



RAPPORT DE STAGE

BTS SIO - 1er Année

SUMBO Daniel
KOLUS

Table des matières

| | |
|--|----|
| Remerciements..... | 1 |
| Introduction | 2 |
| Présentation de l'entreprise..... | 3 |
| 1 - Contexte du Stage | 4 |
| 1.1 - Le produit * | 4 |
| 1.2 - Outils..... | 5 |
| 1.3 - Missions | 5 |
| 2 - Déroulement et Gestion Technique pendant le stage | 6 |
| 2.1 - Familiarisation avec le code, le volume de pages de codes et les imbrications du code .. | 6 |
| 2.2 - Intégration du tri dans un tableau..... | 7 |
| 2.3 - Mise à jour de test playwright | 11 |
| 3 - Correction d'un bug dans le tri du tableau | 12 |
| 4 - Intégration de l'API DIMO – Maint..... | 13 |
| Conclusion | 14 |
| État du projet à la fin du stage..... | 14 |
| Mon ressenti, ma progression, mes difficultés | 14 |
| Annexes | 15 |
| Equipe Kolus | 15 |
| La plateforme Kolus | 15 |
| Atomic Disgn Partern | 16 |
| Volume de code..... | 17 |
| La page d'un Kol | 18 |
| Modal DIMO MAINT | 18 |

Remerciements

Avant toute chose j'aimerais remercier toute l'équipe Kolus qui m'a accueilli à bras ouverts et plus particulièrement mon maître de stage Max Barbet qui a su m'accompagner tout au long du stage.

Je tiens également à remercier et à témoigner ma reconnaissance à mes professeurs de première année de BTS SIO à l'institution des Chartreux pour la formation enrichissante et pleine d'intérêts qu'ils m'ont fait vivre.

Je remercie aussi les personnes qui ont, de près ou de loin, contribué à la réalisation de ce stage.

Introduction

Tout d'abord je m'appelle Daniel Sumbo, j'ai 21 ans et je viens d'effectuer une première année de BTS SIO (services informatiques aux organisations), avant celle-ci j'ai effectué deux années en université, une L1 de mathématiques et informatique puis une autre de mathématiques seulement.

Ce qui m'a motivé à ne pas poursuivre en L2 et de commencer un BTS est le manque d'encadrement à l'université et surtout la distance entre chez moi et l'université (Lyon – Saint-Etienne).

J'ai donc choisi une formation qui me permet d'être aux prés de mes proches tout en étant encadré par des professeurs plus impliqués tout en gardant la possibilité d'avoir un diplôme d'ingénieur.

Le BTS SIO comporte deux options et j'ai choisi l'option SLAM (solutions

logicielles et applications métiers) qui est axé sur le développement informatique car je voulais garder la possibilité de poursuivre mes études en Corée du Sud et je pense que j'aurais plus de chance avec une spécialisation dans le développement ou l'anglais est roi.

Enfin, l'entreprise dans laquelle j'ai effectué mon stage propose un SaaS (Software as a Service) et le produit est au centre du projet, celui-ci était développé en React une librairie JavaScript innovante que j'ai commencé à apprendre en cours d'année donc ça été une bonne occasion pour approfondir mon apprentissage et tester celui-ci dans des conditions réelles.

Présentation de l'entreprise

Du 3 juin au 5 juillet 2024, j'ai effectué un stage au sein de l'entreprise Kolus). Au cours de ce stage avec l'équipe technique, j'ai pu m'intéresser à la structure et au développement d'applications web en React, en participant à la création de composants interactifs, à l'optimisation du code et à l'intégration d'API pour améliorer l'expérience utilisateur.

L'entreprise Kolus se situe dans le 9ème arrondissement de Lyon dans les anciennes usines du chocolatier Voisin. Elle apporte une solution innovante pour améliorer le service après-vente des entreprises industrielles.



L'entreprise proposant un SaaS nous ne parlons pas de chiffre d'affaire mais de revenu annuel récurrent (ARR) celui-ci étant d'environ 242 000€. Le pôle technique développant le produit il a donc une place capitale dans l'entreprise au même cas que les pôles marketing et commercial.

Mon maître de stage Max Barbet y agit en tant que Lead Developer, j'ai donc pu apprendre dans d'excellentes conditions et ai bénéficié d'un soutien de qualité.

L'entreprise s'organise de cette forme* :

- CEO : William Kelbert
 - Directeur Commercial : Maxime Guertin
 - Account Executive : Louis Moreira
 - Account Executive : Valentin Allirand
 - Business Development Representative : Carla Faure
 - Sales Development Representative : Edouard Laborde
 - Lead Développeur : Max Barbet
 - Développeur Web Front (stagiaire) : **Daniel Sumbo**
 - Développeur Web Full Stack (arrivé après le stage) : Joan Magro
 - 2 Développeur Web Full Stack en Freelance depuis le Vietnam
 - CMO : Samar Manaï

1 - Contexte du Stage

1.1 - Le produit *

Kolus assiste les industriels lors de rendez-vous SAV à distance en leur donnant la possibilité de transformer un appel téléphonique en visioconférence et donc qu'un expert puisse correctement aiguiller son client. Elle offre également des outils pour mieux guider le client pendant le live et propose un suivi efficace grâce à un système de ticketing.

La plateforme a pour but d'éviter les déplacements inutiles, garder un historique des interventions/échanges effectuées et pour les déplacements inévitables d'avoir une première prise d'informations à donner à l'intervenant.

1.2 - Outils

La plateforme a été codée avec React ~2020 en utilisant l'atomic design patern*.

Pour communiquer dans l'entreprise, nous avons des canaux Slack avec toute l'équipe. En plus de ceci, l'équipe technique avait un KanBan sur Notion.

La base de données était gérée via PostgreSQL, soit en local, soit sur une base de données commune en ligne, en utilisant les configurations définies dans le fichier .env. pour le développement.

J'ai utilisé Visual Studio Code et Github pour enregistrer mes modifications avant de les soumettre à mon maître de stage, cependant l'entreprise utilise Gitlab. Je n'ai pas pu utiliser ce dernier car je n'allais pas directement push mes modifications dans le projet, elles devaient d'abord être validées par le maître de stage puis push par lui-même.

Slack est un outil qui permet l'intégration de nombreuses API et d'optimiser les échanges d'informations entre les différents pôles (par exemple un bot qui annonce chaque vente réussie).

Notion est l'application de prise de note par excellence et qui dicte la tendance dans ce milieu et offre le plus de fonctionnalités avec un rapport qualité-prix très intéressant.

PostgreSQL est extensible et polyvalent, ce qui lui permet de prendre rapidement en charge tout un panel de cas d'utilisation spécialisés avec un écosystème d'extensions puissant, qui couvre des éléments allant des types de données chronologiques aux analyses géospatiales.

Gitlab a été un choix des anciens développeurs et l'outil marche très bien donc il n'y a pas eu de raison de le changer.

Pour ma part **Visual Studio Code** et **Github** sont des logiciels que j'utilise depuis mes débuts dans l'informatique, ils ont été choisis car c'étaient ceux présents lors des tutos et ils ne m'ont toujours pas donné de raison de les changer.

1.3 - Missions

J'ai occupé le poste de stagiaire Développeur Web Front React.

Mes missions ont été :

- Familiarisation avec le code, le volume de pages de codes et les imbrications du code du au atomic design patern
- Intégration du tri dans un tableau
- Mise à jour de test playwright
- Correction d'un bug dans le tri du tableau
- Intégration de l'API DIMO – Maint

2 - Déroulement et Gestion Technique pendant le stage

Pour commencer, une semaine type chez Kolus en faisant partie de l'équipe technique est organisée autour de deux réunions : un meeting technique le lundi matin sans horaire précis et un meeting le mardi à 9h avec toute l'équipe présente sur place (parfois avec le CEO en visioconférence) pour discuter de ce qui a été effectué la semaine passée et de ce qu'il y a à faire pour la semaine à venir.

2.1 - Familiarisation avec le code, le volume de pages de codes et les imbrications du code

La première journée a été consacrée à l'installation du projet sur ma machine ainsi qu'à l'installation de tous les outils utilisés par l'équipe. Pendant cette tâche, j'ai rencontré mes premières difficultés. D'abord, lors de l'installation du code, celui-ci ayant une base ancienne, j'ai dû installer des plug-ins de versions précédentes. Un autre problème est survenu car, étant récemment passé sur Linux, j'ai dû changer les permissions sur certains fichiers, tout en étant un peu stressé de ne pas pouvoir commencer rapidement et de buter sur une tâche d'apparence facile comme l'installation du projet.

Mon maître de stage a su m'aiguiller sans me donner la solution et j'ai finalement réussi à compiler le code sur ma machine.

Après tout ça j'ai eu ma première mission mais je n'ai pas su par quel bout commencer, le code était tellement imposant* par rapport à tous les projets auxquels j'ai pu participer que je n'ai pas su où chercher ce dont j'avais besoin mais avec le temps j'ai su maîtriser ce code.

2.2 - Intégration du tri dans un tableau

Kolus propose un service nommé le PASS (Parcours d'assistance), qui donne la possibilité au client de créer un chatbot qui dirige les SAV directement vers le bon technicien.

Le problème initial est que dans le tableau des PASS les PASS s'affichent du moins récent au plus récent et le DC voulait l'inverse.

| TITRE | NOMBRE DE KOLS | ÉLÉMENTS PUBLIÉS | DATES | STATUTS | LANGUES | PARTAGER | |
|--------------------|----------------|------------------|---|------------|----------|----------|----------|
| Parcours DÉMO | 198 | 26 | Publié 23/08/2024 Modifié 23/08/2024 | EN LIGNE | Français | Partager | 👁️ ✎️ 🗑️ |
| Test FASSI | 2 | 12 | Publié 07/07/2023 Modifié 10/07/2024 | EN LIGNE | Français | Partager | 👁️ ✎️ 🗑️ |
| Aucun titre publié | 0 | 0 | Modifié 02/08/2024 | HORS LIGNE | Français | Partager | 👁️ ✎️ 🗑️ |
| Aucun titre publié | 0 | 0 | Modifié 12/07/2023 | HORS LIGNE | Français | Partager | 👁️ ✎️ 🗑️ |
| Aucun titre publié | 0 | 0 | Modifié 13/07/2023 | HORS LIGNE | Français | Partager | 👁️ ✎️ 🗑️ |
| Aucun titre publié | 0 | 0 | Modifié 17/07/2023 | HORS LIGNE | Français | Partager | 👁️ ✎️ 🗑️ |
| Aucun titre publié | 0 | 0 | Modifié 17/07/2023 | HORS LIGNE | Français | Partager | 👁️ ✎️ 🗑️ |
| Aucun titre publié | 0 | 0 | Modifié 18/07/2023 | HORS LIGNE | Français | Partager | 👁️ ✎️ 🗑️ |
| Aucun titre publié | 0 | 0 | Modifié 23/08/2023 | HORS LIGNE | Français | Partager | 👁️ ✎️ 🗑️ |
| TEST WILL | 14 | 5 | Publié 26/06/2024 Modifié 10/07/2024 | EN LIGNE | Français | Partager | 👁️ ✎️ 🗑️ |

Le plan initial était de trouver ces PASS et faire $x(-1)$ pour inverser leurs ordre, mais ces PASS faisaient partie d'un objet qui contient plusieurs informations et le tableau y accède grâce à la caractéristique globale ressources.

```
resource="passes"
```

L'ordre des PASS n'était donc pas modifiable juste avant qu'ils soient donnés au tableau, et il fallait donc trouver une solution.

Pendant la recherche de la solution, j'ai trouvé que les anciens développeurs avaient déjà pensé à intégrer du tri dans les tableaux mais cette option était cachée/non utilisée.

```
<TableRow>
  {columns.map((column) => {
    const isSortable = sortableColumns?.find((it) => it.name === column)?.isSortable;

    |

    return (
      <TableCell key={column}>
        {isSortable ? (
          <TableSortLabel
            active={sortItem?.name === column}
            direction={sortItem?.sortType}
            onClick={handleSort && handleSort(column)}
          >
            {t(`panel.${resource}.table.titles.${column}`)}
          </TableSortLabel>
        ) : (
          t(`panel.${resource}.table.titles.${column}`)
        )}
      </TableCell>
    )
  })
}</TableRow>
```

Avec cet objet isSortable vient s'ajouter une variable handleSort qui m'as facilité à réaliser ma mission en quelques étapes, la première a été d'ajouter isSortable et handleSort au code.

```
<Table
  key="pass"
  ref={tableRef}
  load={loadTablePasses}
  sync={sync}
  disableLoader={loadingTable}
  columns={['name', 'kols', 'elements', 'dates', 'status', 'langs', 'links']}
  sortableColumns={[
    { name: 'name', isSortable: true },
    { name: 'kols', isSortable: true },
    { name: 'elements', isSortable: true },
    { name: 'dates', isSortable: true },
    { name: 'status', isSortable: true },
    { name: 'langs', isSortable: true },
    { name: 'links', isSortable: false }
  ]}
  resource="passes"

  handleSort={handleSort}
  sortItem={sortItem}
```

Ce qui a permis d'avoir l'apparition d'une flèche cliquable mais sans que rien ne se passe dans l'ordre des PASS



J'ai donc créé un custom hook que voici :

```
import { useState } from 'react';
import { SortItemType, SortTypeEnum } from '@kolus/shared/types/table.types';

export function useSortItem() {
  const [sortItem, setSortItem] = useState<SortItemType>({
    name: undefined,
    sortType: undefined,
  });

  const handleSort = (name: string) => (event: any) => {
    setSortItem((prevSortItem) => {
      if (prevSortItem.name === name) {
        return {
          name: prevSortItem.sortType === SortTypeEnum.asc ? name : undefined,
          sortType: prevSortItem.sortType === SortTypeEnum.asc ? SortTypeEnum.desc : undefined,
        };
      }
      return { name, sortType: SortTypeEnum.asc };
    });
  };

  return { sortItem, handleSort };
}
```

Il nous permet de récupérer le nom de la colonne cliquée, ainsi que l'orientation de la flèche.

Avec tout ça en main j'ai donc trouvé le fichier d'appel API pour le tableau et donc procédé à l'organisation des PASS directement après l'appel API et avant qu'ils soient envoyés au tableau.

Le premier souci a été que « passes » contient plusieurs informations et pas seulement les PASS, ce que j'ai mis du temps à comprendre car l'erreur lors des tests était « passes is not an array »

Avec des consoles log j'ai trouvé que passes était un objet d'objets et que les PASS étaient donc un array(rows) dans un objet qui lui-même est dans l'objet passes.

J'ai donc pu écrire mon code pour organiser les PASS.

```
sortedPasses = [...res.passes.rows];
sortedPasses.sort((a, b) => {
  const order = sortItem.sortType === SortTypeEnum.asc ? 1 : -1;

  switch (sortItem.name) {
    case 'name':
      if (a.title && b.title) {
        return (a.title > b.title ? order : -order);
      } else if (a.title) {
        return -1;
      } else if (b.title) {
        return 1;
      }
      return 0;
    case 'kols':
      return (b.tickets.length - a.tickets.length) * order;
    case 'elements':
      return (b.nbActiveNodes - a.nbActiveNodes) * order;
    case 'dates':
      const dateA = new Date(a.updatedAt);
      const dateB = new Date(b.updatedAt);
      return (dateB - dateA) * order;
    case 'status':
      if (a.status && b.status) {
        return (a.status > b.status ? order : -order);
      } else if (a.status) {
        return -1;
      } else if (b.status) {
        return 1;
      }
      return 0;
    case 'langs':
      const langsA = a.langs.sort();
      const langsB = b.langs.sort();
      let i = 0;
      while (i < langsA.length && i < langsB.length) {
        const comparison = langsA[i].localeCompare(langsB[i]);
        if (comparison !== 0) {
          return comparison * order;
        }
        i++;
      }
      return (langsA.length - langsB.length) * order;
    default:
      return 0;
  }
});
sortedPassesCache[cacheKey] = sortedPasses;
```

Ici avec [...res.passes.rows] nous décomposons l'array rows et on utilise le hook créé avec une constante « order »

Enfin, selon le nom de la colonne cliquée, nous avons dans un switch le tri adapté.

```
res.passes.rows = sortedPasses;
```

On change donc rows avec nos PASS triés et le tri fonctionne.

| TITRE | NOMBRE DE KOLS ↑ | ÉLÉMENTS PUBLIÉS | DATES | STATUTS | LANGUES | PARTAGER | |
|---------------|------------------|------------------|---|----------|----------|----------|---|
| Parcours DÉMO | 198 | 26 | Publié 23/08/2024 Modifié 23/08/2024 | EN LIGNE | Français | Partager | 🔗 |
| Pass Carla | 17 | 14 | Publié 01/07/2024 Modifié 12/06/2024 | EN LIGNE | Français | Partager | 🔗 |
| Demoo Samar | 16 | 13 | Publié 05/07/2024 Modifié 05/07/2024 | EN LIGNE | Français | Partager | 🔗 |
| Test SAMES | 15 | 7 | Publié 10/07/2024 Modifié 10/07/2024 | EN LIGNE | Français | Partager | 🔗 |
| TEST WILL | 14 | 5 | Publié 26/06/2024 Modifié 10/07/2024 | EN LIGNE | Français | Partager | 🔗 |

2.3 - Mise à jour de test playwright

Playwright est un outil open-source développé par Microsoft pour l'automatisation des tests de bout en bout des applications web. Il permet de simuler les actions d'un utilisateur sur un navigateur, comme cliquer, taper du texte, ou naviguer sur des pages web, afin de tester automatiquement le bon fonctionnement de sites ou d'applications web.

La plateforme ayant quelques soucis de bug d'écran noir ou blanc, il m'a été confié de réadapter les tests Playwright pour que le développeur puisse repérer si leurs modifications créent des bugs.

Cette mission a pris une bonne partie de mon stage car j'ai vraiment essayé de ne pas bâcler les tests et d'avoir des vraies points de contrôle, la plateforme utilisant Mui Tools pour les boutons et i18n React pour la traduction ça a rendu la tâche difficile car Playwright ne trouvait pas des Boutons qui étaient pourtant visibles et fonctionnels.

Ceci étant dit, la vraie difficulté a été dans la répétition du code et le temps d'attente pendant que les tests se déroulaient.

Mais j'ai su utiliser ce temps mort à bon escient. J'ai eu le droit à un cours sur le Growth Marketing par le CEO, un autre sur le rôle de CMO (Chief marketing officer) par la CMO et un dernier sur les étapes dans la création d'un SaaS et les erreurs à éviter par mon maître de Stage

```
test('create simple kol & close kol & open kol', async ({ page }) => {
  // await new Promise(resolve => setTimeout(resolve, 15000));

  await page.goto('http://localhost:3000/panel');
  // await page.locator('#grouped-select').click();
  // await page.locator('text=Testegmm').click();
  await page.locator('text=Lancer un Kol').click();
  await page.locator('text=Kol rapide').click();
  await page.locator('[placeholder="07 00 00 00 00"]').click();
  await page.locator('input[name="phone"]').fill('+33 6 95 17 05 45');
  await page.locator('[placeholder="Titre \\(optionnel\\)"]').fill('TEST' + String);
  await page.locator('text=Lancer le Kol').click();
  await page.locator('text=Fermer Kol').click();
  await page.waitForSelector('text=Ouvrir Kol');
  await page.locator('text=Ouvrir Kol').click();
  |
});
```

Nous avons ici un exemple de test que j'ai créé pour tester la fonctionnalité de création, ouverture et fermeture d'un kol* (un ticket).

3 - Correction d'un bug dans le tri du tableau

A la fin de ma deuxième mission, l'équipe Sales m'a rapporté un bug, le tri ne s'effectuait que sur les PASS affichés.

En effet, le tableau permet d'en afficher 10, 25, 50 ou 100 donc si l'utilisateur possède plus de 10 PASS, le tri n'est pas correct.

Pour corriger ce bug, j'ai dû identifier ce qui limitait le triage.

Simplement l'API est appelée avec le nombre de PASS que le tableau affiche, c'est-à-dire que si le tableau affiche 10 PASS l'API appellera seulement les 10 premiers pass en ignorant les autres.

Pour régler ce problème, j'ai commencé par bypasser la limite de PASS appelée par l'API :

```
limit: sortItem.name ? Number.MAX_SAFE_INTEGER
```

Ensuite, dans l'atome table j'ai récupéré le nombre de rows à afficher ainsi que la page dans laquelle on se trouve :

```
Tablepage = bottomPaginationCurrentPage;
```

```
const onChangeRowsPerPage = (event: React.ChangeEvent<HTMLInputElement | HTMLTextAreaElement>) => {  
  setRowsPerPage(parseInt(event.target.value, 10));  
  rows = parseInt(event.target.value, 10)  
  
  initPaginations();  
};
```

On peut donc ajuster la limit de PASS appelée :

```
limit: sortItem.name ? Number.MAX_SAFE_INTEGER : rows,
```

Ici on appelle tous les PASS si sortItem est défini, donc si l'utilisateur veut trier sinon on le limite au nombre ligne par page.

Enfin, pour couvrir le cas de changement de page tout en ayant un tri actif, on vient récupérer la page actuelle, le nombre de ligne et slice c'est-à-dire récupérer la partie de l'array qui nous intéresse :

```
const startIndex = (Tablepage - 1) * rows;  
const endIndex = startIndex + rows;  
res.passes.rows = sortedPasses.slice(startIndex, endIndex);
```

Avec tout ceci, le code a été corrigé, une question sur les performances du serveur s'est posée mais le cas où plusieurs clients possèdent plus de 50 PASS et trient ce tableau en même temps a une probabilité très rare.

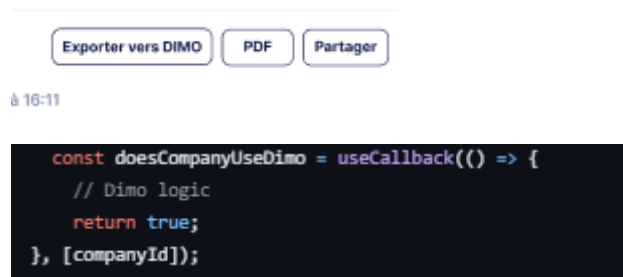
4 - Intégration de l'API DIMO – Maint

Suivant la récente levée de fonds de mai 2024, on a eu accès aux API de DIMO Maint, une entreprise lyonnaise qui propose un logiciel de gestion de la maintenance. Ils utilisent aussi un système de ticket et le but était de pouvoir transférer un kol en bulletin d'intervention DIMO Maint.

Pour des raisons de confidentialités, je ne peux montrer l'écran d'API.

J'ai documenté pour mon maître de stage le fonctionnement des API et ce qui pouvait être transféré ou non, une fois cette partie faite je me suis attelé à l'intégration de cette option dans la plateforme.

Sachant que tous les clients ne sont pas clients DIMO Maint, j'ai créé une condition pour l'affichage de ce bouton :



En m'inspirant d'autres modal déjà existant, j'ai pu créer ce modal* pour l'utilisation de l'API DIMO Maint :

Je n'ai pas eu de souci particulier avec cette dernière mission, c'est celle-ci qui m'a fait sentir comme un vrai employé de la tech, malheureusement je n'ai pas pu la terminer puisque mon stage s'est terminé avant.

Conclusion

État du projet à la fin du stage

À la fin de mon stage, la plateforme était en cours d'amélioration. Elle intègre maintenant la traduction instantanée par exemple. Mes objectifs principaux ont été atteints, notamment le tri du tableau et les tests playwright. Cependant, je n'ai eu le temps de finir l'intégration de l'API DIMO Maint.

Mon ressenti, ma progression, mes difficultés

Ce stage a été une expérience très enrichissante. J'ai beaucoup appris, surtout des compétences Marketing et Sales car étant en open-space j'avais la possibilité d'entendre tout le monde, ce qui m'a permis d'enregistrer certaines méthodes, intonations de voix, conseils donnés aux alternants ou même des phrases multifonctions.

J'ai bien sûr énormément progressé dans l'utilisation du langage React et pu tester mon niveau dans des conditions réelles. Je me sens maintenant prêt à lancer mon propre SaaS à ma petite échelle.

J'ai rencontré des difficultés, notamment au début vu l'énorme projet qu'est la plateforme Kolus, mais j'ai réussi à les surmonter en persévérant et demandant de l'aide quand internet ne pouvait pas. Cette expérience m'a permis de prendre confiance en mes compétences et de mieux comprendre le fonctionnement d'un projet en milieu professionnel.

J'aimerais finir ce rapport par encore une fois remercier mon maître de stage Max Barbet pour m'avoir mis dans de bonnes conditions pour apprendre et avoir fait confiance en mon autonomie, mon CEO William Kelbert pour m'avoir très bien accueilli et donné l'opportunité de faire ce stage ainsi que Tout le reste de l'équipe Kolus.

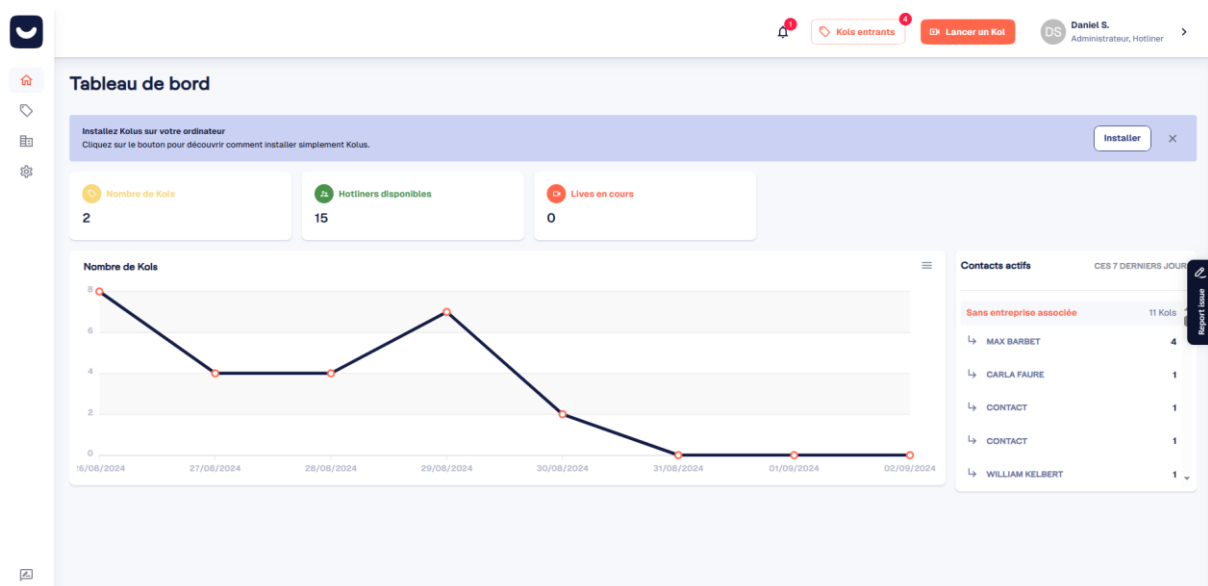
Annexes

Equipe Kolus



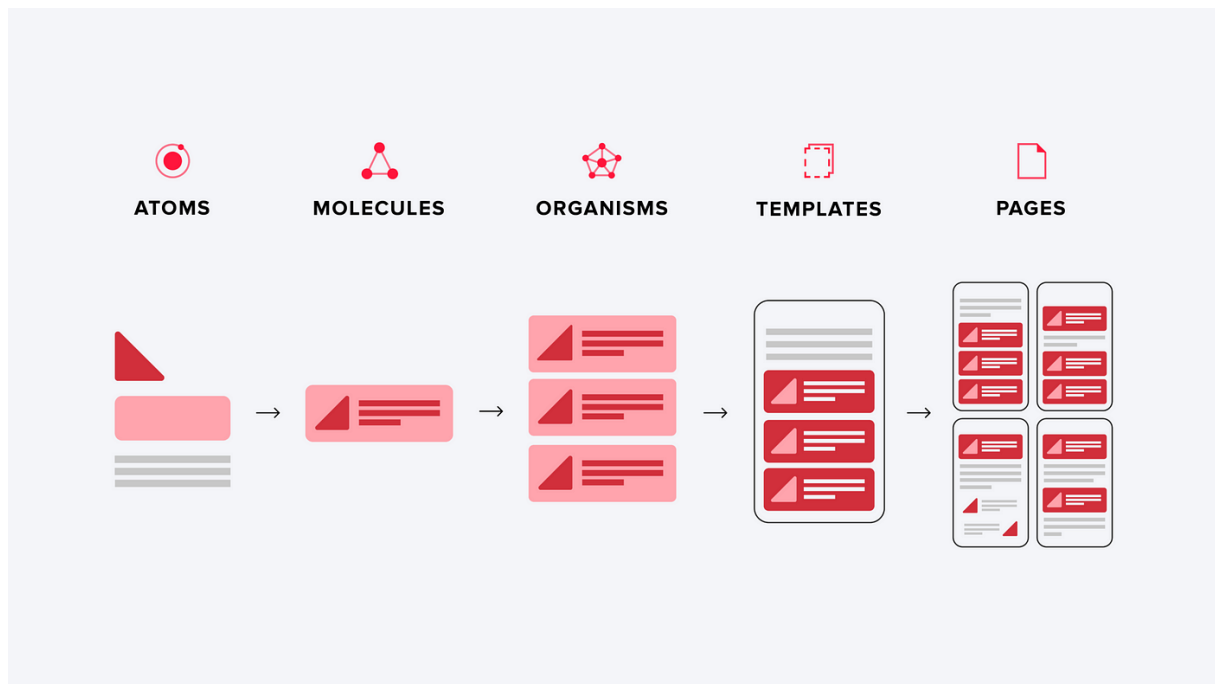
Source ([Nouvel investissement pour Evolem avec Kolus](#))

La plateforme Kolus



Atomic Design Pattern

Chaque atome élément possède sa page de code ce qui crée un code imbriqué.



Source ([Building design systems with atomic design | by Valentina Ali | Bootcamp \(uxdesign.cc\)](#))

Volume de code

Un aperçu du nombre de fichier dans le projet qui ne représente même pas 10% de l'entièreté

```
kolus-develop/server/src/services/message-notification/message-notification.hooks.ts
kolus-develop/server/src/services/message-notification/message-notification.service.ts
kolus-develop/server/src/services/message/message.class.ts
kolus-develop/server/src/services/message/message.hooks.ts
kolus-develop/server/src/services/message/message.service.ts
kolus-develop/server/src/services/notification-log/notification-log.class.ts
kolus-develop/server/src/services/notification-log/notification-log.hooks.ts
kolus-develop/server/src/services/notification-log/notification-log.service.ts
kolus-develop/server/src/services/notification/notification.class.ts
kolus-develop/server/src/services/notification/notification.hooks.ts
kolus-develop/server/src/services/notification/notification.service.ts
kolus-develop/server/src/services/pass-link/pass-link.class.ts
kolus-develop/server/src/services/pass-link/pass-link.hooks.ts
kolus-develop/server/src/services/pass-link/pass-link.service.ts
kolus-develop/server/src/services/pass-node/pass-node.class.ts
kolus-develop/server/src/services/pass-node/pass-node.hooks.ts
kolus-develop/server/src/services/pass-node/pass-node.service.ts
kolus-develop/server/src/services/pass/pass.class.ts
kolus-develop/server/src/services/pass/pass.hooks.ts
kolus-develop/server/src/services/pass/pass.service.ts
kolus-develop/server/src/services/pass/pass.utils.ts
kolus-develop/server/src/services/real-time/real-time.class.ts
kolus-develop/server/src/services/real-time/real-time.hooks.ts
kolus-develop/server/src/services/real-time/real-time.service.ts
kolus-develop/server/src/services/service-worker-device/service-worker-device.class.ts
kolus-develop/server/src/services/service-worker-device/service-worker-device.hooks.ts
kolus-develop/server/src/services/service-worker-device/service-worker-device.service.ts
kolus-develop/server/src/services/services.helpers.ts
kolus-develop/server/src/services/smart-glasses/smart-glasses.service.ts
kolus-develop/server/src/services/subscription/subscription.class.ts
kolus-develop/server/src/services/subscription/subscription.hooks.ts
kolus-develop/server/src/services/subscription/subscription.service.ts
kolus-develop/server/src/services/super-admin/super-admin.service.ts
kolus-develop/server/src/services/tag-category/tag-category.class.ts
kolus-develop/server/src/services/tag-category/tag-category.hooks.ts
kolus-develop/server/src/services/tag-category/tag-category.service.ts
kolus-develop/server/src/services/tag/tag.class.ts
kolus-develop/server/src/services/tag/tag.hooks.ts
kolus-develop/server/src/services/tag/tag.service.ts
kolus-develop/server/src/services/ticket-tag/ticket-tag.class.ts
kolus-develop/server/src/services/ticket-tag/ticket-tag.hooks.ts
kolus-develop/server/src/services/ticket-tag/ticket-tag.service.ts
kolus-develop/server/src/services/ticket-user/ticket-user.class.ts
kolus-develop/server/src/services/ticket-user/ticket-user.hooks.ts
kolus-develop/server/src/services/ticket-user/ticket-user.service.ts
kolus-develop/server/src/services/ticket/ticket.class.ts
kolus-develop/server/src/services/ticket/ticket.hooks.ts
kolus-develop/server/src/services/ticket/ticket.service.ts
kolus-develop/server/src/services/token-management/token-management.class.ts
```

La page d'un Kol

TIBETO

+33 6 24 48 07 70

Lancer le live

Renvoyer l'invitation

Notifications

Vous serez notifié des nouveaux messages dans vos Kols

Autoriser

Fin du live (a duré 1m 23s)

15:34

Début du live

15:44

Fin du live (a duré 1m 59s)

15:46

Daniel S. a renvoyé l'invitation à TI B.

17:59

Début du live

18:00

Fin du live (a duré 49s)

18:01

Début du live

18:01

Fin du live (a duré 14m 28s)

18:15

TI B. a vu jusqu'ici

Ecrire un message

BTS SIO 1

PDF

Partager

Créé le 21/06/2024 à 14:39

Identifiant : wyuwsc08

OUVERT

Fermer Kol

RAPPORT

Modifier

Pas de rapport

TAG ASSOCIÉ

Modifier

Aucun tag sélectionné

HOTLINER ASSIGNÉ

Ajouter

DS Daniel S.

COMMENTAIRE PRIVÉ

Créer

Afficher uniquement les messages des hotliners

Report Issue

Modal DIMO MAINT

Exporter un Kol vers DIMO

Transformez votre kol en un ticket DIMO

Mon BI existe déjà :

Code du BI existant :

Entrez le code

Créer

OU

Créer un nouveau BI :

Titre :

Entrez un titre

Urgence

Très Urgent

Type

Contrôle

Technologie

Hydraulique

Code personnalisé :

Laisser vide pour code auto

Annuler

Créer

18