande permet d'ajouter et d'intégrer des milliers de lignes de code. Lors du projet de RICM5 nommé ECOM (e-commerce), nous avons par exemple rencontré des difficultés pour intégrer Angular Material. Cette intégration faite à la main par des milliers de développeurs peut être faite automatiquement à l'aide d'un générateur.

L'année dernière, deux membres du groupe avait réalisés un générateur permettant l'intégration d'une carte Leaflet-OpenStreetMap dans un projet JHipster. Convaincu par l'utilité de la génération, nous avons travaillé pendant ces semaines de projet à la contribution au projet open-source JHipster.

Nous allons, dans ce rapport, rendre compte du travail que nous avons effectué pendant les semaines de projet. Le fruit de notre travail étant la création de 8 générateurs, tous disponible sur le marketplace de JHipster. Ils s'accompagnent de leurs applications de démonstrations pour comprendre leur utilité dans un projet JHipster.



# Contributions open-source

## Rappel du sujet

Le sujet sur lequel nous avons travaillé est le suivant : “Contribution open-source au projet JHipster”. Il consiste à la réalisation de contributions open-source au générateur open-source JHipster. Les contributions seront sous forme de générateurs JHipster. Ils permettront de faciliter et d’accélérer le développement ou l’intégration de nouveaux outils à des projet JHipster.

## Technologies employés

Pour pouvoir développer des générateurs modulaires pour des projets **JHipster**, il nous était important de bien connaître les technologies utilisées par **JHipster**. Nous avons donc beaucoup travaillé avec **Angular**, **React**, **Spring** ainsi qu’avec **TypeScript**, **HTML** et **CSS**.

Pour la réalisation de nos générateurs nous avons utilisé un **générateur de template de générateur**. Il s’agit d’un générateur permettant de générer un squelette d’application qui permettra de générer à son tour différentes choses dans un projet **JHipster**. Il s’agit de <https://github.com/jhipster/generator-jhipster-module>

L’ensemble des générateurs que nous avons développés ont tous été réalisés en **JavaScript** avec la plateforme **Node.js**. Nous avons utilisés beaucoup de packages open sources disponibles sur <https://www.npmjs.com/> pour nous faciliter la tâche.

Enfin, pour deux de nos générateurs (“Sauvegarde automatique de bases de données” et “Page d’administration de bases de données”) nous avons dû utiliser **Docker**. Nous avons donc dû nous former rapidement à cette technologie que nous découvrons tous les quatre.

## Réalisations techniques



### Sauvegarde automatique de bases de données

Ce générateur permet de mettre en place un système de backup automatique, configurable, de différents type de base de données : MySQL, MariaDB, PostgreSQL, MongoDB.



### Page d'administration des bases de données

Ce générateur met à disposition une page d'administration de base de données. Ces pages d'administration sont présentes pour : MySQL, MariaDB, PostgreSQL, MongoDB.



### Traduction automatique des entités et des messages

Lorsque l'on génère un projet dans plusieurs langues avec JHipster, tout ce qui est généré de base est déjà traduit par JHipster.

Ce générateur permet de récupérer l'ensemble des messages qui n'ont pas été traduits automatiquement par JHipster, comme par exemple l'ajout des entités ou des messages par les développeurs et grâce à une API key de Google Traduction, de les traduire automatiquement et en quelques secondes, depuis le langage natif de l'application (que vous avez sélectionné lors de la création de votre projet JHipster) vers l'ensemble des langues disponibles sur votre application.



### Gestion de quotas de création d'entités

Ce module vous permet de créer des entités qui vous permettent de gérer un quota de création d'entité pour chaque utilisateur. Tout ce que vous aurez à faire est d'entrer un quota pour chaque utilisateur directement dans la base de données. Si vous n'entrez pas de quota, il sera de base illimité pour chaque utilisateur.



### Intégration d'Angular Material

Ce module intègre à un projet Jhipster les différentes dépendances nécessaires à l'utilisation des composants d'Angular Material.

## Intégration de la plateforme de paiement Stripe

Ce module intègre la plateforme de paiement STRIPE à un projet JHipster. (Il comprend une page de paiement Web et une entité de paiement JHipster qui récupère les informations quand un paiement stripe est effectué.). Il permettra de réaliser rapidement et simplement des paiement par carte bleu sur votre application JHipster.



## Intégration de la plateforme de paiement Paypal

Ce module vous permet d'intégrer le paiement Paypal dans un projet JHipster qui fonctionne avec angular. Il crée une nouvelle page de paiement paypal sur votre projet jhipster et ajoute une nouvelle entité (PaypalCompletedPayments) qui récupère les informations quand un paiement paypal est effectué.



## Intégration d'un chatbot Rasa

Ce module intègre une interface de chat à un projet JHipster React ou Angular. Ce chat peut se connecter à un bot utilisant la Rasa Stack et utilisant la librairie SocketIO. Le module React utilisé est le module utilisé sur le site de Rasa et nous avons développé le module pour Angular car il était inexistant.

# Les étapes dans la réalisation d'un générateur

En règle général, nous avons suivi les étapes suivantes pour le développement d'un générateur :

1. Création d'un projet JHipster simple.
2. Intégration de la fonctionnalité dans le projet, à la main.
3. Création d'un générateur grâce au module "generator-jhipster-module"
4. Développement de la génération du code écrit à l'étape 2, de façon modulaire pour pouvoir fonctionner avec toutes les applications générées par JHipster.
5. Test d'intégration de la fonctionnalité en exécutant le générateur sur différentes applications.
6. Publication de la contribution sur github, npmjs et sur la JHipster Marketplace.
7. Réalisation d'une application de démonstration utilisant le générateur
8. Publication de l'application de démonstration sur gitHub.

# Gestion de projet

Notre projet de fin d'étude s'est déroulé sur 5 semaines. Nous avons décidé d'appliquer la méthode scrum avec des sprints d'une durée de 1 semaine.

Le premier sprint a été un sprint de formation et de recherches. Nous n'avons rien publié lors de ce premier sprint. Nous avons recherchés puis testés les API de traduction gratuites existantes puis avons conclu que l'utilisation de Google Translate avec une API Key était une bien meilleure solution. Nous nous sommes aussi formés sur Docker pour la réalisation des générateurs en liens avec des bases de données.

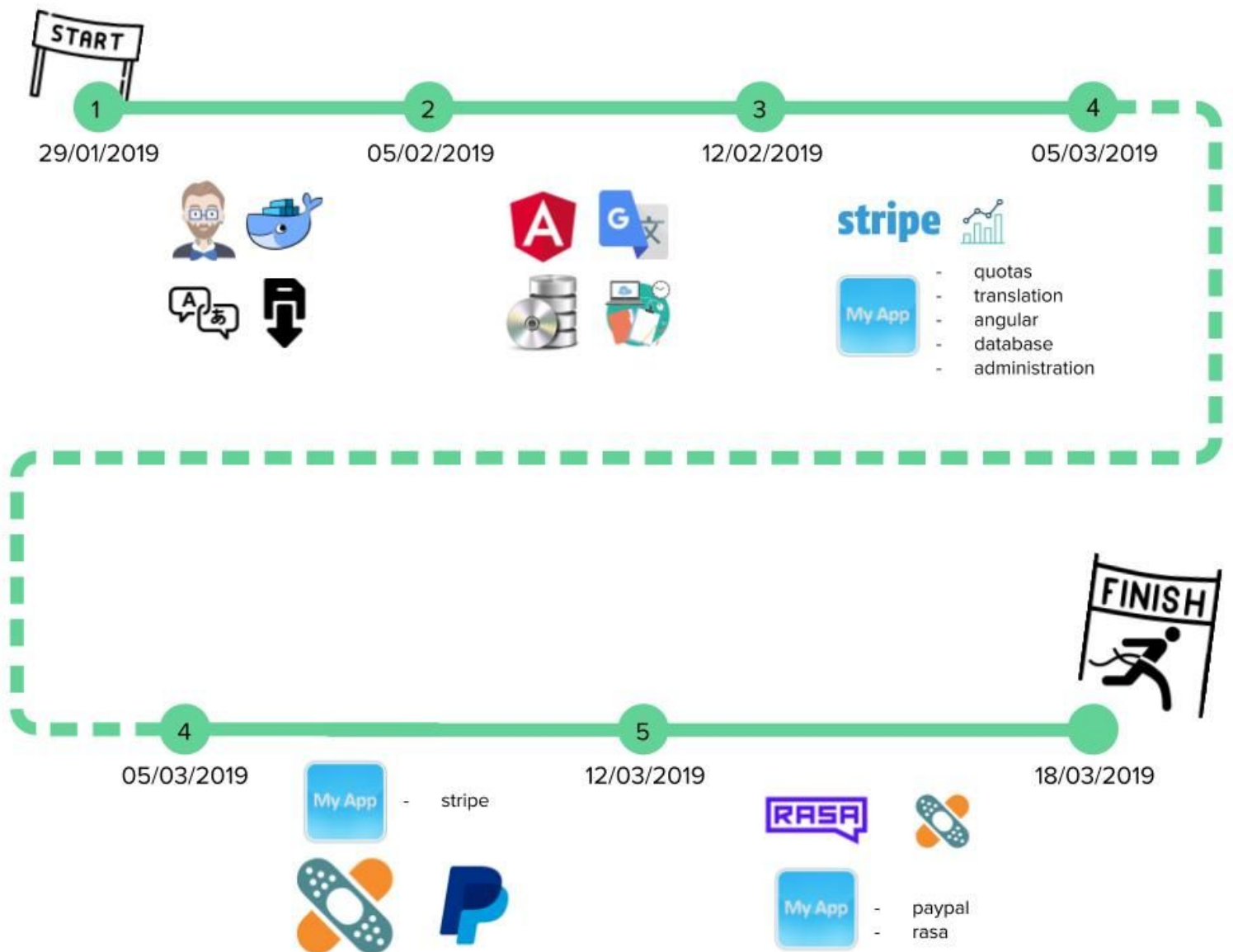
Lors du deuxième sprint, nous avons réalisés et publiés 4 générateurs : La sauvegarde automatique et l'administration des bases de données, la traduction automatique des entités et des messages avec google translate ainsi qu'un module d'intégration d'angular material à JHipster.

Pour le troisième sprint, nous avons réalisés et publiés 2 nouveaux générateurs. Un permettant d'intégrer la plateforme de paiement Stripe à un projet JHipster, l'autre permettant de gérer des quota de création d'entités en fonction des utilisateur. Nous avons aussi réalisés des applications de démonstrations pour tous les générateurs que nous avons réalisés jusqu'à maintenant.

Lors de notre quatrième sprint, nous nous sommes rendu compte que certaines choses n'allaient pas dans certains de nos générateurs. Nous avons donc apporté des correctifs pour les générateurs : Angular Material, Traduction automatique, Stripe et Sauvegarde automatique de base de données. Nous avons aussi développé un nouveau générateur lors de ce Sprint, l'intégration de la plateforme de paiement Paypal. Enfin, nous avons aussi réalisé une application de démonstration pour le générateur de paiement Stripe.

Enfin, pour notre dernier sprint, nous avons réalisé un huitième générateur, l'intégration d'un chatbot Rasa à une application JHipster. Nous nous sommes aussi rendu compte de la présence de bogues sur nos générateurs Stripe et Paypal. Nous avons donc passés plusieurs jours à résoudre les différents bogues et à faire des tests fonctionnels. Finalement, nous avons finis par la réalisation de deux applications de démonstration pour nos derniers générateurs.

# Récapitulatif





## Métriques logicielles

Le nombre de ligne de code écrit durant toute la durée du projet est difficilement estimable. En effet, la création de la majorité des générateurs est plus un travail de recherche et de modification de code existant que de créations. Par contre, le nombre de lignes de codes que nos générateurs généreront dans le futur sera potentiellement immense.

Les langages que nous avons principalement utilisé au cours de ce projet sont : HTML/CSS, TypeScript, Node.js, Java.

Le node.js nous a permis de coder les générateurs, tandis que les autres langages nous ont permis de coder les fonctionnalités à générer et les sample application.

A l'heure actuelle, nous possédons sur le Github de l'organisation 26 stars, réparti sur la plupart de nos générateurs et sur certaines des sample application.

Nous accumulons un total de {{ download }} sur l'ensemble de nos générateurs.

## Conclusion

Nous avons pour ce projet développé 8 générateurs fonctionnels pour les applications générées avec JHipster (Sauvegarde automatique de bases de données, Page d'administration des bases de données, Traduction automatique des entités et des messages, Gestion de quotas de création d'entités, Intégration d'Angular Material, Intégration de la plateforme de paiement Stripe, Intégration de la plateforme de paiement Paypal et Intégration d'un chatbot Rasa) ainsi qu'une application de démonstration pour chacun des générateurs. L'ensemble de nos générateurs sont disponibles sur la Marketplace de JHipster et sont téléchargés tous les jours par des personnes du monde entier.

Nous n'avons donc pas eu pour ce projet qu'un seul objectif sur lequel nous nous sommes concentrés tout le long ce qui nous a permis de découvrir différentes technologies (Stripe, Paypal, Rasa.com, etc.) mais aussi de nous former sur les technologies qu'utilise JHipster (Angular, React, Spring et Node.js).

Nous sommes donc grâce à ce projet montés en compétences sur différentes technologies et sur l'aspect génératif. Nous avons découvert différentes choses tout au long du projet et avons gagnés en notoriété auprès de la communauté open source JHipster. Cela à donc été pour nous un succès d'avoir été affectés sur ce projet.

# Glossaire

Docker : Logiciel libre permettant de lancer des applications dans un conteneur logiciel

UGA : Université Grenoble Alpes

JHipster : Générateur d'application libre et open source utilisé pour développer rapidement des applications Web modernes en utilisant Angular et le framework Spring

Stripe : Société proposant des services de paiement sécurisé par internet pour professionnel et particulier

Paypal : Service de paiement en ligne permettant de payer des achats, de recevoir des paiements, ou d'envoyer et de recevoir de l'argent.

Rasa : organisation open-source produisant des Intelligences Artificielles Assistants et Chatbots.

# Bibliographie

<https://www.jhipster.tech/>

<https://www.npmjs.com/>

<https://stripe.com/fr>

<https://developer.paypal.com/>

<https://cloud.google.com/translate/docs/>

<https://rasa.com/>