Ensemble Scolaire La Salle
Campus Godefroy de Bouillon – Clermont-Ferrand

Rapport de stage

Étudiant : Gabin JUILLARD
Date : du 08/01/2024 au 16/02/2024
Entreprise : be ys Cloud
Tuteur : Florian FORESTIER

SOMMAIRE

PRESENTATION DE L'ENTREPRISE

A ces débuts, be-ys est une start-up. Créée en 2000, c'est un éditeur de solutions pour l'assurance santé. Puis, elle devient un leader national sur le marché de la gestion des prestations de santé. Dans les années 2010, be-ys entame une diversification de son activité, et devient notamment un mandataire de confiance numérique universelle avec 2400 collaborateurs répartis sur quatre continents.

be-ys[1] est spécialisée dans l'identité numérique, les flux d'informations personnelles et le traitement des données sensibles. Elle apporte des solutions numériques pour protéger, gérer et optimiser le patrimoine de données des sociétés. Elles sont basées sur des concepts de confidentialité et de sécurité tout en garantissant le respect de toutes les exigences légales et réglementaires.

Sa mission est d'apporter la meilleure solution pour améliorer les performances financières des entreprises en facilitant leur gouvernance et leur gestion.

be-ys est implanté dans plusieurs villes en France, notamment à Clermont-Ferrand (son siège social), mais également à Paris, Toulouse et Annecy. Par ailleurs, l'entreprise est présente dans plusieurs pays, notamment le Luxembourg, la Tunisie, Malte, la Roumanie, Madagascar et la Bulgarie.
La structure du groupe est basée sur le concept d'entreprise indépendantes, disposant chacune de son nom, son identité et sa gestion qui lui est propre.

Figure 1: Liste des entreprises du groupe be-ys

[1]Le nom de l'entreprise ne prend pas de majuscule, y compris au début de la phrase.

Dans le cadre de mes études, j'ai eu l'occasion de réaliser un stage en entreprise afin de parfaire mes connaissances en informatique grâce à une application pratique et directe des enseignements que j'ai pu avoir au cours de mes deux années de BTS. Ce stage doit également me permettre de confirmer ma voie professionnelle et mon envie de travailler dans ce domaine.

J'ai donc postulé auprès de la société be ys Cloud, spécialisée dans l'hébergement et la fourniture d'infrastructure virtuelles. Cette société a été fondée en 2020, et compte 24 collaborateurs, tous employés sur le site de Clermont-Ferrand. Cette société est dirigée par M. Christophe PRUGNAUD, que je remercie pour sa disponibilité et pour m'avoir offert l'opportunité d'intégrer son entreprise.

Figure 2: Locaux de l'entreprise à Clermont-Ferrand

Il est à noter qu'en supplément des locaux, be-ys Cloud opère son propre datacenter, hautement sécurisé, nommé La Citadelle. J'ai eu l'occasion, au cours de mon stage, de pouvoir le visiter et découvrir les sujets liés au monde de l'infrastructure physique.

CONTEXTE DU STAGE

Locaux et équipe

Comme expliqué précédemment, mon stage s'est déroulé dans les locaux de be ys Cloud, durant six semaines. Plus précisément, j'ai été projeté dans le bureau A106, avec trois autres développeurs, dont mon tuteur, qui m'ont accompagné tout au long du stage.

Cette équipe est en charge de l'ensemble des développements liés à l'activité principale de la société : réalisation des sites et des interfaces pour les utilisateurs, création et conception des interfaces de programmation destinés aux autres développeurs et utilisateurs techniques de leurs solutions, etc.

Mission confiée

La mission qui m'a été confiée durant ce stage a été la réalisation de nouvelles fonctionnalités sur la page de statut servant à informer l'ensemble des clients de l'état des services mis à disposition par be ys Cloud.

Cette page de statut était partiellement construite, mais n'était pas suffisamment développée pour être considérée comme prête à aller en production. Mon stage a permis de corriger cela, en apportant les fonctionnalités attendues afin de rendre son usage possible.

La page de statut permet aux utilisateurs de consulter un ensemble d'états de services. Ces états sont calculés à partir d'un outil de supervision, nommé Shinken, qui récupère des informations sur l'ensemble du Système d'Information de be ys Cloud. Par ailleurs, il est possible pour un administrateur d'ajouter un incident manuellement, afin de pouvoir fournir du contexte à l'utilisateur sur une mise à jour, un incident de production, ou autre. Ce projet permet donc de sensiblement améliorer l'expérience utilisateur, et la communication entre l'entreprise et ses clients.

Parmi les fonctionnalités à ajouter, les plus importantes ont été :

- Afficher le détail des anciens incidents ;
- Rendre dynamique la configuration de l'outil, en utilisant une base de données au lieu d'un fichier de configuration ;
- Créer une interface d'administration permettant d'éditer la configuration de manière simple et intuitive.

Étude de l'existant

A mon arrivée, il existe déjà une page de statut, qui n'a jamais été publiée. Cette dernière est basée sur un modèle client-serveur, interagissant par le biais d'API[2]. Le client est conçu avec un framework nommé VueJS. Ce framework permet aux développeurs de construire des pages dynamiques, et des sites complets sans temps de chargement (SPA[3]).

Le serveur est construit en Golang, langage de programmation impératif et fortement typé. Ce langage est choisi chez be ys Cloud pour sa légèreté, et la puissance de son écosystème. Ce dernier se connecte à une base de données relationnelle, MariaDB, pour stocker ses données. Cependant, l'ensemble de la configuration est défini dans un fichier de configuration JSON.

Afin d'éviter d'être redondant avec d'autres systèmes, ce serveur se base sur d'autres services pour récupérer les informations à afficher à l'utilisateur :

- Shinken, qui est un outil de supervision, permet de récupérer l'état de chacun des services. En résumé, chaque service est surveillé par une ou plusieurs **sondes**. Ces sondes ont chacune un **état**, qui indique le statut actuel du service. Par exemple, une sonde peut vérifier l'usage actuelle de la mémoire ou du processeur d'un serveur, ou encore si un serveur Web répond.
- GitLab, qui est un outil de gestion du code, sert quant à lui de base de vérité pour les incidents. En effet, à chaque incident, un ticket (**issue**) est ouvert sur GitLab, pour permettre aux équipes de se coordonner entre elles dans le but de résoudre le souci rencontré. L'idée est donc de récupérer ces informations, et de les afficher aux clients.

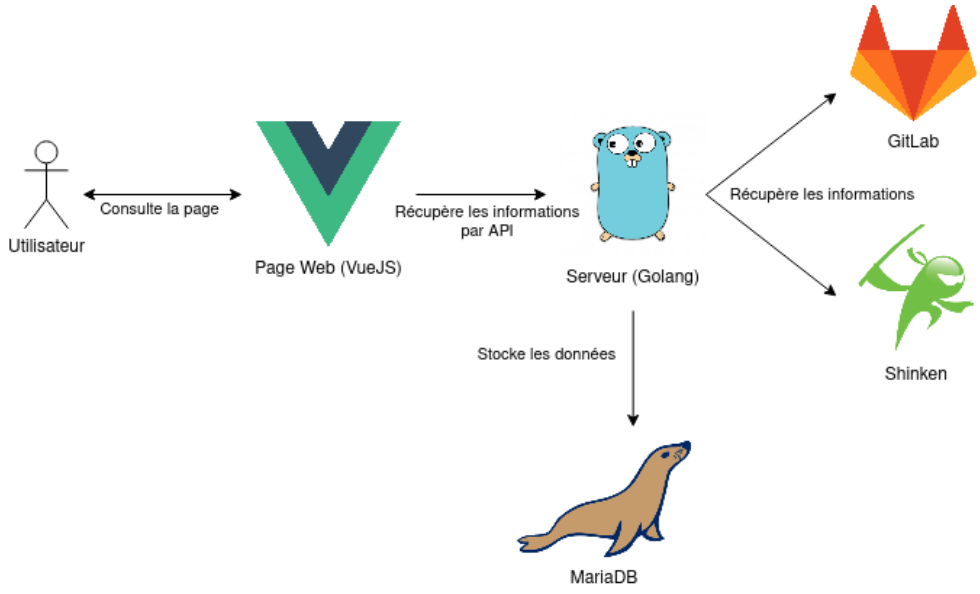

Figure 3: Schéma résumant l'interaction entre les services

[2]API : Application Programming Interface
[3]SPA : Single Page Application

TRAVAIL REALISE

Affichage des détails des anciens incidents

A mon arrivée sur le projet, la page de statut ne permettait pas d'avoir le détail des incidents passés. Or, un utilisateur ayant rencontré un souci sur ses produits souhaite avoir accès à une rétrospective sur les évènements qui se sont déroulés. L'idée est donc de rajouter un historique des anciens incidents. Cette modification m'a également permis de découvrir davantage VueJS et ses concepts.

Pour ce faire, j'ai d'abord créé une zone en VueJs, où sera affiché le contexte complet de l'incident passé, avec la date et l'heure de celui-ci, ainsi que son niveau de gravité.

J'ai ensuite créé un attribut de type booléen aux incidents passés, pour que l'utilisateur puisse obtenir le résultat en cliquant sur un bouton précédemment créé et stylisé.

Et pour finir j'ai relié la zone d'affichage à l'attribut dans une condition.

Figure 4: Page de statut affichant le détail d'un ancien incident

```
<button class="btn-incidentdetails" v-on:click="i.displayDetails = !i.displayDetails">
  <template v-if="i.displayDetails">
    <span class="fas fa-minus mr-2"></span>{{ $t("status.hide_details") }}
  </template>
  <template v-else>
    <span class="fas fa-plus mr-2"></span>{{ $t("status.show_details") }}
  </template>
</button>
```

Figure 5: Partie du code gérant le bouton d'action

Par mesure de lisibilité, le reste du code est omis.

Déplacement de la configuration vers la base de données

Comme mentionné précédemment, l'idée de déplacer la configuration d'un fichier de stockage classique vers la base de données est de permettre de gérer dynamiquement (autrement dit, sans avoir à redéployer le serveur) les paramètres pouvant régulièrement changer. Il est à noter que certaines parties de la configuration ne peuvent pas être mise en base de données, notamment les paramètres de connexion à ladite base.

J'ai d'abord créé et défini les tables SQL, ainsi que leurs structures associées ; puis j'ai ajouté un pilote MariaDB à la base de code Go déjà existante pour pouvoir communiquer entre le serveur et la base de données.

Par la suite, j'ai dû créer une API pour manipuler les données via des URL. Pour cela, j'ai créé une API de type REST[4], permettant d'effectuer des actions de création, mise à jour et suppression des données (CRUD). L'API est conçue en respectant le modèle MVC (Modèle – Vue – Contrôleur). Dans le détail :

- Un premier fichier va définir la liste des URLs, leurs paramètres et leur verbe HTTP (main.go) ;
- Un second fichier, (le **contrôleur**) va permettre de traiter la requête en elle-même, et notamment de vérifier que les données envoyées par le client sont lisibles ;
- Puis, le **service** va effectuer les actions à proprement parlé (récupération d'un élément dans la base, chiffrement/déchiffrement de données, etc) ;
- Un fichier **repository** va effectuer les actions en base ;
- Et enfin, un fichier de **modèle** va définir chaque type de données que nous aurons.

Cette structure permet d'isoler chaque partie du code et de séparer les responsabilités, ce qui permet de maintenir plus facilement le code.

Enfin, ces URLs vont être sécurisées grâce à KeyCloak, l'outil de gestion d'identité utilisé chez be ys Cloud.

L'ensemble de ce travail est documenté grâce à un outil nommé OpenAPI, qui permet de créer une interface graphique listant l'intégralité des URLs exposés par l'API, ainsi que les paramètres requis à chaque appel.

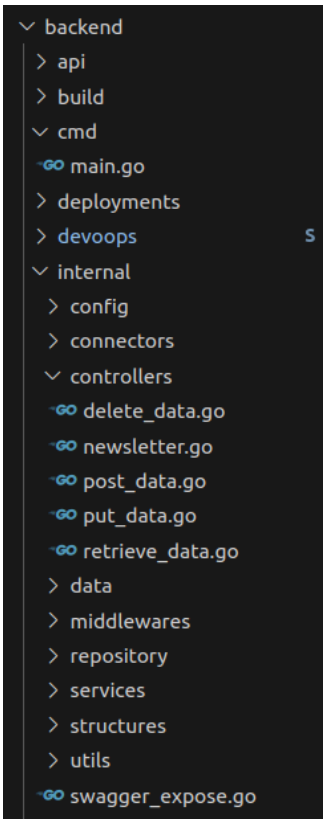

Figure 6: Structure du serveur

[4]REST : Representational State Transfer. Il s'agit d'une architecture d'API permettant d'exploiter les caractéristiques de HTTP.

Création de l'interface d'administration

Afin d'implémenter l'interface administrateur, j'ai créé un template en VueJS afin de présenter un titre et un tableau contenant les données des groupes, des sondes, et des sondes Shinken. Ces données sont organisées dans des composants distincts pour une meilleure modularité et une gestion plus efficace.

Un composant est un ensemble de code HTML et Javascript, qui peut être appelé depuis un autre composant.

Le tableau lui est construit avec l'élément HTML <table> permettant une présentation structurée des informations telles que leurs ID et noms respectifs. De plus, ce tableau offre des fonctionnalités interactives permettant à l'utilisateur de visualiser les détails des sondes appartenant à un groupe spécifique, de les mettre à jour ou de les supprimer selon ses besoins.

Dans le détail, chaque ligne du tableau représente un groupe, et chaque colonne affiche l'ID du groupe, son nom, ainsi que des options pour interagir avec celui-ci. Les options incluent la possibilité de visualiser les détails des sondes associées au groupe, de les mettre à jour, ou de les supprimer. Ces actions sont déclenchées par des événements click sur des icônes spécifiques qui affichent.

Le tableau est conditionnellement affiché en fonction du chargement des données. Une fois les données chargées, le tableau est affiché, permettant ainsi à l'utilisateur d'interagir avec les différentes fonctionnalités proposées.

Pour faciliter la communication entre le composant parent et le composant enfant, j'ai utilisé des props. Les props sont utilisées pour transmettre des données d'un composant parent à un composant enfant de manière unidirectionnelle. Par exemple, j'ai transmis l'identifiant d'un groupe en tant que prop, ce qui permet au composant enfant d'afficher les sondes associées à ce groupe spécifique. Cette utilisation des props rend le code plus modulaire et permet une meilleure organisation des données au sein de l'application VueJS.

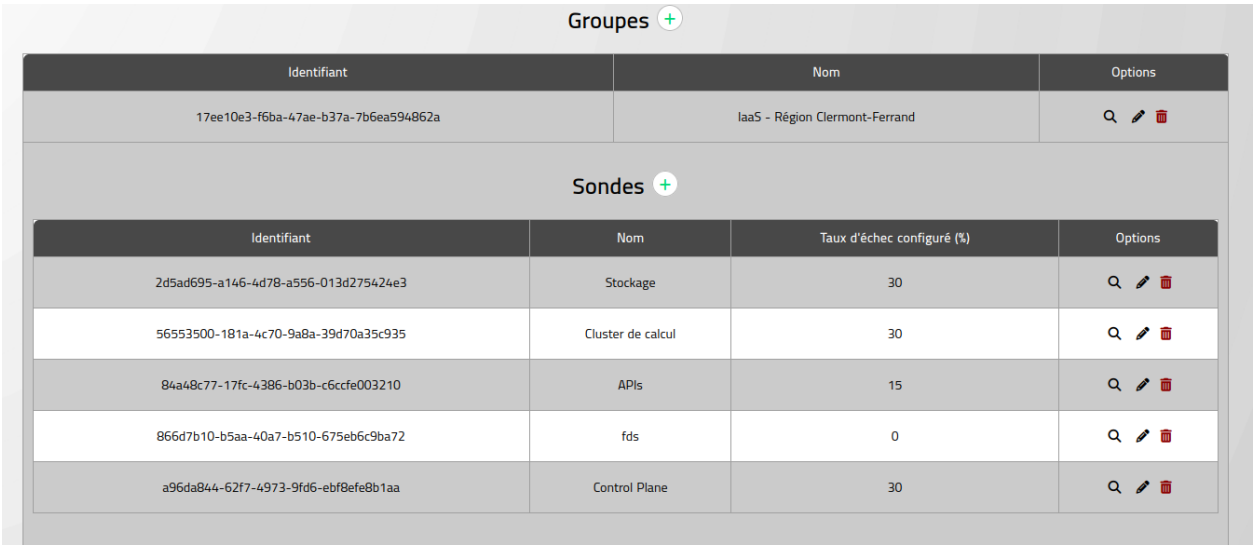

Figure 7: Tableau de gestion d'administration

CONCLUSION

Ce stage m'a ouvert les yeux sur certains aspects du métier. A la base, j'avais ma vision du travail par les éléments que je connaissais de l'enseignement scolaire mais en approfondissant certains points j'ai découvert d'autres choses.

D'un point de vue projet, l'ensemble des modifications que j'ai apportées ont pu être déployées après avoir été vérifiées par l'équipe. Le travail rendu est donc accessible sur **https://status.be-ys.cloud/**.