

Développement d'un Framework générateur d'interfaces web

par
ANDRIAMANARIVO Ericko

Mémoire présenté
en vue de l'obtention du grade de Licence
en Informatique
option Développement d'application

Novembre 2022

Jury :

Mme Baovola RAJAONARISON, présidente
M. Aina RAZAFINJATOVO , examinateur
M. Steave LEONG, encadreur professionnel

Table des matières

| | |
|--|-----|
| Liste des tableaux | ii |
| Liste des figures | iii |
| Glossaire | v |
| Avant-propos | 1 |
| L'IT University | 1 |
| BICI Madagascar | 2 |
| Remerciements | 3 |
| Introduction | 4 |
| 1 Présentation du projet | 5 |
| 1.1 Objectifs du projet | 5 |
| 1.2 Planning de réalisation | 5 |
| 1.3 Technologies utilisées | 7 |
| 2 Réalisation de l'application | 10 |
| 2.1 Analyse et conception | 10 |
| 2.1.1 Analyse de l'existant | 10 |
| 2.1.2 Conception de l'application | 10 |
| 2.2 Développement par fonctionnalité ou module | 11 |
| 2.2.1 Gestion des projets | 11 |
| 2.2.2 Génération des pages APJ | 13 |
| 2.2.3 Modification des pages | 25 |
| 2.2.4 Exécution d'un build depuis le générateur | 26 |
| 2.2.5 Génération des pages d'une application de type React JS | 27 |
| 2.3 État d'Analyse et Statistiques | 32 |
| 2.3.1 Analyse des pages générées | 32 |
| 2.3.2 Analyse du temps de développement des pages | 32 |
| 2.4 Problèmes rencontrés et solutions | 33 |
| 3 Évaluation du projet et connaissances acquises | 34 |
| 3.1 Bilan pour l'entreprise | 34 |
| 3.2 Bilan personnel | 34 |
| 3.3 Extension et évolution de l'application | 34 |
| Conclusion | 35 |
| Bibliographie | 36 |

● Liste des tableaux

| | |
|---|----|
| Tableau 1 : Comparaison entre les langages Java et PHP | 7 |
| Tableau 2 : Tableau de comparaison entre les SGBD PostgreSQL et Oracle | 8 |
| Tableau 3 : Tableau de comparaison entre les technologies React JS et Angular | 9 |
| Tableau 4 : Analyse du temps de développement | 32 |

● Liste des figures

| | |
|--|----|
| Figure 1 : Diagramme de réalisation | 7 |
| Figure 2 : Différence entre le cycle de vie d'un SPA et d'une page traditionnelle. Représentation de Wasson Mike (2013) | 9 |
| Figure 3 : Modèle conceptuel des données | 11 |
| Figure 4 : Page de saisie d'un projet | 12 |
| Figure 5 : Page de liste des projets | 12 |
| Figure 6 : Page de fiche d'un projet | 13 |
| Figure 7 : Page de modification d'un projet | 13 |
| Figure 8 : Page de recherche des tables | 14 |
| Figure 9 : Page de recherche des tables en cas de CRUD multiple | 14 |
| Figure 10 : Interface pour générer un page de liste | 15 |
| Figure 11 : Exemple de page liste généré après un build | 15 |
| Figure 12 : Interface pour générer un page de saisie | 16 |
| Figure 13 : Exemple de page de saisie générée après un build | 16 |
| Figure 14 : Interface pour générer un page de saisie multiple | 17 |
| Figure 15 : Exemple de page de saisie multiple générée après un build | 18 |
| Figure 16 : Interface pour générer un page de modification | 18 |
| Figure 17 : Exemple de page de modification générée après un build | 19 |
| Figure 18 : Interface pour générer un page de modification multiple | 19 |
| Figure 19 : Exemple de page de modification multiple générée après un build | 20 |
| Figure 20 : Interface pour générer un page de fiche | 21 |
| Figure 21 : Exemple de page de fiche générée après build | 21 |
| Figure 22 : Interface pour générer un page d'analyse | 22 |
| Figure 23 : Exemple de page d'analyse générée après un build | 23 |
| Figure 24 : Interface pour générer un page de graphe | 24 |
| Figure 25 : Exemple de page de graphique après un build | 24 |
| Figure 26 : Fiche d'une page générée | 25 |
| Figure 27 : Interface de modification d'une page | 26 |
| Figure 28 : Paramètre de build dans la génération d'une page | 26 |
| Figure 29 : Liste des pages qui n'ont pas encore été buildés dans le projet | 27 |
| Figure 30 : Menu dynamique | 28 |
| Figure 31 : Interface pour générer une page de liste de type React | 29 |

| | |
|---|----|
| Figure 32 : Exemple de page React de liste générée | 29 |
| Figure 33 : Interface pour générer une page de saisie | 30 |
| Figure 34 : Page de saisie générée | 30 |
| Figure 35 : Interface pour générer une page de graph | 31 |
| Figure 36 : Page de graph générée | 31 |
| Figure 37 : Analyse des pages générées par projet et par type de page | 32 |

● Glossaire

API: Application Programming Interface. Il s'agit d'un outil permettant la communication entre différentes applications.

APJ: Affichage Personnalisé JRews

AJAX: Asynchronous JavaScript And XML.

Backend : Partie d'une application qui sert indirectement le frontend

EJB : Architecture de composants logiciels côté serveur pour la plateforme de développement Java EE

Framework: Ensemble d'outils fournissant des solutions personnalisées pour accélérer le développement d'une application

Frontend : Partie d'un système informatique ou d'une application avec laquelle l'utilisateur interagit directement.

Java EE: Java Enterprise Edition

JSON: Javascript Object Notation

JSP: Java Server Pages

Low code: Principe de programmation par laquelle les développeurs développent des applications à partir d'une interface graphique

SGBD: Système de gestion de base de données

SPA: Single Page Application ou Application à page unique

XML. Extensible Markup Language : Langage à balises libre qui sert à structurer, enregistrer et échanger des données

● **Avant-propos**

Le présent mémoire présente les résultats du travail effectué lors de mon stage de fin d'études de Licence en Informatique de l'IT University; stage effectué au Bureau International de Consultance en Informatique (BICI) durant 3 mois, du mois de Juin 2022 jusqu'au mois de Septembre 2022.

Afin de poser clairement le contexte de ce mémoire, je vais présenter succinctement d'une part l'IT University et d'autre part mon entreprise d'accueil.

○ **L'IT University**

Fondée en 2011, l'IT University (ou ITU) est une université privée, spécialisée en informatique, formant les jeunes bacheliers, de préférence scientifiques :

- En trois ans, pour l'obtention d'une :
 - Licence, option Développement, Réseaux et Bases de Données ou Web et Design
 - Licence, option Graphic Design ou Communication Digitale
- En cinq ans, pour l'obtention d'un :
 - Master MBDS en coopération avec l'Université Côte d'Azur à Nice Sophia Antipolis – France
 - Master BIHAR en coopération avec l'ESTIA du Pays Basque - France

Étant une formation professionnalisante, l'ITU a tissé des liens forts avec ses partenaires industriels, dont l'opérateur convergent TELMA et la plupart des entreprises et institutions du secteur des TIC¹. Ces partenaires participent effectivement à la formation par la fourniture de connexion Internet à haut débit, l'envoi de conférenciers ou par l'accueil des étudiants en stage. Beaucoup de ces partenaires recrutent aussi les sortants dès leur sortie d'École.

D'autre part, le corps enseignant de l'ITU est constitué intégralement de spécialistes de très haut niveau et obligatoirement actifs professionnellement dans leurs domaines respectifs.

Enfin, l'ITU est une Microsoft Imagine Academy et un Oracle Gold Partner.

¹ TIC : Technologies de l'Information et de la Communication

○ **BICI**

Le BICI ou Bureau International de Consultance en Informatique est une branche du groupe « Vidy Varotra », une société de Service en Ingénierie Informatique Malgache créée en 2003, sous l'appellation « JRews Informatique » à l'époque. Sa principale activité est l'assistance informatique et BICI offre aujourd'hui les services suivants :

1. Développement d'applications d'entreprises :

Le BICI est une entreprise spécialisée dans le développement d'applications d'entreprises et possède une équipe de développeurs experts en technologies variées. Il conçoit des solutions logicielles personnalisées en écoutant les besoins spécifiques de ses clients. L'entreprise assure également l'intégration, les tests, la sécurité des données et offre un support continu pour garantir le bon fonctionnement des applications sur le long terme.

2. Audit informatique :

Le BICI fait aussi l'audit informatique, il possède une expertise technique pour évaluer les systèmes informatiques, identifier les vulnérabilités et proposer des améliorations.

3. Formation (en partenariat avec l'IT University) :

BICI contribue aussi à la formation des étudiants de l'IT University car il dispose d'une expertise en informatique, propose des formateurs qualifiés, des contenus de formation pertinents et des projets pratiques. L'entreprise offre également des stages et collabore étroitement avec les enseignants pour soutenir le développement des étudiants.

○ **Remerciements**

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Tout d'abord, je tiens à exprimer toute ma reconnaissance aux membres du jury, qui m'ont fait l'honneur de bien vouloir étudier avec attention mon travail.

Je désire aussi remercier les professeurs de l'IT University qui m'ont fourni les outils nécessaires à la réussite de mes études universitaires.

Mes sincères gratitude vont également vers la société BICI, de m'avoir accueilli pour ce stage de trois mois, et m'a fourni tous les éléments dont j'ai eu besoin pour mener à bien mes travaux.

Enfin, je tenais à exprimer sincèrement toute ma reconnaissance, à toutes les autres personnes qui, même sans être citées distinctement dans ce présent mémoire, ont contribué de près ou de loin, à en garantir son aboutissement et sa réussite.

● Introduction

L'essentiel du travail que j'ai réalisé a porté sur le thème :

“Développement d'un Framework générateur d'interfaces d'une application avec APJ”

Le développement d'une application grâce à l'aide d'un framework permet d'accélérer son temps de développement, et augmente sa fiabilité car un framework fournit les fonctionnalités standards afin que les développeurs se concentrent plus sur les fonctionnalités spécifiques de l'application qu'ils développent. BICI dispose de son propre environnement interne nommé APJ, afin d'accélérer le développement des applications, chaque développeur doit se conformer aux standards de cet environnement, c'est-à-dire plusieurs bibliothèques personnalisées et optimisées et des fonctions généralisées.

Cependant, pour créer des pages, les développeurs doivent copier-coller des codes des pages antérieures et les modifier à chaque fois. C'est une tâche assez itérative et provoque souvent des bugs au cours du développement.

Par conséquent, il était nécessaire de développer un outil permettant d'optimiser le développement des applications avec APJ en donnant comme résultats :

- La gestion des projets en développement
- La génération des pages comme les pages de liste, saisie, analyse et graphique basée sur les tables de la base de données de chaque projet.
- La possibilité de modifier les pages générées.
- La possibilité d'exécuter un build depuis le générateur et de consulter directement les changements effectués dans un projet.

Ce sont les tâches demandées par la société, mais pour ma contribution, j'ai aussi ajouté une option permettant de générer des interfaces pour les applications de type React JS comme couche de présentation.

A travers ce présent mémoire, nous verrons dans un premier temps la présentation du projet réalisé, nous devons également exposer la réalisation de l'application avec les différentes étapes et méthodes afin de mener à bien son développement, avant de finalement faire le bilan du stage et les résultats obtenus.

1 Présentation du projet

1.1 Objectifs du projet

L'objectif principal du projet est d'accélérer le développement des applications développées avec APJ, en générant les interfaces comme les CRUD, les analyses et les graphiques. L'utilisateur aura la possibilité de générer soit une interface de type APJ, soit une interface de type React JS dépendant de la couche de présentation de son application. C'est une plateforme de développement "low code", c'est-à-dire qui aura pour but de simplifier le développement d'une application afin que les développeurs développent des applications plus rapidement au moyen d'une interface graphique et puissent se consacrer plus sur les fonctionnalités plus complexes.

1.2 Planning de réalisation

Le stage effectué au sein de l'entreprise BICI a duré 3 mois, en plus de la réalisation du projet, d'autres travaux ont été réalisés. Ces 3 mois de stage sont détaillés dans le diagramme ci-dessous :

Figure 1 : Diagramme de réalisation

1.3 Technologies utilisées

- **Java EE (Enterprise Edition)**

Java EE est une plateforme de développement d'application web basée sur le langage Java, son utilisation offre plusieurs avantages tels qu'une architecture basée sur les composants permettant le découpage des couches de développement, la prise en charge de plusieurs API puissantes et la possibilité de choisir les outils et serveurs pour le développement d'une application. Voyons dans le tableau ci-dessous les avantages que présente Java par rapport à d'autres langages de développement web comme PHP :

Tableau 1 : Comparaison entre les langages Java et PHP

| | Java | PHP |
|-------------|--|--|
| Performance | Plus performant car c'est un langage précompilé | Prends plus de temps lors de la compilation |
| Stabilité | Plus stable car le code est plus long et bien structuré | Moins stable car le code présente plus de risque de crashes |
| Sécurité | Considéré comme le langage de référence en matière de sécurité | Moins sécurisé, utilise d'autres frameworks pour la sécurité |

- **APJ**

APJ est un environnement interne de BICI basé sur l'architecture Java EE, avec ses bibliothèques personnalisées et optimisées, et ses fonctions généralisées, il permet de développer une application très rapidement.

- **Oracle**

Oracle est un SGBD développé par Oracle Corporation, c'est l'un des moteurs de base de données les plus fiables et les plus utilisés et qui est adapté pour les entreprises. Il possède une mise en mémoire cache efficace pour assurer la performance optimale des très grandes bases de données. Voyons dans le tableau ci-dessous les avantages que présente Oracle par rapport à d'autres SGBD comme PostgreSQL :

Tableau 2 : Tableau de comparaison entre les SGBD PostgreSQL et Oracle

| | PostgreSQL | Oracle |
|---|--|--|
| Productivité | Assez faible car elle fournit moins de transactions par seconde qu' Oracle. | Élevée car elle est supérieure techniquement et fournit plus de transactions par seconde que PostgreSQL. |
| Sécurité | A une bonne sécurité, mais pas aussi avancée que celle d'Oracle | A plus de sécurité et une sécurité plus avancée |
| Traitement d'un important volume de données | Traite une grande quantité de données de manière efficace, mais pas plus qu' Oracle, et dépend plus du serveur | Traite une grande quantité de données de manière plus efficace que PostgreSQL |

- **React JS**

React JS est une librairie Javascript développée par Facebook qui permet de créer des interfaces utilisateurs d'une application de type Single Page Application (SPA), c'est-à-dire une application qui charge une page unique comme son nom l'indique et met à jour son contenu via l'appel des APIs lorsqu'il y a un changement. Les ressources de la page unique ne sont chargées qu'une seule fois pendant la durée de fonctionnement de l'application et seules les données sont transmises entre le client et le serveur par l'intermédiaire des appels AJAX essentiellement sous format JSON. Cette approche est complètement différente de celui d'un site web classique ou application multipage comme le montre la figure ci-dessous :

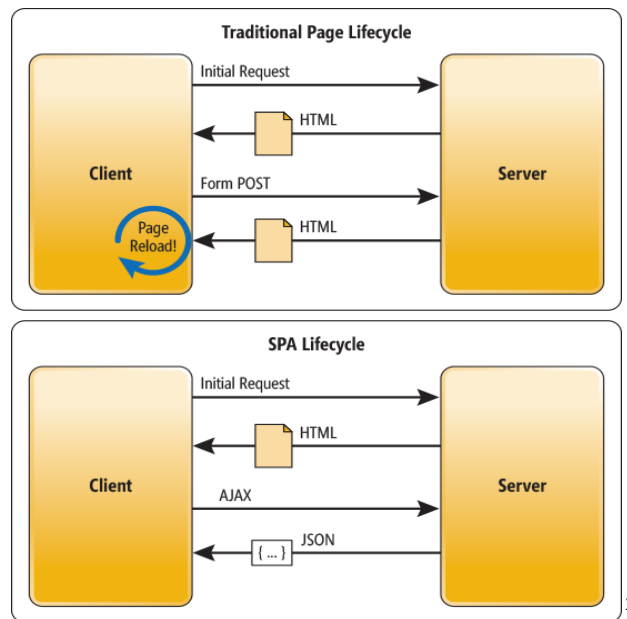


Figure 2 : Différence entre le cycle de vie d'un SPA et d'une page traditionnelle.
Représentation de Wasson Mike (2013)

Pour les sites traditionnels, à chaque requête, le serveur génère une nouvelle page et déclenche un rafraîchissement du navigateur contrairement aux SPA qui ne se chargent qu'au moment de la requête initiale et sont plus fluides et plus réactives. Une autre avantage des SPA est que la transmission des données de l'application crée une séparation entre la couche présentation et la couche logique de l'application et change la manière de structurer une application web.

Voyons dans le tableau ci-dessous la différence entre React JS et une autre technologie SPA qu'est Angular :

Tableau 3 : Tableau de comparaison entre les technologies React JS et Angular

| | React JS | Angular |
|--------------------------|--|---|
| Performance | Plus performant grâce au DOM virtuel et la liaison des données à sens unique | Moins performant avec le DOM réel et la liaison des données à sens double |
| Architecture | Plus flexible | Fixe et assez complexe |
| Langage de développement | Javascript | Typescript |

| | | |
|-------------|-------------------------------|-------------------------------|
| Utilisation | Applications web plus lourdes | Applications web interactives |
|-------------|-------------------------------|-------------------------------|

2 Réalisation de l'application

2.1 Analyse et conception

2.1.1 Analyse de l'existant

Durant mon stage chez BICI, j'ai participé au développement de plusieurs projets sur l'environnement que l'entreprise utilise en interne dénommé APJ. Le déroulement du développement d'une application avec APJ commence par la création des tables et vues dans la base de données du projet, ensuite, il faut créer les classes correspondantes, et enfin, créer les pages jsp pour la couche présentation. Pour développer les interfaces qui sont assez complexes avec APJ, les développeurs doivent suivre des modèles des projets antérieurs, et changer les configurations pour chaque page, mais cela entraîne souvent des bugs car il y a beaucoup de paramètres à prendre en compte. Cela diminue la productivité chez les développeurs, et le ralentissement du temps de développement d'un projet sachant qu'il faut développer de nombreuses pages donc la tâche est assez répétitive. Ainsi, nous avons développé un outil pour générer les interfaces afin d'accélérer le développement d'une application APJ et atténuer les risques de bugs.

2.1.2 Conception de l'application

Afin de mener à bien le développement du projet, il est crucial d'établir la conception c'est-à-dire de planifier et organiser le développement en plusieurs étapes. La conception permet de comprendre et atteindre les objectifs du développement, minimiser les risques, gagner du temps pendant le développement et de prendre en compte les améliorations futures.

Nous avons donc établi les travaux suivants après avoir effectué la conception:

- Nombre de tables créées : 4
- Nombre de tables vues : 2
- Nombre de classes : 48
- Nombre d'écrans : 25

Voici le modèle conceptuel de données qu'on a utilisé :

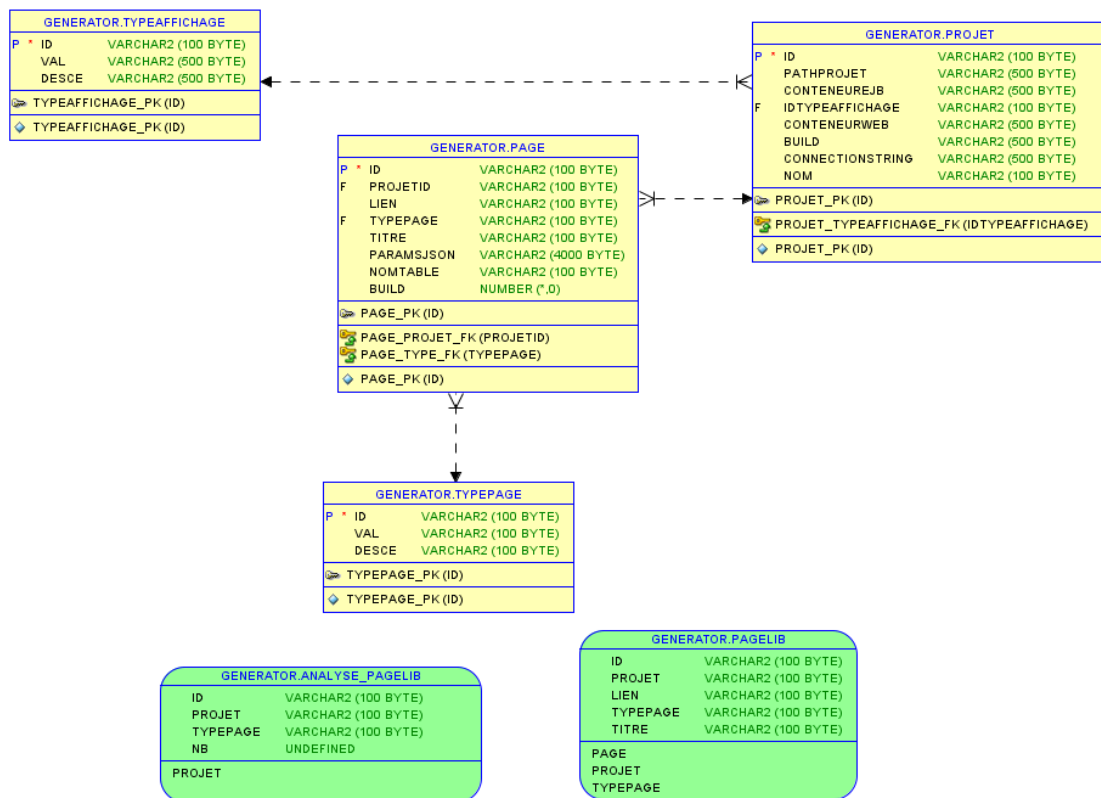


Figure 3 : Modèle conceptuel des données

2.2 Développement par fonctionnalité ou module

2.2.1 Gestion des projets

Le but du projet étant de générer les pages d'un projet APJ, il était donc primordial de pouvoir gérer les projets, c'est-à-dire d'ajouter, lister, modifier, et consulter la fiche d'un projet. Pour ajouter un projet, il nous faut le chemin du projet, le chemin de son conteneur EJB, le type de l'affichage du projet (JSP ou React), le chemin de son conteneur web, le nom de son fichier build XML nécessaire pour builder le projet et enfin la connection du projet sous-forme de connection String pour se connecter à sa base de données. Nous avons la possibilité de saisir, lister, consulter, et modifier comme représenté dans les figures ci-dessous :

Saisie de projet

| | |
|----------------|---|
| Path Projet | C:\Works\Stage\apj-generator |
| Conteneur EJB | C:\Works\Stage\apj-generator\APJ-Generator-ejb |
| Type Affichage | APJ ▼ |
| Conteneur WEB | C:\Works\Stage\apj-generator\APJ-Generator-war |
| Build | apj-generator-build-exploded-application.xml |
| Connection | jdbc:oracle:thin:@26.41.116.127:1521:orcl;prod;prod |
| Nom | APJ Generator |

Réinitialiser Enregistrer

Figure 4 : Page de saisie d'un projet

Liste des projets

Recherche avancée +

Id

Type Affichage
 Tous ▼

Choix des colonnes à afficher +

Afficher
Réinitialiser

Tri +

| Récapitulation | | | | |
|----------------|--------|--|--|--|
| | Nombre | | | |
| Total | 0 | | | |

| LISTE | | | | |
|--------------------|------------------------------|--|----------------|--|
| Id | Path Projet | Conteneur EJB | Type Affichage | Conteneur WEB |
| PR0000000061115650 | D:\Works\Stage\APJ-Generator | D:\Works\Stage\APJ-Generator\APJ-Generator-ejb | TA0001 | D:\Works\Stage\APJ-Generator\APJ-Generator-war |
| PR0000000041215654 | D:\Works\BIC\teamtask | D:\Works\BIC\teamtask\teamTask-ejb | TA0001 | D:\Works\BIC\teamtask\teamTask-war |
| PR0000000021062425 | D:\Works\Stage\APJ-Generator | D:\Works\Stage\APJ-Generator\APJ-Generator-ejb | TA0002 | D:\Works\Stage\prod-react |
| PR000000008121502 | D:\Works\BIC\prod | D:\Works\BIC\prod\prod-ejb | TA0001 | D:\Works\BIC\prod\prod-war |
| PR000000007120927 | OI | OI | TA0001 | OI |

Figure 5 : Page de liste des projets

Fiche du projet

| | |
|------------------|---|
| id | PR0000000121150946 |
| pathProjet | C:\Works\BICI\prod |
| Conteneur EJB | C:\Works\BICI\prod\prod-ejb |
| idTypeAffichage | TA0002 |
| Conteneur WEB | C:\Works\Stage\template-apj-react-prod |
| build | prod-build-exploded-application.xml |
| connectionString | jdbc:oracle:thin:@26.41.116.127:1521:orcl;prod;prod |
| nom | Production React |

[Liste des pages non-buîdés](#)
[Liste des pages](#)
[Générer pages](#)
[Modifier](#)

[Effectuer un build](#)

Figure 6 : Page de fiche d'un projet

Modification projet

| | |
|----------------|--|
| Path Projet | <input type="text" value="C:\Works\BICI\prod"/> |
| Conteneur EJB | <input type="text" value="C:\Works\BICI\prod\prod-ejb"/> |
| Type Affichage | <input type="text" value="React"/> ▼ |
| Conteneur WEB | <input type="text" value="C:\Works\Stage\template-apj-react-prod"/> |
| Build | <input type="text" value="prod-build-exploded-application.xml"/> |
| Connection | <input type="text" value="jdbc:oracle:thin:@26.41.116.127:1521:orcl;prod;prod"/> |
| Nom | <input type="text" value="Production React"/> |

[Réinitialiser](#)
[Enregistrer](#)

Figure 7 : Page de modification d'un projet

2.2.2 Génération des pages APJ

La génération des pages APJ permet de développer des pages plus rapidement sans se soucier du code qui constitue la page. Elle est basée sur la structure de la base de données existante d'un projet. Pour générer une page, il faut tout d'abord choisir un projet parmi les

projets présents dans l'application. Ensuite, on doit choisir la table ou la vue de base de la page à générer. Il faut choisir 2 tables s'il s'agit d'un CRUD multiple.

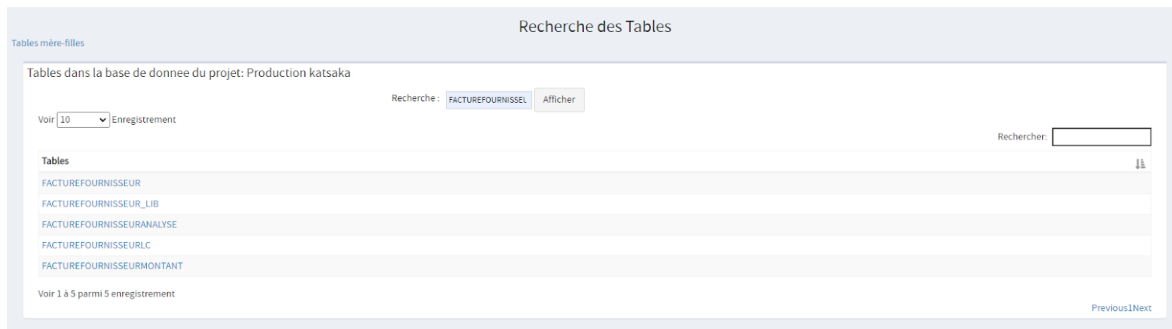


Figure 8 : Page de recherche des tables

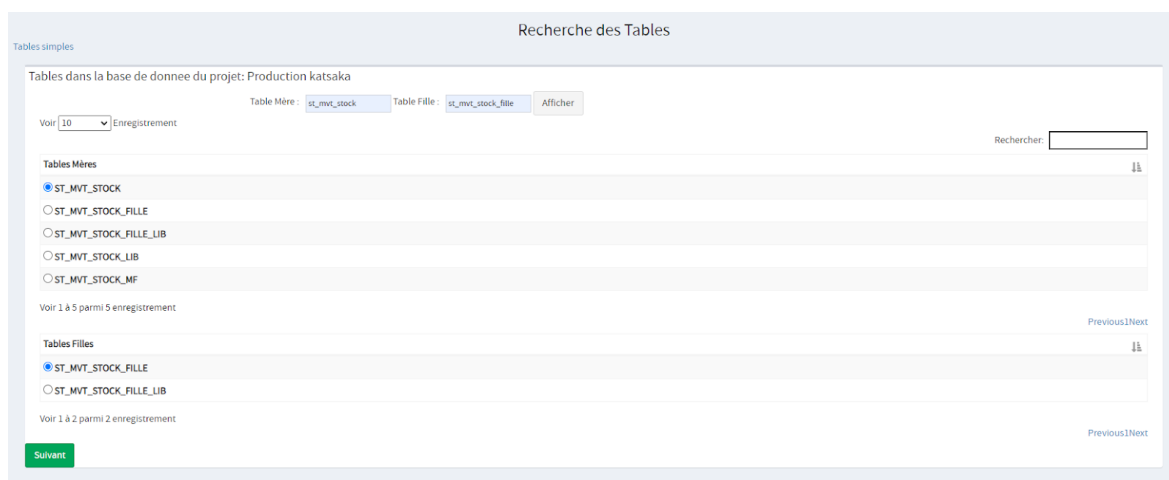


Figure 9 : Page de recherche des tables en cas de CRUD multiple

Après avoir choisi la table de base, il faut maintenant choisir le type de page à créer soit une page de saisie, liste, analyse, graphe, modification ou fiche.

2.2.2.1 Générer une page de liste

Une page de liste est très importante pour lister les données d'une entité dans une application, elle permet aussi de filtrer ces données par rapport aux critères voulus. Pour générer une page de liste, il faut entrer les informations suivantes : le titre de la page, le lien de la page, le nom de la table et la classe de mapping correspondante et le nombre d'éléments par page pour la pagination. Ensuite, on peut choisir si on veut afficher le tableau de récapitulation des données ou encore d'effectuer un build après avoir généré la page afin de pouvoir consulter directement les changements dans le projet correspondant après. Et enfin, on peut paramétrer les colonnes à afficher pour la liste et changer leurs libellés ou encore ajouter des liens pour chaque colonne avec les paramètres.

Générer une Page Recherche à partir de : FACTUREFOURNISSEUR_LIB

Informations de la page

Titre
Liste des Facturefournisseur_libs

Lien
Lien vers cette page(.jsp ou .js)

Classe De Mapping (Avec Package)
stock.FactureFournisseur

☒ **Nom De La Table**
facturefournisseur_lib

Nombre D'éléments Par Page
20

☐ Afficher Le Tableau De Récapitulation

☐ Effectuer Un Build Après Génération De La Page

Champs

Voir 10 Enregistrement

Rechercher:

| Entêtes | Nom Du Colonne | Libellé | Type | A Sommer | Critère | Options Du Critère | Lien | Option Du Lien |
|-------------------------------------|----------------|---------------|--------|-------------------------------------|-------------------------------------|--------------------|--------------------------|----------------|
| <input checked="" type="checkbox"/> | ID | ID | String | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DATY | DATY | Date | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | ETAT | ETAT | double | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | REF | REF | String | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDFOURNISSEUR | IDFOURNISSEUR | String | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |

Voir 1 à 5 parmi 5 enregistrement

Previous1Next

Générer

Cette page comporte aussi un formulaire de recherche avancée qu'on peut paramétrer en choisissant les critères visibles et ajoutant les options comme leurs libellés, leurs types : liste déroulante, intervalle ou popup, ou encore choisir les colonnes à sommer pour la récapitulation des données.

Figure 10 : Interface pour générer un page de liste

Liste des Facturefournisseur_libs

Recherche avancée

ID DATY ETAT

REF IDFOURNISSEUR

Choix des colonnes à afficher ☒ Récapitulation Tri

Afficher **Réinitialiser**

LISTE

| ID | DATY | ETAT | REF | IDFOURNISSEUR |
|----------|------------|------|-----|---------------|
| FF000102 | 11/07/2022 | 11 | | Laetitia |

Nombre de résultat : 1

page 1 sur 1

Exporter

Figure 11 : Exemple de page liste généré après un build

2.2.2.2 Générer une page de saisie

Pour insérer des informations dans une application, nous avons besoin des formulaires de saisie. Pour en créer, on a besoin des informations suivantes : le titre de la page, le lien qui mène vers la page, le nom de la table avec la classe de mapping correspondante, la page de traitement et enfin le but ou la page de destination après avoir validé le formulaire.

Il nous faut aussi configurer les champs du formulaire qui par défaut seront tous affichés, il est possible pour un champ de changer sa visibilité, changer son libellé, définir sa valeur par défaut, de le mettre en lecture seul et de changer son type, soit en champ libre, liste déroulante ou popup.

Figure 12 : Interface pour générer un page de saisie

Figure 13 : Exemple de page de saisie générée après un build

La saisie multiple est utile lorsqu'on a besoin d'insérer les données de type mère-filles. Pour générer une page de tel type, la configuration est la même que celle d'une page de saisie simple, mais en plus, nous devons ajouter la table mère avec la classe mère correspondante et la table fille avec la classe fille correspondante. Pour les champs, il faut configurer les champs de la table mère et ceux de la table fille comme pour les champs de la saisie simple.

Générer une Page de Saisie Multiple à partir des tables : St_mvt_stock | St_mvt_stock_fille

Information de la Page

Titre
Saisie mouvement de stock

Lien
test/mvtstock-saisie.jsp

Classe De Mapping Mère
stock.MvtStock

Nom De La Table Mère
St_mvt_stock

Classe De Mapping Fille
stock.MvtStockFille

Nom De La Table Fille
St_mvt_stock_fille

Page Après
apresTarif.jsp

But

☒ Effectuer Un Build Après Génération De La Page

Champs table mère :

Voir 10 Enregistrement

Rechercher:

| Visibilité | Nom Du Colonne | Libellé | Type | Valeur Par Défaut | Read Only | Autres |
|-------------------------------------|----------------|-------------|--------|-------------------|-------------------------------------|--------|
| <input checked="" type="checkbox"/> | ID | ID | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DATY | DATY | Date | 23/09/2022 | <input type="checkbox"/> | ... |
| <input type="checkbox"/> | DESIGNATION | DESIGNATION | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDMAGASIN | IDMAGASIN | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDBL | IDBL | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | ETAT | ETAT | double | 0 | <input checked="" type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDTRANSFERT | IDTRANSFERT | String | | <input type="checkbox"/> | ... |

Voir 1 à 7 parmi 7 enregistrement

Previous1Next

Champs table fille

| Visibilité | Nom Du Colonne | Colonne Mère | Libellé | Type | Valeur Par Défaut | Read Only | Autres |
|-------------------------------------|----------------|----------------------------------|-----------|--------|-------------------|-------------------------------------|--------|
| <input type="checkbox"/> | ID | <input type="radio"/> | ID | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDMVT | <input checked="" type="radio"/> | IDMVT | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDARTICLE | <input type="radio"/> | IDARTICLE | String | | <input type="checkbox"/> | ... |
| <input type="checkbox"/> | ENTREE | <input type="radio"/> | ENTREE | String | 0 | <input checked="" type="checkbox"/> | ... |
| <input type="checkbox"/> | SORTIE | <input type="radio"/> | SORTIE | String | 0 | <input checked="" type="checkbox"/> | ... |

Voir 1 à 5 parmi 5 enregistrement

Previous1Next

Figure 14 : Interface pour générer un page de saisie multiple

Après génération et build, nous obtenons la page ci-dessous :

Saisie mouvement de stock

| | |
|-------------|-----------------|
| DATY | 24/10/2022 |
| IDMAGASIN | Dépôt Antsirabe |
| IDBL | |
| ETAT | 0 |
| IDTRANSFERT | |

| | ID | IDMVT | IDARTICLE | |
|--------------------------|----|-------|-----------|---|
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |

Figure 15 : Exemple de page de saisie multiple générée après un build

2.2.2.3 Générer une page de modification

Pour modifier des informations, nous avons besoin de formulaires de modification. La génération de pages de modification est semblable à celle d'une génération de pages de saisie.

Générer une Page de Modification à partir de : FACTUREFOURNISSEUR

Information de la Page

| | |
|---|---|
| Titre Modification Facturefournisseur | Page Après apresTarif.jsp |
| Lien test/ff/modification-ff.jsp | But <input type="checkbox"/> Effectuer Un Build Après Génération De La Page |
| Classe De Mapping (Avec Package) <input checked="" type="checkbox"/> stock.FactureFournisseur | |
| Nom De La Table Facturefournisseur | |

Champs

Voir 10 Enregistrement

| Visibilité | Nom Du Colonne | Libellé | Type | Valeur Par Défaut | Read Only | Autres |
|-------------------------------------|----------------|----------------|--------|-------------------|-------------------------------------|--------|
| <input checked="" type="checkbox"/> | ID | ID | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DATY | DATY | Date | 24/10/2022 | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | ETAT | ETAT | double | | <input checked="" type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | REF | REF | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDFOURNISSEUR | ID Fournisseur | String | | <input type="checkbox"/> | ... |

Voir 1 à 5 parmi 5 enregistrement

PreviousNext

Figure 16 : Interface pour générer un page de modification

Voici la page de modification que l'on obtient après :

Modification Facturefournisseur

| | |
|----------------|------------|
| Id | |
| DATY | 11/07/2022 |
| ETAT | 11 |
| REF | |
| ID FOURNISSEUR | FR000005 |

Réinitialiser Enregistrer

Figure 17 : Exemple de page de modification générée après un build

Pour les pages de modification multiple, le processus est le même que pour générer une page de saisie multiple.

Générer une Page de Modification Multiple à partir des tables : St_mvt_stock | St_mvt_stock_fille

Information de la Page

| | |
|---|--|
| <p>Titre Modification St_mvt_stock_st_mvt_stock_fille</p> <p>Lien test/mvtstock-modif.jsp</p> <p>Classe De Mapping Mère stock.MvtStock</p> <p>Nom De La Table Mère St_mvt_stock</p> | <p>Classe De Mapping Fille stock.MvtStockFille</p> <p>Nom De La Table Fille St_mvt_stock_fille</p> <p>Page Après apresTarif.jsp</p> <p>But</p> <p><input checked="" type="checkbox"/> Effectuer Un Build Après Génération De La Page</p> |
|---|--|

Champs table mère :

Voir 10 Enregistrement

| Visibilité | Nom Du Colonne | Libellé | Type | Valeur Par Défaut | Read Only | Autres |
|-------------------------------------|----------------|-------------|--------|-------------------|-------------------------------------|--------|
| <input checked="" type="checkbox"/> | ID | ID | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DATY | DATY | Date | 24/10/2022 | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DESIGNATION | DESIGNATION | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDMAGASIN | IDMAGASIN | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDBL | IDBL | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | ETAT | ETAT | double | | <input checked="" type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDTRANSFERT | IDTRANSFERT | String | | <input type="checkbox"/> | ... |

Voir 1 à 7 parmi 7 enregistrement

Champs table fille

Voir 1 à 5 parmi 5 enregistrement

| Visibilité | Nom Du Colonne | Colonne Mere | Libellé | Type | Valeur Par Défaut | Read Only | Autres |
|-------------------------------------|----------------|--------------|-----------|--------|-------------------|--------------------------|--------|
| <input checked="" type="checkbox"/> | ID | ○ | ID | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDMVT | ● | IDMVT | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDARTICLE | ○ | IDARTICLE | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | ENTREE | ○ | ENTREE | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | SORTIE | ○ | SORTIE | String | | <input type="checkbox"/> | ... |

Voir 1 à 5 parmi 5 enregistrement

Previous1Next

Previous1Next

Générer

Figure 18 : Interface pour générer un page de modification multiple

Voici la page de modification multiple générée :

Modification St_mvt_stock_st_mvt_stock_fille

| | |
|-------------|------------------|
| DATY | 11/07/2022 |
| DESIGNATION | Envoye Transfert |
| IDMAGASIN | SM0001 |
| IDBL | |
| ETAT | 11 |
| IDTRANSFERT | TS000126 |

| IDMVT | IDARTICLE | ENTREE | SORTIE | |
|------------------------------------|-----------|--------|--------|---|
| <input type="checkbox"/> SMS000349 | SA0001 | 0 | 100 | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |
| <input type="checkbox"/> | | | | ✕ |

Figure 19 : Exemple de page de modification multiple générée après un build

2.2.2.4 Générer une page de fiche

Pour connaître un élément de l'application, on a besoin de consulter sa fiche. Afin de générer une page de fiche, il faut entrer les éléments suivants : son titre, le lien qui mène à la page, la table et la classe mapping correspondante et le lien vers la page de modification de l'objet en question. Il faut aussi configurer les attributs de l'objet, choisir les attributs visibles et changer leur libellé.

Générer une Page de Fiche à partir de : FACTUREFOURNISSEUR_LIB

Informations sur la page

Titre
Fiche Facturefournisseur_lib

Lien
test/ff/fiche-facture-fournisseur.jsp

Classe De Mapping (Avec Package)
stock.FactureFournisseur

Nom De La Table
facturefournisseur_lib

Lien Vers La Page De Modification
Lien vers la page de modification(.jsp)

☒ Effectuer Un Build Après Génération De La Page

Champs

Voir 10 Enregistrement

| Nom Du Colonne | Type | Visibilité | Libellé |
|----------------|--------|-------------------------------------|----------------|
| DATY | Date | <input checked="" type="checkbox"/> | DATE |
| ETAT | double | <input checked="" type="checkbox"/> | ETAT |
| ID | String | <input checked="" type="checkbox"/> | ID |
| IDFOURNISSEUR | String | <input checked="" type="checkbox"/> | ID FOURNISSEUR |
| REF | String | <input checked="" type="checkbox"/> | REF |

Voir 1 à 5 parmi 5 enregistrement

[Générer](#)

Rechercher: [Previous](#)[Next](#)

Figure 20 : Interface pour générer un page de fiche

La page générée après est comme ci-dessous :

Fiche Facturefournisseur_lib

| | |
|----------------|------------|
| ID | FF000102 |
| DATE | 11/07/2022 |
| ETAT | VISÉ(E) |
| REF | |
| ID FOURNISSEUR | Laetitia |

[Modifier](#)

Figure 21 : Exemple de page de fiche générée après build

2.2.2.5 Générer une page d'analyse

L'analyse des données est très importante pour la gestion d'une société. Dans le but de générer une page d'analyse, il nous faut entrer les informations suivantes : le titre de la page, le lien menant vers la page, la table et la classe mapping correspondante, le nombre d'éléments à analyser. Il nous faut aussi choisir le type d'analyse soit une analyse simple, soit une analyse avec pourcentage ou encore une analyse croisée, on peut aussi choisir si on veut afficher le tableau de récapitulation de l'analyse.

Pour la configuration de l'analyse, on peut choisir les colonnes de groupement, et les colonnes à mettre en pourcentage, on peut aussi ajouter les liens pour chaque colonne.

Comme pour une page de liste, cette page comporte aussi une formulaire de recherche avancée qu'on peut paramétrer en ajoutant les options comme leurs libellés, leurs types: liste déroulante, intervalle ou popup, ou encore choisir les colonnes à sommer pour la récapitulation des données.

Générer une Page d'analyse à partir de : analyseproduction_lib

Informations de la page

Titre
Analyse des Analyseproduction_libs

Lien
test/analyse-prod.jsp

Classe De Mapping (Avec Package)
stock.Analyseprod

Nom De La Table
analyseproduction_lib

Nombre D'éléments Par Page
20

Type page analyse

☐ Page Simple
 ☒ Page Pourcentage
 ☐ Page CroisÃ©E

☒ Afficher Le Tableau De Récapitulation
 ☒ Effectuer Un Build Après Génération De La Page

Champs

Voir 10 Enregistrement

Rechercher:

| Nom Du Colonne | Entêtes | Colonne De Groupement | Colonne De Groupement Colonne | Libellé | Type | A Sommer | Pourcentage | Critère | Options Du Critère | Lien | Option Du Lien |
|----------------|-------------------------------------|-------------------------------------|-------------------------------|-----------|--------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------|-------------------------------------|----------------|
| DATY | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | DATY | Date | | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| ID | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ID | String | | | <input checked="" type="checkbox"/> | ... | <input checked="" type="checkbox"/> | ... |
| IDARTICLE | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | IDARTICLE | String | | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| IDMERE | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | IDMERE | String | | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| QTE | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | QTE | double | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| TYPE | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | TYPE | String | | | <input type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| UNITEPROD | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | UNITEPROD | String | | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |

Voir 1 à 7 parmi 7 enregistrement

Previous1Next

Figure 22 : Interface pour générer un page d'analyse

On obtiendra après build une page comme ceci :

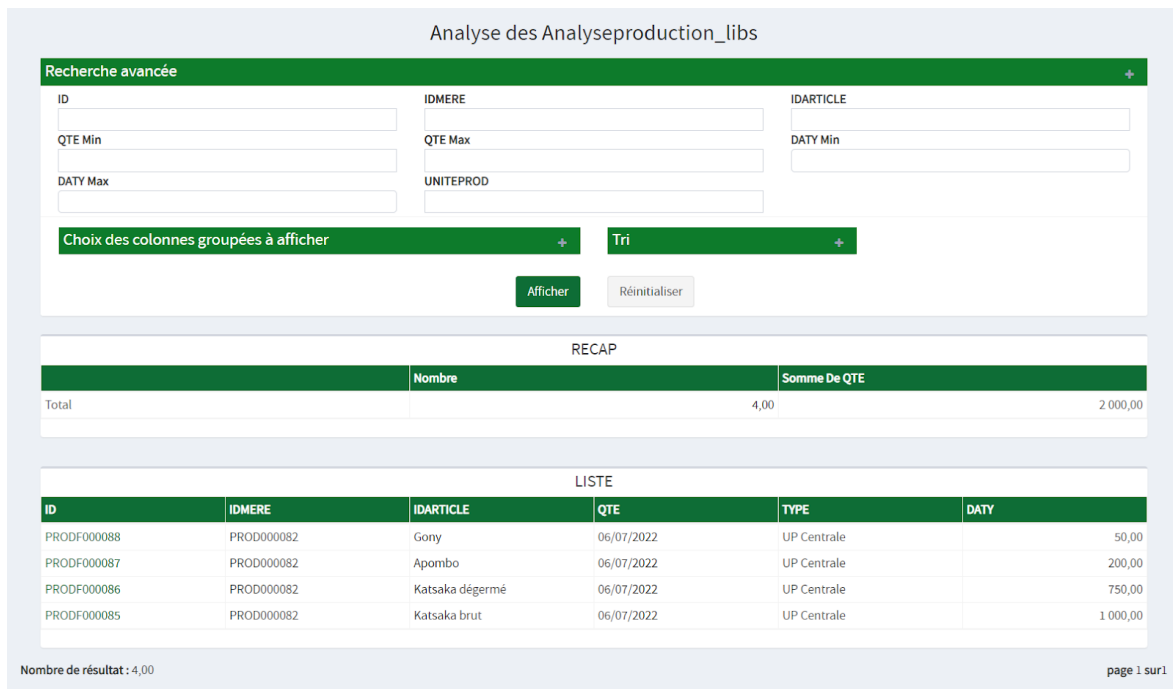


Figure 23 : Exemple de page d'analyse générée après un build

2.2.2.6 Générer une page de graphe

Les diagrammes et les graphiques sont des représentations visuelles des données. Ce sont des outils puissants pour l'analyse de données et facilitent leur compréhension. Afin de générer une page de graphe, il faut enregistrer les données suivantes : le titre de la page, son lien, la table et la classe de mapping correspondante, le type de graphique à générer comme : un diagramme en ligne, en bâton, en camembert ou en aire et sa description. Pour la configuration du graphe, on peut choisir les colonnes sur l'axe des X et les colonnes sur l'axe des Y et aussi les colonnes par défaut. Cette page comporte aussi une formulaire de recherche avancée qu'on peut paramétrer en ajoutant les options comme leurs libellés, leurs types: liste déroulante, intervalle ou popup, ou encore choisir les colonnes à sommer pour la récapitulation des données.

Générer une Page de Graphe à partir de : analyseproduction_lib

Information sur la page

Titre
Analyse Analyseproduction_lib

Lien
test/analyse-graph-prod.jsp

Classe De Mapping
stock.Analyseprod

☒
Nom De La Table
analyseproduction_lib

Type Bâton
Description Du Graphe
☒
Effectuer Un Build Après Génération De La Page

Champs

Voir 10 Enregistrement

Rechercher:

| Nom Du Colonne | Type | Col X | Col Y | Colonne Défaut | Colonne À Sommer | Critère | Options Du Critère |
|----------------|--------|----------------------------------|----------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------|
| DATY | Date | <input type="radio"/> | | <input checked="" type="checkbox"/> | | <input checked="" type="checkbox"/> | ... |
| ID | String | <input type="radio"/> | | <input type="checkbox"/> | | <input type="checkbox"/> | ... |
| IDARTICLE | String | <input checked="" type="radio"/> | | <input checked="" type="checkbox"/> | | <input checked="" type="checkbox"/> | ... |
| IDMERE | String | <input type="radio"/> | | <input type="checkbox"/> | | <input type="checkbox"/> | ... |
| QTE | double | <input type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ... |
| TYPE | String | <input type="radio"/> | | <input type="checkbox"/> | | <input type="checkbox"/> | ... |
| UNITEPROD | String | <input type="radio"/> | | <input checked="" type="checkbox"/> | | <input checked="" type="checkbox"/> | ... |

Voir 1 à 7 parmi 7 enregistrement

Générer

PreviousNext

Figure 24 : Interface pour générer une page de graphe

Voici la page générée après un build du projet :

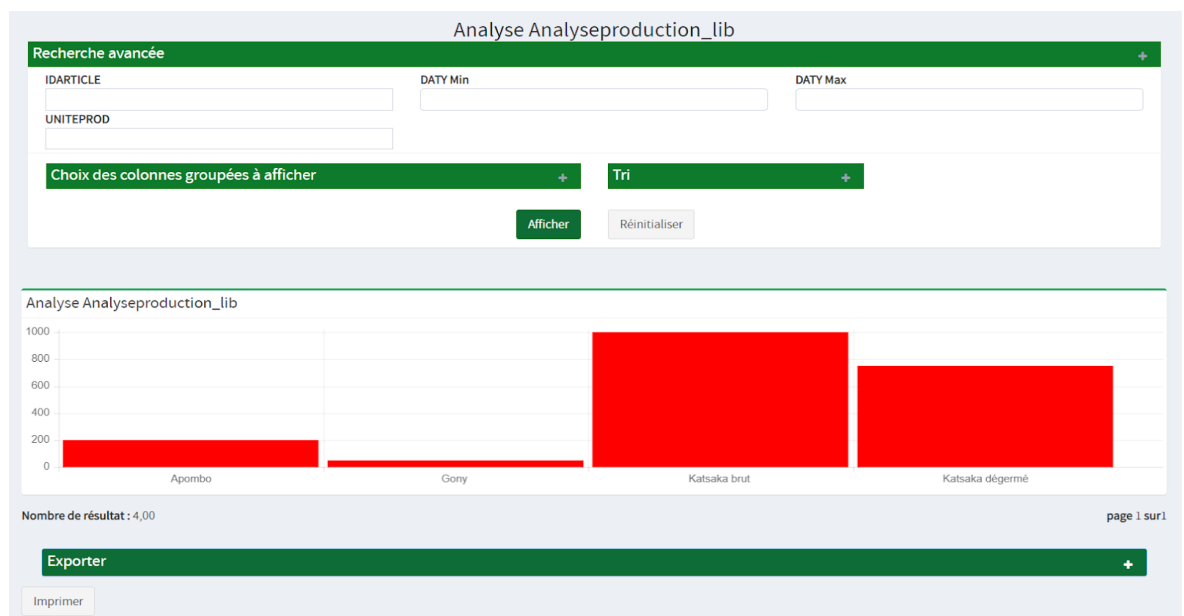


Figure 25 : Exemple de page de graphique après un build

2.2.3 Modification des pages

En cas de manque de paramètres ou de fausses manipulations, il est possible de modifier la page qu'on a généré. Après la génération d'une page, ces informations sont stockées dans la base de données du générateur, que l'on a la possibilité de modifier. Les paramètres de génération de la page sont stockés sous forme de JSON.

Il est donc plus facile de modifier une page car il nous faut juste récupérer les paramètres JSON et les injecter dans les formulaires grâce à la manipulation du DOM avec JQuery et AJAX. Ainsi, pour modifier une page, il suffit juste de consulter sa fiche et de cliquer sur "Modifier" comme suit :

Fiche Page

| | |
|----------|---------------------------------------|
| ID | CP0000000141123425 |
| PROJET | Production katsaka |
| LIEN | test/ff/liste-facture-fournisseur.jsp |
| TYPEPAGE | PageRecherche |
| TITRE | Liste des Facturefournisseur_libs |

ConsulterModifier

Figure 26 : Fiche d'une page générée

On arrive sur la page de modification de la page avec les mêmes paramètres qu'on a ajouté pendant sa génération :

Modifier une Page Recherche à partir de : facturefournisseur_lib

Informations de la page

Titre
Liste des Facturefournisseur_libs

Lien
test/ff/liste-facture-fournisseur.jsp

Classe De Mapping (Avec Package)
stock.FactureFournisseur

☒ **Nom De La Table**
facturefournisseur_lib

Nombre D'éléments Par Page
20

☐ **Afficher Le Tableau De Récapitulation**

☒ **Effectuer Un Build Après Génération De La Page**

Champs

Voir 10 Enregistrement

Rechercher:

| Entêtes | Nom Du Colonne | Libellé | Type | A Sommer | Critère | Options Du Critère | Lien | Option Du Lien |
|-------------------------------------|----------------|---------------|--------|-------------------------------------|-------------------------------------|--------------------|--------------------------|----------------|
| <input checked="" type="checkbox"/> | ID | ID | String | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DATY | DATY | Date | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | ETAT | ETAT | double | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | REF | REF | String | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | IDFOURNISSEUR | IDFOURNISSEUR | String | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |

Voir 1 à 5 parmi 5 enregistrement

[Previous](#)[Next](#)

[Modifier](#)

Figure 27 : Interface de modification d'une page

2.2.4 Exécution d'un build depuis le générateur

Afin que le changement soit visible au niveau du projet dans lequel on a généré ou modifié une page, il est impératif d'effectuer un build du projet. Pour ce faire, on a la possibilité d'exécuter un build depuis le générateur. On a donc le choix entre exécuter le build directement après avoir généré une page, que l'on va spécifier dans les paramètres de génération de la page, ou encore générer plusieurs pages et effectuer un build après.

Nom De La Table

mappingsimple

Lien Vers La Page De Modification

Lien vers la page de modification(.jsp)

☐

**effectuer un build après
génération de la page**

Figure 28 : Paramètre de build dans la génération d'une page

| Recherche avancée | | | |
|-------------------------------|--------|----------------|-----|
| ID | PROJET | LIEN | |
| TYPEPAGE | TITRE | BUILD | |
| Tous | | 0 | |
| Choix des colonnes à afficher | | Récapitulation | Tri |
| Afficher | | Réinitialiser | |

| LISTE | | | |
|--------------------|---------------------------------------|----------|--|
| ID | LIEN | TYPEPAGE | TITRE |
| CP0000000141123425 | test/ff/liste-facture-fournisseur.jsp | TPP0001 | Liste des Facturefournisseur_libs |
| CP0000000142121910 | test/mvtstock-modif.jsp | TPP0006 | Modification St_mvt_stock_st_mvt_stock_fille |
| CP0000000144222738 | test/ff/fiche-facture-fournisseur.jsp | TPP0002 | Fiche Facturefournisseur_lib |
| CP0000000145223425 | test/ff/modification-ff.jsp | TPP0003 | Modification Facturefournisseur |
| CP0000000143172239 | test/analyse-prod.jsp | TPP0004 | Analyse des Analyseproduction_libs |
| CP0000000146223819 | test/ff/saisie-ff.jsp | TPP0003 | Saisie Facturefournisseur |
| CP0000000045142427 | test/analyse-prod.jsp | TPP0004 | Liste des Analyseproduction_libs |

Effectuer un build

Figure 29 : Liste des pages qui n'ont pas encore été buildés dans le projet

Après l'exécution du build, on a la possibilité de voir les pages que l'on vient de générer depuis le générateur dans le projet concerné.

2.2.5 Génération des pages d'une application de type React JS

Le développement des applications à page unique comme avec React JS permet d'avoir une application plus performante et ayant une architecture avec des couches bien séparées. Ici la couche de présentation est développée en Javascript avec la librairie React JS et la couche logique est développée en Java avec le framework APJ. D'abord, il nous a fallu développer une API généralisée avec APJ afin que les 2 couches puissent communiquer entre elles et transmettre les données. Ensuite, pour la partie présentation, on a créé une application squelette généralisée adaptée à la structure des données du framework APJ qu'on pourra utiliser avec n'importe quel application APJ, et qui communiquera avec l'API généralisée. Dans l'application squelette servant de couche de présentation, nous avons développé les fonctionnalités généralisées suivantes :

- L'authentification pour gérer l'accès aux utilisateurs
- Le menu dynamique pour la navigation
- Les différents services pour communiquer avec l'API
- Les différents composants permettant de recevoir, afficher les données de l'API, et interagir avec ce dernier.

Il nous suffit donc juste de générer les pages où insérer ces fonctionnalités pour avoir une application fonctionnelle.

2.2.5.1 Un menu dynamique

Afin qu'on puisse naviguer dans l'application, on a développé un menu dynamique généralisé. Les données du menu sont stockées dans le cache de l'application et seront récupérées uniquement lors de la requête initiale.

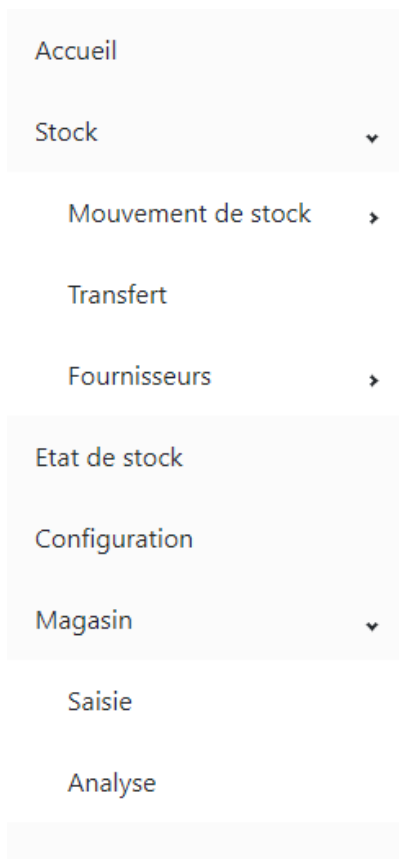


Figure 30 : Menu dynamique

2.2.5.2 Générer une page de liste

La génération d'une page de liste de type React JS se fait exactement comme pour une page de liste de type APIJ, nous avons aussi la possibilité d'exporter les données au format CSV ou format PDF. Voyons dans les figures ci-dessous la génération d'une page de liste :

Générer une Page Recherche à partir de : MAPPINGSIMPLE

Informations de la page

Titre
Liste des Mappingsimples

Lien
mappingsimple/saisie.js

Classe De Mapping (Avec Package)
test.MappingSimple

☒ **Nom De La Table**
mappingsimple

Nombre D'éléments Par Page
20

☐ **Afficher Le Tableau De Récapitulation**

Champs

| Entêtes | Nom Du Colonne | Libellé | Type | A Sommer | Critère | Options Du Critère | Lien | Option Du Lien |
|-------------------------------------|----------------|-------------|--------|-------------------------------------|-------------------------------------|--------------------|--------------------------|----------------|
| <input checked="" type="checkbox"/> | ID | ID | String | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DER | DER | String | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DATY | DATY | Date | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | NOMBREALEA | NOMBRE ALEA | double | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | REMARQUE | REMARQUE | String | | <input checked="" type="checkbox"/> | ... | <input type="checkbox"/> | ... |

Voir 1 à 5 parmi 5 enregistrement

Générer

Figure 31 : Interface pour générer une page de liste de type React

Liste des Mappingsimples

Recherche avancée

Filtres :

ID

DATY min
jj/mm/aaaa

DATY max
jj/mm/aaaa

NOMBREALEA min

NOMBREALEA max

REMARQUE

DER
Tous

Colonne à afficher

Récapitulation

Tri

Afficher Réinitialiser

| ID | DER | DATY | NOMBREALEA | REMARQUE |
|---------------------|---------|-------------------------|------------|---------------|
| MSP0000000063192801 | MPD0001 | 2022-11-12 00:00:00.000 | 22 | Rmq |
| MSP0000000062192725 | | 2022-11-12 00:00:00.000 | 11 | Mappingsimple |
| MSP0000000061192530 | MPD0001 | 2022-11-12 00:00:00.000 | 11 | |

Nombre de résultats : 3

Exporter

Export CSV

Export PDF

Figure 32 : Exemple de page React de liste générée

2.2.5.3 Générer une page de fiche

Voyons dans les figures ci-dessous la génération d'une page de fiche :

Générer une Page de Fiche à partir de : mappingsimple

Informations sur la page

Titre

Fiche Mappingsimple

Nom De La Table

mappingsimple

Lien

test/mappingsimple/id.js

Lien Vers La Page De Modification

test/mappingsimple/modif/id.js

Classe De Mapping (Avec Package)

test.MappingSimple

Champs

Voir 10 Enregistrement

Rechercher:

| Nom Du Colonne | Type | Visibilité | Libellé |
|----------------|--------|-------------------------------------|------------|
| DATY | Date | <input checked="" type="checkbox"/> | DATY |
| DER | String | <input checked="" type="checkbox"/> | DER |
| ID | String | <input checked="" type="checkbox"/> | ID |
| NOMBREALEA | double | <input checked="" type="checkbox"/> | NOMBREALEA |
| REMARQUE | String | <input checked="" type="checkbox"/> | REMARQUE |

Voir 1 à 5 parmi 5 enregistrement

PreviousNext

Générer

On obtiendra alors une page comme celui-ci :

Fiche Mappingsimple

| | |
|--------------------|-------------------------|
| ID | MSP0002 |
| DER | |
| DATY | 2022-11-12 00:00:00.000 |
| NOMBRE ALEA | 11 |
| REMARQUE | Mappingsimple |

Modifier

2.2.5.4 Générer une page de saisie et modification

Voyons dans les figures ci-dessous la génération d'une page de saisie :

Générer une Page de Saisie à partir de : MAPPINGSIMPLE

Information de la Page

Titre
Saisie Mappingsimple

Lien
mappingsimple/saisie.js

Classe De Mapping (Avec Package)
test.Mappingsimple

Nom De La Table
Mappingsimple

But
mappingsimple/liste.js

Champs

Voir 10 Enregistrement

Rechercher:

| Visibilité | Nom Du Colonne | Libellé | Type | Valeur Par Défaut | Read Only | Autres |
|-------------------------------------|----------------|-------------|--------|-------------------|--------------------------|--------|
| <input checked="" type="checkbox"/> | ID | ID | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DER | DER | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DATY | DATY | Date | 06/11/2022 | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | NOMBREAlea | NOMBR EALEA | double | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | REMARQUE | REMARQUE | String | | <input type="checkbox"/> | ... |

Voir 1 à 5 parmi 5 enregistrement

[Previous](#) [Next](#)

[Générer](#)

Figure 33 : Interface pour générer une page de saisie

Saisie Mappingsimple

DATY

NOMBRE
ALEA

REMARQUE

DER

Insérer

Réinitialiser

Figure 34 : Page de saisie générée

Générer une Page de Modification à partir de : Mappingsimple

Information de la Page

Titre
Modification Mappingsimple

But
mappingsimple/liste.js

Lien
mappingsimple/saisie.js

Classe De Mapping (Avec Package)
test.MappingSimple

Nom De La Table
Mappingsimple

Champs
Voir 10 Enregistrement

Rechercher:

| Visibilité | Nom Du Colonne | Libellé | Type | Valeur Par Défaut | Read Only | Autres |
|-------------------------------------|----------------|-------------|--------|-------------------|-------------------------------------|--------|
| <input type="checkbox"/> | ID | ID | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DER | DER | String | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | DATY | DATY | Date | 06 / 11 / 2022 | <input checked="" type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | NOMBREALEA | NOMBRE ALEA | double | | <input type="checkbox"/> | ... |
| <input checked="" type="checkbox"/> | REMARQUE | REMARQUE | String | | <input type="checkbox"/> | ... |

Voir 1 à 5 parmi 5 enregistrement

PreviousNext

Générer

Modification Mappingsimple

DER

DATY

12/11/2022

NOMBRE
ALEA

11

REMARQUE

Mappingsimple

Modifier

Réinitialiser

2.2.5.5 Générer une page de graph

La génération des pages de graph se fait suivant les illustrations ci-dessous :

Générer une Page de Graphe à partir de : analyseproduction_lib

Information sur la page

Titre
Analyse Analyseproduction_lib

Lien
test/graph/graph.js

Classe De Mapping
stock.Analyseprod

Nom De La Table
analyseproduction_lib

Type Baton

Description Du Graphe

Champs
Voir 10 Enregistrement

Rechercher:

| Nom Du Colonne | Type | Col X | Col Y | Colonne Défaut | Colonne À Sommer | Critère | Options Du Critère |
|----------------|--------|----------------------------------|----------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------|
| DATY | Date | <input type="radio"/> | | <input checked="" type="checkbox"/> | | <input checked="" type="checkbox"/> | ... |
| ID | String | <input type="radio"/> | | <input type="checkbox"/> | | <input type="checkbox"/> | ... |
| IDARTICLE | String | <input checked="" type="radio"/> | | <input checked="" type="checkbox"/> | | <input checked="" type="checkbox"/> | ... |
| IDMERE | String | <input type="radio"/> | | <input type="checkbox"/> | | <input type="checkbox"/> | ... |
| QTE | double | <input type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ... |
| TYPE | String | <input type="radio"/> | | <input type="checkbox"/> | | <input type="checkbox"/> | ... |
| UNITEPROD | String | <input type="radio"/> | | <input checked="" type="checkbox"/> | | <input checked="" type="checkbox"/> | ... |

Voir 1 à 7 parmi 7 enregistrement

Générer
PreviousNext

Figure 35 : Interface pour générer une page de graph

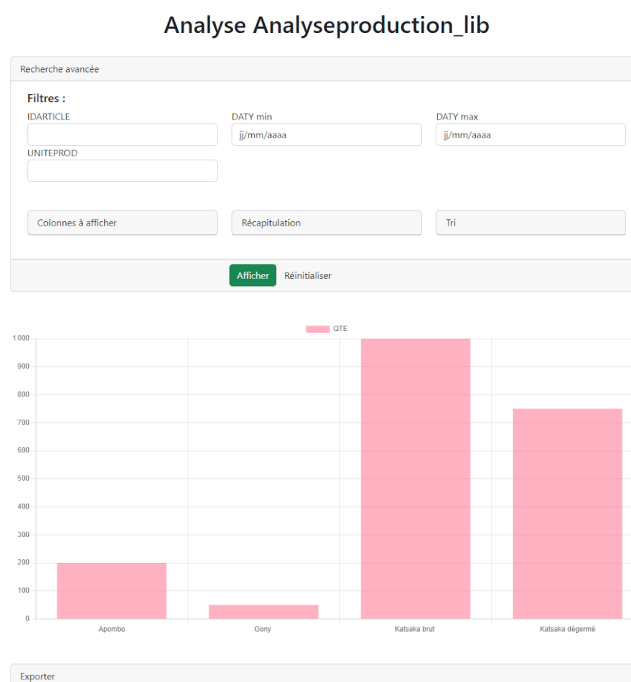


Figure 36 : Page de graph générée

2.3 État d'Analyse et Statistiques

2.3.1 Analyse des pages générées

On a la possibilité de consulter l'analyse des pages générées par projet et par type de page comme le montre la figure ci-contre :

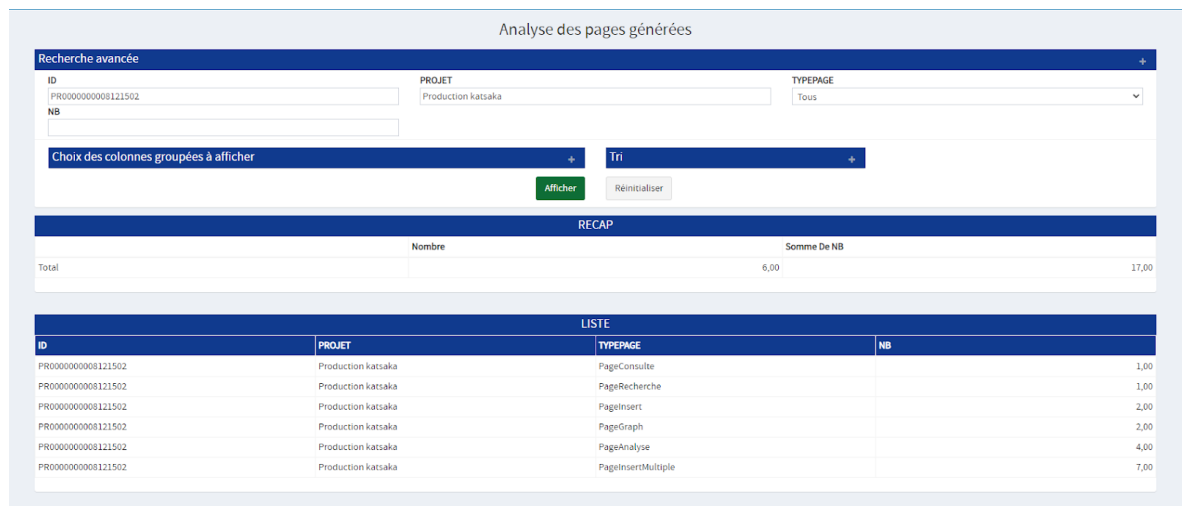


Figure 37 : Analyse des pages générées par projet et par type de page

2.3.2 Analyse du temps de développement des pages

En utilisant le générateur pour le développement d'un projet en cours, avec 15 pages, et en comparant avec la méthode de développement classique, voici le résultat de l'analyse :

Tableau 4 : Analyse du temps de développement

| Nombre de pages | Temps moyen en utilisant la méthode classique | Temps moyen en utilisant le générateur | Gain de temps |
|-----------------|---|--|---------------|
| 15 | 15 minutes | 3 minutes | 3h 15 min |

On a donc pu gagner à peu près 3h 15 min, sans compter les risques de bugs si on avait utilisé la méthode classique. Cela représente un avantage considérable en matière de temps de développement. L'entreprise gagnera alors plus de temps à délivrer ces applications, et les développeurs pourront se concentrer sur les tâches plus complexes.

2.4 Problèmes rencontrés et solutions

Pour générer une page, il faut écrire une longue chaîne de caractères dans le fichier de la page. On a utilisé la concaténation de chaîne de caractères avec les variables, dont la quantité à concaténer avec la chaîne est assez importante. Pour lire ou modifier le contenu de la page, on peut facilement se perdre car le code est assez difficile à lire.

On a donc opté pour une autre alternative, l'utilisation des chaînes de caractères formatées, c'est-à-dire le code du contenu d'une page sera chargé à partir d'un fichier de ressources ou template. Cette méthode permet de déterminer facilement à quoi ressemble la chaîne rendue parce que c'est une chaîne continue avec des espaces réservées pour injecter les variables.

3 Évaluation du projet et connaissances acquises

3.1 Bilan pour l'entreprise

La réalisation du framework représente un avantage pour la société BICI car la génération des pages avec l'intégration du build permettra aux développeurs de gagner beaucoup de temps pour développer les applications au sein de l'entreprise. Ainsi, ils mettront beaucoup moins de temps pour livrer leurs applications et peuvent se concentrer plus sur les développements plus complexes.

3.2 Bilan personnel

Sur le plan personnel, ce stage a été une expérience pleine d'expérience et la mise en pratique des savoirs acquis pendant ma formation en "Développeur d'application". J'ai eu la chance de travailler sur plusieurs projets de la société, notamment le développement et l'intégration du site de JEJOO. Grâce à ce stage j'ai développé mes capacités organisationnelles, la conception et développement d'applications web suivant une architecture bien définie et l'intégration et déploiement continue d'une application avec les outils comme Gitlab CI/CD. J'ai aussi appris à emballer et lancer des applications dans un conteneur avec Docker et l'utilisation d'un système de contrôle de version centralisé comme SVN. Enfin, ce stage a été pour moi une confirmation du choix de carrière et un commencement dans le monde de l'informatique professionnel.

3.3 Extension et évolution de l'application

Actuellement, l'application prend en charge les interfaces API et React JS. Puisqu'on a déjà développé une API généralisée, on a la possibilité de l'utiliser pour communiquer avec tout type d'applications. Avec le générateur, on pourra donc générer d'autres types d'applications qui pourra communiquer avec l'API généralisée comme :

- Une application web de type SPA différente de React JS, comme Angular ou Vue JS
- Une application mobile
- Une application desktop

● Conclusion

Pour conclure, j'ai réalisé mon stage en tant que développeur d'application chez BICI Madagascar. Pendant ce stage de 3 mois, j'ai pu mettre en pratique mes connaissances en développement acquises durant ma formation à l'IT University, tout en étant confronté aux difficultés réelles du monde du développement professionnel. Les objectifs fixés ont été atteints à savoir la réalisation du projet “**Framework générateur d'interfaces d'une application avec APJ**” dont le point principal à retenir est la génération des interfaces d'une application afin d'accélérer et simplifier son développement.

Sur le plan personnel, ce stage m'a permis de découvrir le métier de développeur. Il m'a permis d'avoir conscience des exigences pour exercer le métier, à savoir, écrire un code bien structuré, toujours effectué les tests, de savoir gérer son temps et de suivre les bonnes pratiques de développement quel que soit le langage de programmation utilisé.

En résumé, ce stage a été une expérience excellente et enrichissante et les objectifs fixés ont été atteints. Il faut reconnaître que les aspects techniques du travail que j'ai effectué ne sont pas sans failles et pourraient être améliorés en ajoutant la génération des interfaces d'autres types d'applications comme les applications mobiles ou desktop qui vont communiquer avec l'API généralisé.

● Bibliographie

Jean-Michel DOUDOUX. *J2EE / Java EE*. Disponible sur :

<https://www.jmdoudoux.fr/java/dej/chap-j2ee-javace.htm> (consultée le 06-06-2022)

W3 School, *jQuery - AJAX get() and post() Methods*. Disponible sur :

https://www.w3schools.com/jquery/jquery_ajax_get_post.asp (consultée le 09-06-2022)

Chrysanthus Forcha, *How to Read From a Local File in Java*. Disponible sur :

<https://linuxhint.com/how-to-read-from-local-file-in-java/> (consultée le 17-06-2022)

Baeldung. *How to Read a File in Java*. Disponible sur :

<https://www.baeldung.com/reading-file-in-java> (consultée le 17-06-2022)

Canberk Akduygu. *Files API in JAVA 8*. Disponible sur :

<https://www.swtestacademy.com/files-api-in-java-8/> (consultée le 30-06-2022)

Jakob Jenkov. *Java NIO Path*. Disponible sur:

<https://jenkov.com/tutorials/java-nio/path.htm> (consultée le 30-06-2022)

Oracletutorial.com. *Oracle Subquery*. Disponible sur:

<https://www.oracletutorial.com/oracle-basics/oracle-subquery/> (consultée le 20-07-2022)

Shaharyar Lalani, *How To Format a String In Java With Examples*. Disponible sur :

<https://xperti.io/blogs/how-to-format-string-in-java> (consultée le 02-08-2022)

Mike Wasson, *Single-Page Applications: Build Modern, Responsive Web Apps*. Disponible sur :

<https://learn.microsoft.com/en-us/archive/msdn-magazine/2013/november/asp-net-single-page-applications-build-modern-responsive-web-apps-with-asp-net> (consultée le 19-08-2022)

Amit Phaujdar, *PostgreSQL vs Oracle: 6 Critical Differences*. Disponible sur :

<https://hevodata.com/learn/postgresql-vs-oracle/> (consultée le 23-08-2022)

Vijay Muthu, *PostgreSQL vs. Oracle: Let's Compare*. Disponible sur :

<https://www.datavail.com/blog/postgresql-vs-oracle-lets-compare/> (consultée le 23-08-2022)

Dawid Karczewski, *PHP vs Java – The Best Choice For Web Development*. Disponible sur :

<https://www.ideamotive.co/blog/php-vs-java-the-best-choice-for-web-development> (consultée le 25-08-2022)

Nihar Raval, *React vs Angular: Which JS Framework to Pick for Front-end Development?*.

Disponible sur : <https://radixweb.com/blog/react-vs-angular> (consultée le 15-09-2022)

● Annexe

1. Exemple de structure JSON généré et stocké après avoir généré une page :

```
"data" : {
  "action" : "liste",
  "build" : "false",
  "classe" : "stock.FactureFournisseur",
  ⊖ "filles" : [
    ⊖ "0" : {
      "crt" : "true",
      "crt_lib" : "ID",
      "der_affiche" : "",
      "der_classe" : "",
      "der_table" : "",
      "der_valeur" : "",
      "isLink" : "false",
      "lib_max" : "ID max",
      "lib_min" : "ID min",
      "libelle" : "ID",
      "link" : "",
      "nom" : "ID",
      "pop" : "",
      "type" : "libre",
      "visibilite" : "true"
    },
    ⊕ "1" : { ... },
    ⊕ "2" : { ... },
    ⊕ "3" : { ... },
    ⊕ "4" : { ... }
  ],
  "lien" : "test/index.js",
  "nomtable" : "facturefournisseur_lib",
  "npp" : "20",
  "recap_show" : "false",
  "titre" : "Liste des Facturefournisseur_libs"
},
```