



Master 2 Informatique : Génie Logiciel

---

# Rapport de stage
# Cirrusware

---

*Professeur :* Laurent Réveillère
*Maitre de stage :* Vincent Jaulin

*Auteur :*
Alexandre Erard

9 août 2021

# 1 Remerciment

# 2 Avant propos

Ce rapport s'inscrit dans le cadre du stage de fin d'études d'informatique de l'Université de Bordeaux, plus précisement dans la spécialité Génie Logiciel.

Au cours de mon cursus, j'ai pu m'interesser à divers pan de l'informatique comme par exemple le développement d'application mobile native, l'intelligence artificielle, la gestion des bases de données et plus largement le Big Data, le devellopement web. Celui qui m'a le plus passioné est l'architecture logiciel, c'est à dire construire des applications scalable, maintenable tout en restant performante. Pour moi il prend tout sont sens dans le développement d'application web qui ont pour obligation d'optimiser leur performances, ainsi que leur consomation de ressource tout en étant capable d'ajouter facilement de nouvelles fonctionnalités afin de s'adapter aux nouveaux besoins.

J'ai donc recherché un stage dans ce domaine et l'entreprise Cirrusware m'ont offert l'opportunité de participer à leur projet afin que je développe mes compétences de développeur full stack.

# Table des matières

# 3 Contexte du stage

## 3.1 Cirrusware

Cirrusware est une PME informatique française spécialisée dans le développement d'applications web ayant pour objectif de faciliter les communications numériques au seins des entreprises ainsi que de facilter les gestions des relations entre tiers et contact.

Fonder en 2013, elle a su convaincre de grands groupes français tels que *MAÏSADOUR*, *OCEALIA*, *CARREFOUR*, ou encore *INTERCONTINENTAL - grand hotel de Bordeaux*. (Pour plus d'information, voir)

Aujourd'hui l'entreprise compte 13 employés dont 6 développeur 5 chefs de projet et 2 commerciaux.

Cirrusware développe deux type principaux de produits :

La plateforme Send-Up [1]

Le XRM.

## 3.2 La plateforme Send Up


Send-Up est le premier produit de Cirrusware et permet au client de créer, envoyé, analyser, et automatiser leur campagne de communication. Et cela au travers de divers outils présents sur la plateforme.

# 4 Outils et technologie utilisées

Lors de mon arrivé dans l'entreprise, j'ai du m'habitué à utiliser leurs outils de, gestion de projet, de communication, mais aussi de developement. Lors de nos projets d'études, là où on était une équipe de maximum 6 et on utilisait principalement des outils tels que Discord pour communiquer, VSCode pour developper, et GitHub Issue ou trello pour gérer le projet. Ici j'ai du m'habitué à utiliser des outils plus professionel et complet afin de pouvoir travailler sur des projets de plus grande taille. Voici la liste exhaustive de ces outils.

## 4.1 Outils

### 4.1.1 Monday

Monday est un outil de gestion de projet simple qui permet de créer des tableaux et y ajouter des tâches. Chaque tâche peut etre attribué à un quelqu'un, posséde un état (en cours, terminé, etc.) et une importance. De plus chaque tâche posséde un espace "discussion" qui permet de faire état de l'avancement ou encore de noter certaines informations importantes.

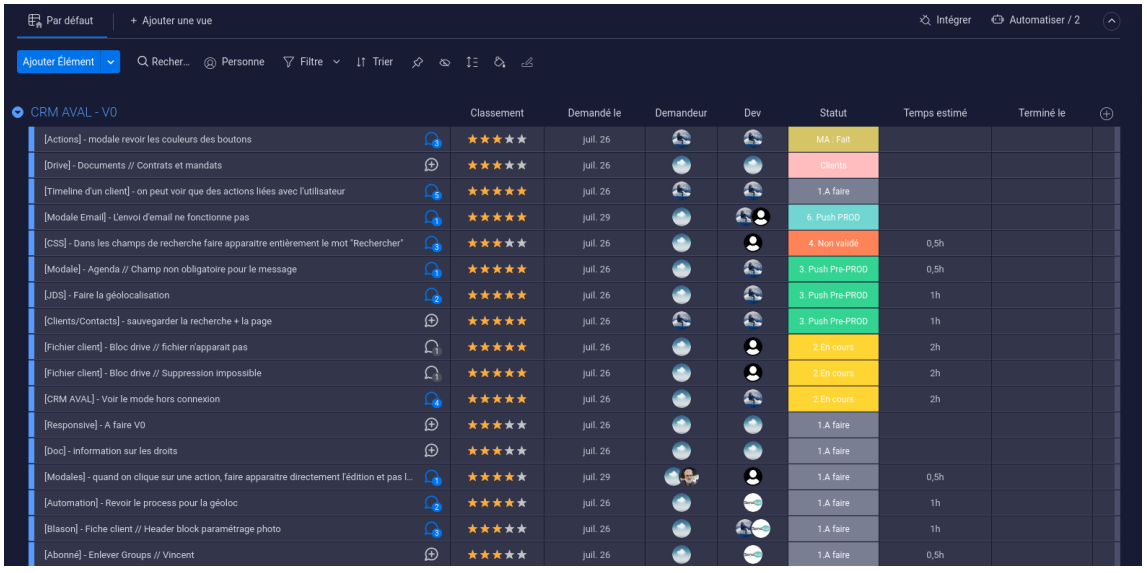


FIGURE 1 – Exemple de tableau Monday

### 4.1.2 Google Workspace

Afin de faciliter la communication au sein des équipes, l'entreprise utilise Google Workspace afin de produire de la documentation (Cahier des charges, etc...), GMail

et Chat afin d'échanger rapidement et Agenda pour planifier les rendez-vous. De plus, l'entreprise ayant des clients se trouvant dans toute la France, et tenant compte de la situation sanitaire actuelle l'applications Google Meet est utilisée comme application de visio conférence, pour les diverses réunions et rendez-vous projets .

### 4.1.3 Git et GitKraken

Git est un logiciel de gestion de version, il permet de créer des projets et de les versionner. Il permet également le partage du code ainsi que le développement sur plusieurs branches, afin de ne pas empiéter sur le travail des autres membres de l'équipe.

GitKraken lui est une interface graphique utilisant git. En effet git s'utilisant pricipalement par ligne de commande il est parfoit difficile de l'utiliser. GitKraken permet de faciliter l'utlisation de git et permet de tout faire en passant par l'interface (pull, merge, se déplacer entre les branche, etc..).

//TODO Screen GitKraken

De plus il permet une totale intégration des git flow. C'est à dire créer facilement une branche par nouvelle feature, et simplement merge avec la branche de développement lorsque celle ci est terminée. Les git flow permettent également une meilleure gestion des HotFix et des release mais je ne les ai pas utilisé lors de mon stage.


FIGURE 2 – Déroulement d'un git flow

### 4.1.4 PHPStorm

PHPStorm créé par JETBRAIN et l'un des IDE les plus complet pour développement web. A l'aide de ses nombreux pluggins il permet une totale prise en charge des framework comme Symphony et VueJS. De plus il permet de facilement refactorer du code ou encore de créer de nouveaux snippets afin de faciliter les phases de

développement. Malgré ses nombreux points positifs il est possible qu'il reste un peu dur à prendre en main au début mais par chance au cours de mes études supérieures j'ai été amené à utiliser InteliJ IDEA, ou encore Android studio qui sont deux IDE de JetBrain qui ont exactement la méme structure.

//TODO : Screen PHPStorm

### 4.1.5 Postman

En ce moment l'entreprise ouvre son API de plus en plus à ces clients afin qu'il puissent eux même intégré les différents services de la platforme. Et c'est pour cette raison que l'API doit devenir de plus en plus robuste. Dans cette optique nous utilisons l'application Postman afin de facilement créer nos requêtes ainsi que des test associé. L'outils permet de faire hériter des tests à des sous collections ce qui nous permet de gagner du temps lorsque certains test sont génériques comme par exemple un temps de réponse inférieure à 200ms ou encore un statut de réponse différent de 500.

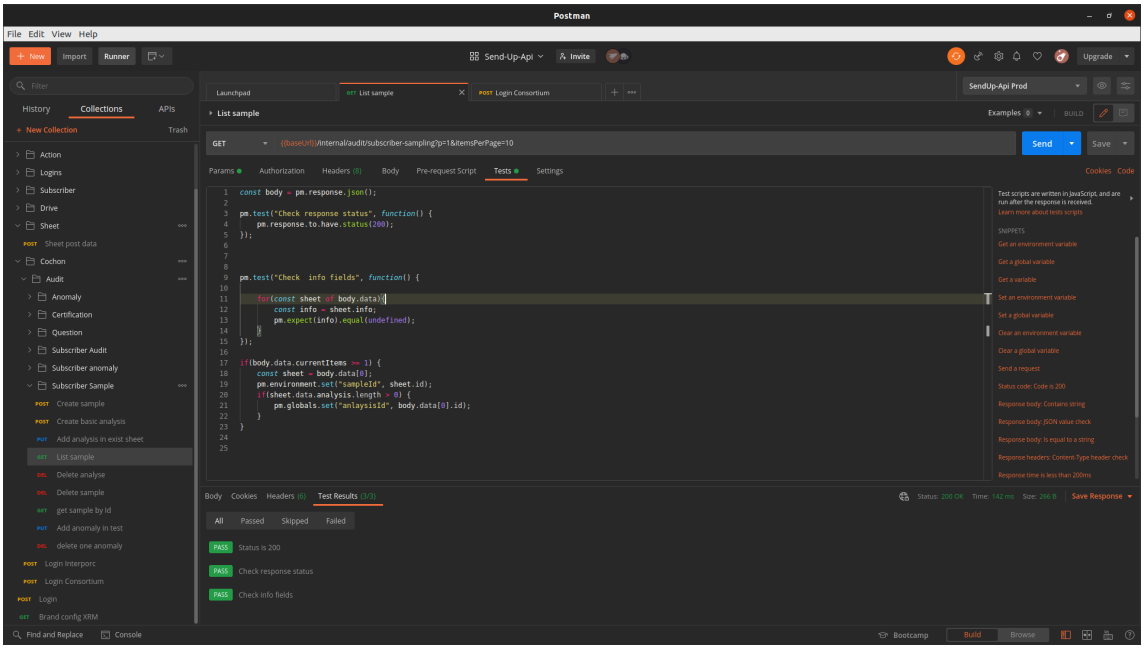


FIGURE 3 – Exemple de test Postman

De plus il intègre un *collection runner* qui permet d'agencer l'ordre des requêtes, ainsi que de les lancer $x$ fois, à un intervalle de temps $y$.

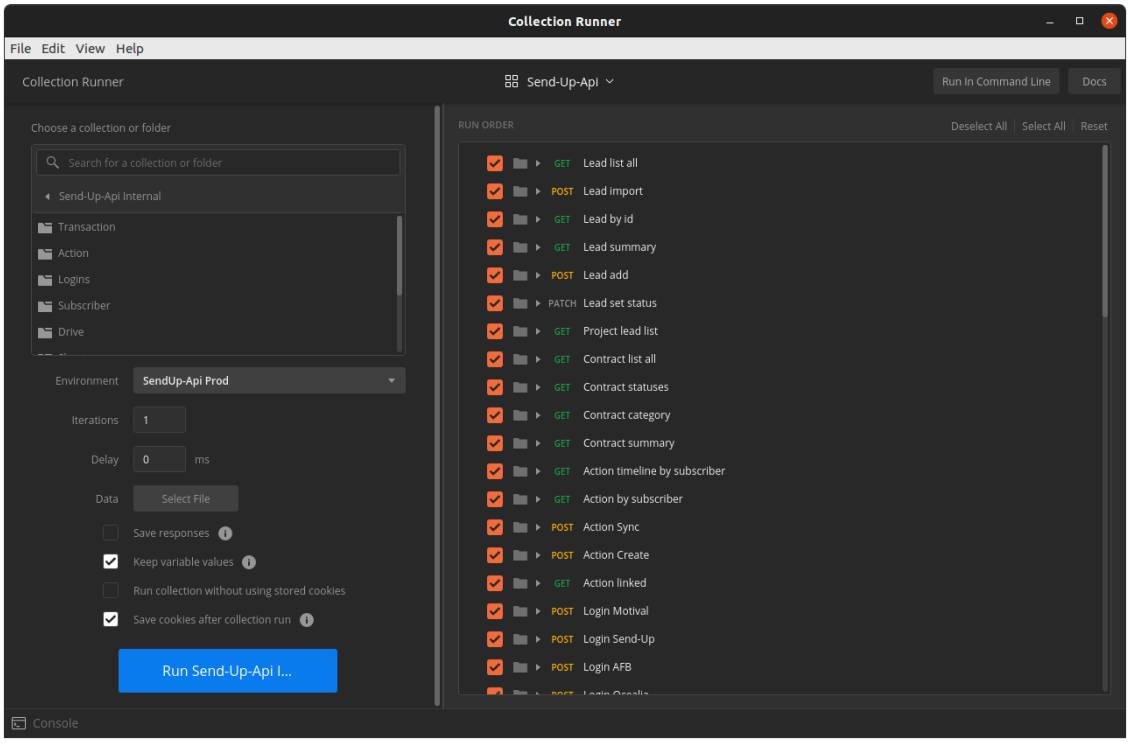


FIGURE 4 – Exemple de collection runner Postman

## 4.2 Technologies

Historiquement la platforme Send-Up à était developé sur une stack Symphony 2.8/Jquery/Bootstrap. Aujourd'hui l'entreprise est en pleine transition technologique. Son objectif est de se passer de Jquery et Bootstrap au profit des nouvelles technologies du web et plus particulièrement du framework VueJS. Cette transition reste un défi pour l'entreprise car certains outils du site comme le *studio digital* doivent garder ce relicat afin de garder une rétro-compatibilité mais aussi afin de facilité son usage auprés des clients. Voyons plus en détails les technologies choisies par l'entreprise à savoir Symfony et VueJS.

### 4.2.1 Symfony

Symfony [2] est un framework PHP français développée en 2005 par Fabien Potencier. Basé sur le pattern Modèle-Vue-Controller (MVC), il est actuellement en version 5.x.x Le gros avantage de Symfony sur les autres framework php du même type comme Laravel, ou CakePHP est sa documentation qui en plus d'être riches et également traduite en français. De plus, il possède une communauté trés importante et réactive. En effet a ce jour il existe plus de $X$ bundle sur la version 2.8. Et énormement de réponses sur les forums de developpeur. Il intègre également des outils qui sont extrement pratique dans la gestion des données comme par exemple **Doctrine** [3] un ORM (Object Relation Manager).

Un ORM permet de facilement gérer les données en transformant directement les résultats des requêtes en Entité php. Il intègre également un langage d'intérogation de la base de données nommé DQL. Ce langage rend plus permet de rendre plus lisible les requêtes et l'intégration de variable dans celle-ci. D'autre outils sont également présent comme l'injecteur de dépendance ou encore les outils de débbug disponnible en environement de developement.

### 4.2.2 VueJS

Pour remplacer la partie Jquery de Send-Up le framework VueJs [4] a était choisi par l'équipe. Il s'agit d'un framework open-source développer à l'origine par Evan You en 2013. Ce framework facilite la création d'interface graphique et de Single Page Application (SPA). Pour présenter rapidement le framework voici un exemple de composant basique.

Un fichier .vue est composé de 3 élements principaux.

- **template** : contient le code HTML de l'interface graphique.
- **script** : contient la logique et le comportement du composant.
- **style** : contient les styles CSS du composant. Le style peux être utilisé avec une props *scoped* afin de limiter les styles à l'intérieur du composant.

```vue
<template>
  <div>
    <h1>
      {{ helloworld }}
    </h1>
    <button @click="count">CLICK ME</button>
  </div>
</template>

<script>
export default {
  name: "HelloWorld",
  components: {},
  mixins: [],
  props: {
    msg: String,
  },
  data() {
    return {
      hello: "Hello",
      count: 0,
    };
  },
  computed: {
    helloworld() {
      return this.hello + " " + this.msg + "!";
    },
  },
  methods: {
    count: function () {
      this.count++;
    },
  },
  mounted() {
    console.log("HelloWorld component mounted");
  },
};
</script>

<style></style>
```

FIGURE 5 – Exemple de composant VueJS

La balise script est l'endroit ou on va déclarer le composant et tout ce dont il a besoin pour fonctionner. On y retrouve donc les sous composant qu'il utilise dans sont rendu, les paramètres (appelé *props*) que son parent peut lui passer, ici on attend un paramètres *msg* de type *String*. On retrouve également l'état du composant avec toutes les données utilisé,un compteur et la chaine de caractère *hello*, les methodes qui vont déterminer la logique, les propriété calculé du composant. Les propriété calculé sont appalable comme les variables indiqué dans data. Mais ne sont calculé qu'une seule fois à la création du composant puis lors de la modification de variable utilisé à l'interieur. De plus le framework possède une catégorie d'objet nommé *Mixins* dont on parelera plus en détails plus tard.

Voici le rendu de ce composant appeler avec $msg = Friends$ :

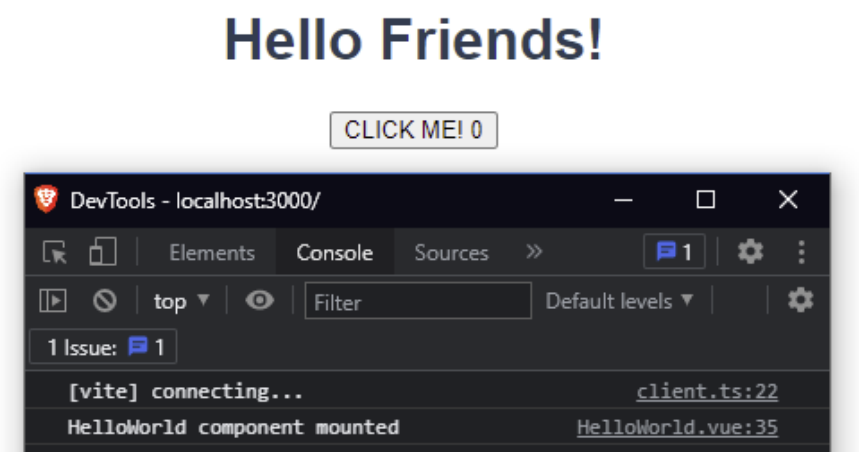


FIGURE 6 – Exemple de composant

On y retrouve également les divers évenement du cycle de vie auquelle on peut se brancher pour éffectuer certains traitement spécifique. Voici le schema du cyle ce vie d'un composant, pour plus de détails voir l'annexes 7

# 5 Missions réalisées

Au cours de mon stage j'ai pu intervenir sur différents projets allant d'un simple script permettant de dynamiser une page web d'un client à la reprise d'un projet complet. Ici nous allons détailler les deux principales missions de mon stage à savoir le développement de widget en vueJS pour le studio Digital et la finalisation des XRM du *Consortium du Jambon de Bayonne* et d'*Interporc Franche-Compté*.

## 5.1 Widget pour le studio Digital

Lors de mon arrivé dans l'entreprise, celle ci cherchait a valorisé ses formation en devenant Organisme de formation certifié **QUALIOPI**. QUALIOPI est une certification qui garanti des formations avec :

- des bons résultats
- des adaptations au besoins
- des formateurs qualifiés
- une bonne prise en compte des retours face au formations

Afin de correspondre à ces critères, nous avions besoin de construire un site web qui permettrait d'effectuer les questionnaires des différents modules de la formations, ainsi que le formulaire de satisfaction. Puis par la suite de pouvoir consulter les différentes statistiques des différents modules sous 3 vue différentes, formés, qui doit voir uniquement ses résultats, entreprise qui ne voit que les statistiques de ses employés formés, une visions globale qui permettra de voir toutes les statistiques des personnes formées.

La première étape réalisé afin de pouvoir réaliser ceci à était d'adapter l'enregistrement des formulaires du studio qui jusque là ne nous permet pas d'effectuer simplement des statistiques.

### 5.1.1 Serializations des données des formulaires

Afin de récolter les données des formulaire, le studio digital passe par un serializer qui va enregistrer dans un fichier json les valeurs des différents champs associé à leur id. Ce système présente plusieurs avantage comme la facilité et la rapidité d'implémentation. Maleureusement ceci ne nous permet pas de savoir si une réponse est, dans notre cas, correcte ou non. Ou encore pouvoir associé le label d'une question à son résultat.

Pour atteindre le résultat souhaité j'ai du implémenter un nouveaux widget pour le studio digital en VueJS. Ce widget est un simple bouton de correction qui va devoir récupérer tout le contenu nécéssaire du formulaire et les stocker dans des champs cachés afin de se servir du serializer. Le reste du formulaire étant coder en

JQuery/html, la seule solution était donc de parcourir le formulaire et de récupérer les information dans le DOM afin de réccupérer les informations nécéssaire. A savoir les information permettant le calcule de la note, diverses meta-données par rapport au question comme le label, l'id du champ concerné,le résultat obtenu a la question, etc..

Dès que ces informations fut récupérer il ne rester plus qu'a établir les différentes requêtes nécéssaire à la récupération des données en fonction du périmetre défini (formé, entreprise, admin)

### 5.1.2 Compréhension de l'API et implémentation des requêtes

Dans l'actuel implémentation du serveur Send-Up, la partie gérent les *Recordset* - Table MySQL comportant un champ info destiné à stocker du JSON - est gérer par une partie de l'API dont les routes pointant vers cette partie sont générer automatiquement à partire du nom de la méthode défini dans le controlleur.Exemple la méthode postRecordset est appelée lors d'une requête POST sur /Recordset. De plus l'une des complications a été de comprendre le format des données à envoyer dans la requête. En cherchant plus profondément dans le code je me suis rendu compte que seuls les **FormData** était accepter pour la requête existante. Une fois le fonctionnment compris il suffisait d'implémenter une requête à la base de donées prenant en compte la portée dans laquel on se trouvé afin de renvoier uniquement les donées nécéssaire.

### 5.1.3 Création des widgets

Une fois les données récupérer et les requêtes créer il fallait que je créer 5 widgets différents qui permeté de :

- visualiser les résultats d'une personnes formée pour chaque tentatives.
- visualiser les réponses avec correction d'une personne formée.
- afficher les statistiques globale de la formations c'est a dire le nombre de bonne réponse par rapport au nombre de personne.
- afficher les statisques globale pour le formulaire de satisfaction client.
- afficher les réponses d'une personnes formé au questionnaires de satisfaction.

Après avoir développer certains d'entre eux, il s'avèra que certains traitement se répeter ou se resembler fortement. Or VueJS integre ce qui appèle les *Mixins*, ces éléments possède les mêmes caractéristique dans leur balise script qu'un composant vueJS classique. Mais ce dernier à la possibilité d'être intégré à l'intérieurs de composant (ou autre Mixins). Chaque composant hérite donc des donnés, méthodes et autres propriétés calculé. Ceci empêche donc la redondance et facilite la maintenabilité. C'est donc pour cela, que j'ai créer un mixins spéciale afin de facilité la lisibilité et la maintenance du code.

## 5.2 Reprise d'un projet XRM

Suite au départ d'un développeur de l'équipe. On m'a placer sur la reprise de son projet qui consite a creer un XRM permetant tous les points de contrôle concernant le *Consortium du Jambon de Bayonne* et *Interporc Franche-Compté*. Un XRM

# 6 Conclusion

# 7 Anexes

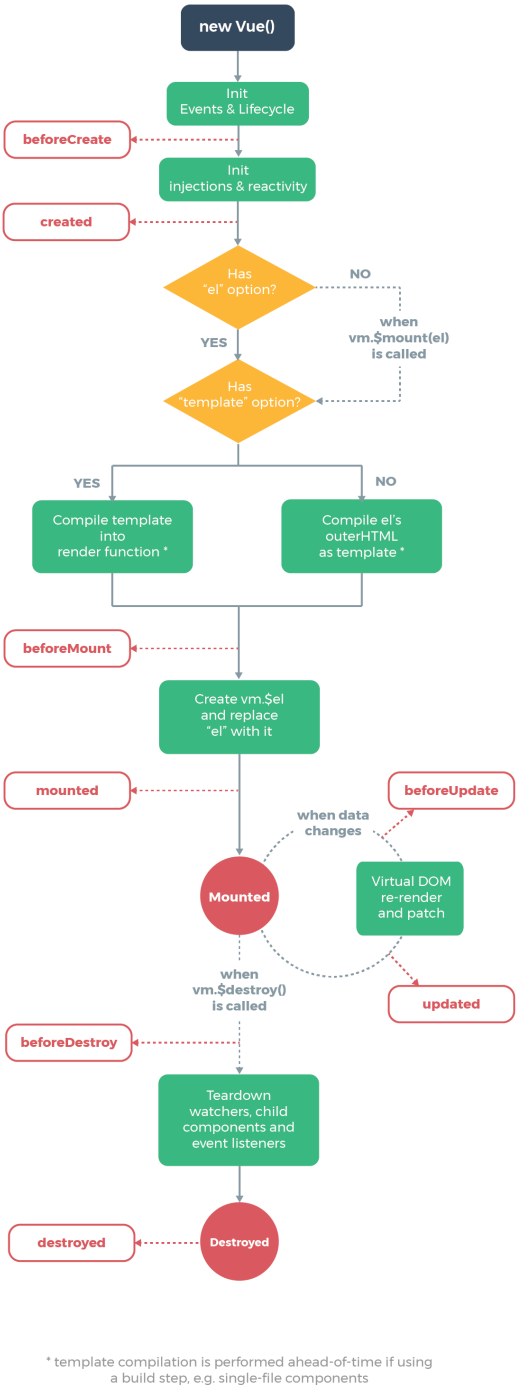


FIGURE 7 – Schéma du cycle de vie - d'un composant vue [5]

# Références

[1] Cirrusware. Send-up. http://www.send-up.net/.

[2] Symfony. Symfony. http://symfony.com/doc/current/index.html.

[3] Doctrine Project. Doctrine. https://www.doctrine-project.org/projects/doctrine-orm/en/2.9/index.html.

[4] Vue. Vue. https://vuejs.org/.

[5] Vue. The vue instance. https://vuejs.org/v2/guide/instance.html.

[6] Red Hat. What is a rest api ? https://www.redhat.com/fr/topics/api/what-is-a-rest-api.