



BTS SIO – Option SLAM
Documentation d'épreuve

**E4 - Mission 8 : Moteur de
récolte de ImmoZia**

Par Baptiste Grimaldi



E4-Mission8-Moteur-collecte-immobilier-Baptiste-Grimaldi

Liens utiles

Article Un moteur de collecte de données

Check out the portfolio of Baptiste Grimaldi, a full-stack web developer specializing in creating dynamic and responsive websites. Explore his projects and contact him for your next web development project.

 <https://portfolio.baptistegrimaldi.info/projects/view/moteur-immozia>

1. Cahier des charges

1.1 Introduction

Lors de ma deuxième année de BTS SIO en alternance, j'ai été amené à accompagner la création d'une mission phare de l'entreprise Oxianet : la création d'un moteur de collecte de données immobilières. Mais

Cette moteur a été codée en PHP vanilla, c'est-à-dire qu'aucun framework n'a été utilisé. Seule une librairie de gestion de date a récemment été implémentée par mes soins.



La base de donnée résultante de ce projet est décrite dans un autre de mes projets:

Modélisation de BDD pour ImmoZia



Certaines fonctionnalités ou particularités de mes algorithmes sont la propriété de l'entreprise, je dois donc omettre certaines parties. Cela n'empêchera pas la clarté de mes propos ni la compréhension globale de cette réalisation.

1.2 Expression fonctionnelle du besoin

Oxianet souhaite se développer dans le milieu de l'immobilier. Pour cela, on m'a confié la création d'un site web de type "agrégateur immobilier" : ImmoZia.

Pour m'aider dans ce grand projet, l'entreprise et moi avons recruté un autre alternant avec qui je travaillerai tout au long de cette année.



Un agrégateur de bien immobilier est **une personne qui réunit en un seul site web plusieurs offres immobilières**. En effet, ils servent à indexer les offres de ventes ou offres de location.

([Lien de source](#))

Dans cette mission, je me concentre sur la partie de récolte et stockage des données. La partie de traitement des données est encadrée par mon collègue.

1.3 Contraintes

- La plus grande contrainte de ce projet est **la taille phénoménale de toutes les données** données immobilières mises à disposition des différents sites internet.

La quantité de données immobilières est immense, ce qui rend la collecte et le stockage de ces données très compliqués. Nous avons dû mettre en place des stratégies pour optimiser la collecte et éviter les doublons dans la base de données. Nous avons également dû éviter les doublons entre les sites, ainsi que les doublons résultant de la republication d'une annonce avec de nouvelles informations.

- En deuxième position vient **la rapidité d'exécution**, qui est un élément crucial pour de nombreuses entreprises qui cherchent à maximiser leur efficacité et leur rentabilité. En effet, une exécution rapide des tâches permet non seulement de gagner du temps, mais également d'améliorer la satisfaction des clients en

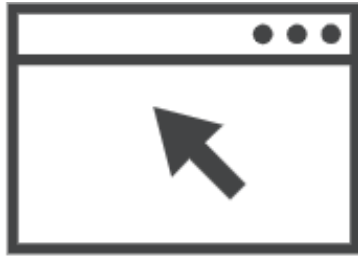
répondant rapidement à leurs demandes. De plus, cela peut permettre d'avoir une longueur d'avance sur la concurrence en étant capable de livrer des produits ou des services plus rapidement.

- Lors de la création d'un outil qui parcourt Internet, il faut prendre en compte le fait que lors de la mise à l'échelle, **tous les petits coûts deviennent énormes**. Il est donc nécessaire de se débrouiller autant que possible sans services et d'optimiser autant que possible notre code afin de **réduire les besoins en ressources**.

2. Description des environnements

Un environnement de récupération de donnée est constitué de plusieurs parties. Voici les trois plus importantes:

- **Passerelle** de récupération de données.
- **Analyse** rapide de la donnée.
- **Distribution** des données entre les points de stockages.



Source de données immobilières



Passerelle

La passerelle

La passerelle est un terme général signifiant point de sortie ou d'entrée. Dans ce domaine d'application, la passerelle peut être un Proxy, une API tierce qui s'occupe de certains paramètres, exécute ma requête passée en paramètres puis nous envoie son retour.



Moteur

Passerelle de récupération de données

Pour récupérer des données sur des sites internet, avec le temps, j'ai trouvé trois méthodes principales:

- **Récupération de toute la page HTML** et extraction des informations à l'aide de sélecteurs CSS. Cette méthode donne accès à tout ce qui est visible directement pour l'utilisateur.

- **Récupération d'un fichier de stockage** dans la page pour les pages dont les premières données sont envoyées avec la page principale. En effet, sur certains sites Internet, on trouve un objet JSON si on cherche bien, directement dans la page HTML, qui contient les informations des annonces affichées sur la page. Souvent, des informations plus précises sont disponibles, même celles qui ne sont pas visibles par l'utilisateur.
- **Rendu de la page dans un "headless browser"** et simulation de clics utilisateurs pour accéder aux données voulues. Cette méthode donne accès aux mêmes informations que la deuxième méthode. Cette méthode est souvent la plus fiable.



Un **headless browser** est un navigateur web qui peut être utilisé sans interface graphique. Cela signifie que les utilisateurs peuvent utiliser des scripts pour automatiser des tâches telles que la navigation sur des sites web, le remplissage de formulaires et le téléchargement de fichiers. Les headless browsers sont souvent utilisés dans le développement web pour tester des sites web et des applications web, mais ils peuvent également être utilisés dans des tâches de récupération de données où des scripts doivent naviguer sur des sites web pour extraire des données.

Afin d'assurer la récupération et la fiabilité des données, les trois méthodes sont utilisées selon les différences sources de données.

3. Méthodologie

3. Méthodologie

En tant que développeur web fullstack, l'adoption de méthodologies et de rigueur est d'une importance capitale. Il est essentiel de choisir une stratégie de versioning appropriée pour le projet en question, ainsi que pour la gestion des tests et de la documentation.

3.1 Méthodologie et versioning

Bonnes pratiques

- ne committez que des choses sur les mêmes sujets (style, front, back, etc...)
- Si vous ne pouvez pas écrire de messages de commit concis, cela indique trop de sujets dans le même commit.
- Utilisez un titre et un corps avec seulement la commande commit et en ajoutant une ligne vide entre le titre et le corps.

```
git commit
```

```
::
```

```
TITRE
```

```
CORPS
```

```
\# rest comments
```

Stratégies de branching

- Une convention écrite pour organiser l'équipe.

Deux options principales:

1. Développement **Mainline**:

- **quelques** branches
- commits relativement petits
- normes de test de haute qualité
-

2. Branches **State**, **Release** et **Feature**

- **Deux** types de branches différents qui remplissent des types de travail différents

1. **LongRunning**

- existe tout au long de la vie du projet
- souvent, ils reflètent les "étapes" de votre cycle de vie de développement

2. **Short Running**

- pour les nouvelles fonctionnalités, les corrections de bogues, le refactoring, les expériences
- sera supprimé après l'intégration (fusion/rebase)

Deux exemples de stratégies de branching

1. GitHub Flow

- très simple, très léger : seulement long-running
- branche ("main") + branches de fonctionnalités

2. GitFlow

- plus de structure, plus de règles
- long-running : "main" + "develop"
- de courte durée : fonctionnalités, versions, correctifs

Pour la gestion de version de ce projet, nous avons opté pour une stratégie de branching avec un main, un dev et des branches pour les fonctionnalités. Cette méthode de versioning permet de travailler sur plusieurs fonctionnalités en parallèle tout en minimisant les conflits de code.

L'utilisation de branches pour les fonctionnalités permet de travailler sur des fonctionnalités spécifiques de manière isolée, sans perturber le code principal de l'application. Chaque branche est associée à une fonctionnalité spécifique et est testée avant d'être fusionnée dans la branche de développement (dev).

La branche dev est la branche principale sur laquelle l'équipe travaille. Les fonctionnalités développées dans des branches distinctes sont fusionnées dans la branche dev une fois qu'elles ont été testées et validées. Cela permet de s'assurer que toutes les fonctionnalités sont en mesure d'interagir correctement les unes avec les autres.

En outre, toutes les fonctionnalités sont vérifiées par une série de tests documentés ainsi que par d'autres développeurs, afin de s'assurer qu'elles répondent aux exigences spécifiées dans le cahier des charges. Cette approche contribue à minimiser les erreurs et à garantir la qualité de l'interface web.

En résumé, la stratégie de versioning avec un main, un dev et des branches pour les fonctionnalités est efficace pour travailler sur plusieurs fonctionnalités en parallèle tout en minimisant les conflits de code. Les fonctionnalités sont testées avant d'être

fusionnées dans la branche principale, ce qui permet de garantir la qualité de l'interface web.



3.2 Gestion des tests de la solution



Le test est une phase importante dans le développement web car il permet de s'assurer que le site web ou l'application web fonctionne correctement et répond aux exigences et spécifications. Pendant la phase de test, les développeurs et les testeurs identifient et corrigent les défauts ou bugs dans le code, et vérifient que le site web ou l'application web fonctionne comme prévu.

Dans ce projet, les tests sont particulièrement importants. Lorsque nous récupérons des données de manière instable, la source peut changer sa structure à tout moment pendant le processus de développement. Cela m'est déjà arrivé plusieurs fois et pour éviter de coder dans le vide, il est préférable de mettre en place des tests continus.

Mise en place de tests

Afin d'assurer des tests continus, j'ai développé une page de tests ainsi qu'une route de test.

- **Page de test :**

- Cette page contient toutes les fonctionnalités sur lesquelles je travaille actuellement. Je sélectionne la fonctionnalité que je souhaite tester et elle s'exécute. J'utilise cette page à chaque phase de code et après chaque intégration.
- **Route de tests :**
 - Lorsque nous travaillons avec de faux navigateurs, il est souvent important, voire nécessaire, de passer certains paramètres supplémentaires dans une requête (en-têtes personnalisés, cookies, etc.). Ces données ne sont pas toujours faciles à visualiser clairement. J'ai donc créé une route de test qui accepte la plupart des types de requêtes et affiche toutes les données pertinentes dans la requête que je viens d'effectuer.

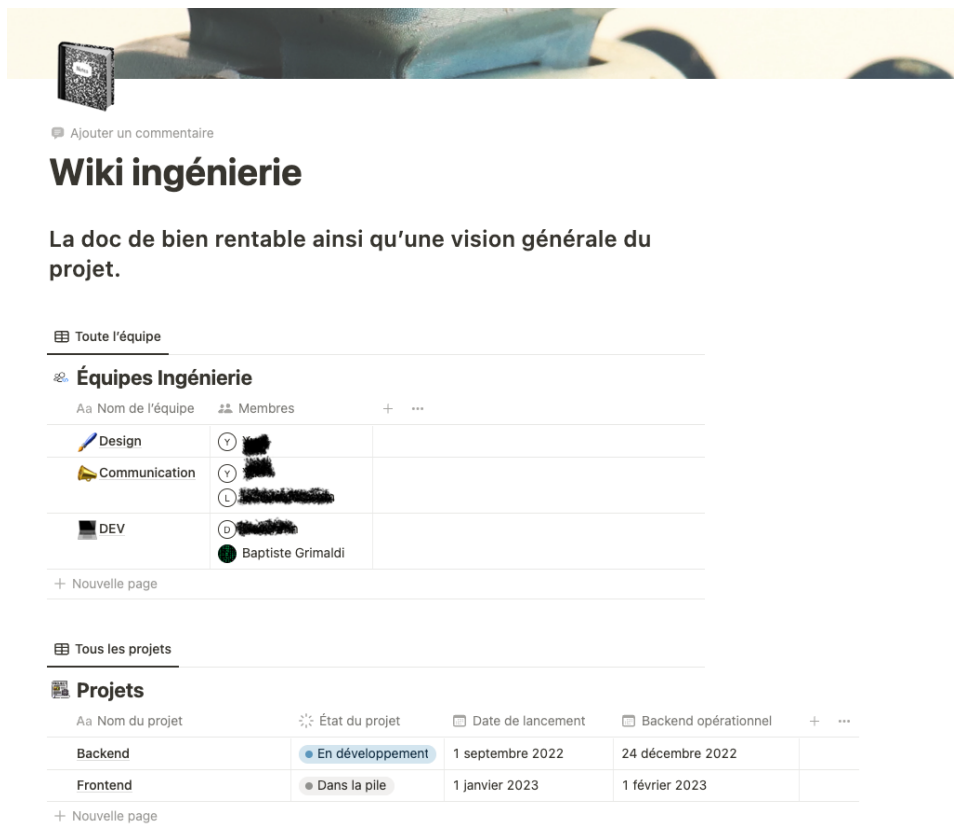
TRASH



collaboration qui permet à chaque membre de l'équipe de contribuer à la documentation et de la mettre à jour au fur et à mesure que le projet avance. La documentation est rédigée en utilisant une syntaxe de balisage simple pour faciliter la lecture et la compréhension.

La documentation inclut également des captures d'écran et des exemples de code pour illustrer les fonctionnalités développées et les modifications apportées. Cela permet à l'équipe de développement de comprendre rapidement les changements apportés au code et de voir comment les nouvelles fonctionnalités ont été implémentées.

En somme, la rédaction de la documentation est une étape importante du processus de développement. Elle permet à l'équipe de développement de se tenir informée des modifications apportées au code et des nouvelles fonctionnalités développées, et de garantir que toutes les informations pertinentes sont stockées dans un endroit centralisé et facilement accessible.



The screenshot displays the 'Wiki ingénierie' interface. At the top, there's a header with a small icon and a link to 'Ajouter un commentaire'. Below this is the title 'Wiki ingénierie' and a subtitle 'La doc de bien rentable ainsi qu'une vision générale du projet.'.

The main content is divided into two sections: 'Toute l'équipe' and 'Tous les projets'.

Toute l'équipe

Équipes Ingénierie

Aa	Nom de l'équipe	Membres		
	Design	(Y) [Avatar]		
	Communication	(Y) [Avatar] (L) [Avatar]		
	DEV	(D) [Avatar] (G) Baptiste Grimaldi		

+ Nouvelle page

Tous les projets

Projets

Aa	Nom du projet	État du projet	Date de lancement	Backend opérationnel	
	Backend	En développement	1 septembre 2022	24 décembre 2022	
	Frontend	Dans la pile	1 janvier 2023	1 février 2023	

+ Nouvelle page

Extrait du wiki-ingénierie

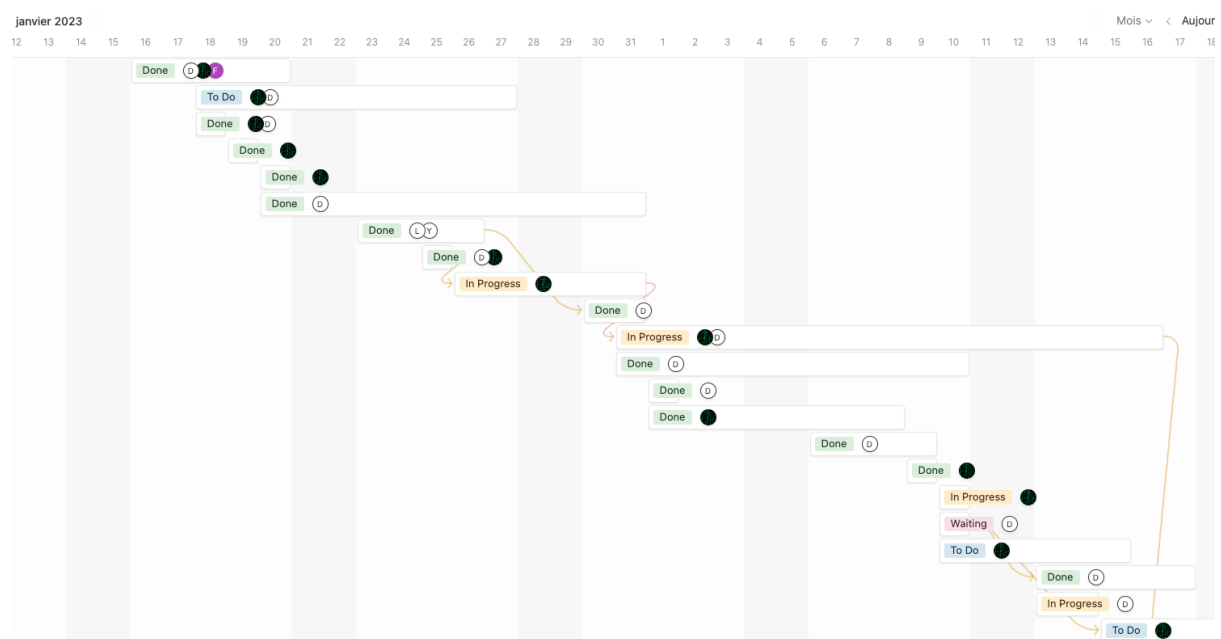
3.4 Gestion de projet

Pour la gestion du projet, l'équipe utilise Notion pour stocker une base de données de tâches. Cette base de données est utilisée pour suivre l'avancement du projet et pour assigner des tâches à différents membres de l'équipe.

Notion permet également d'utiliser différentes visualisations pour la gestion de projet, telles que le tableau Kanban ou le diagramme de Gantt. Le tableau Kanban est utilisé pour suivre l'avancement des tâches, tandis que le diagramme de Gantt est utilisé pour visualiser les dépendances entre les tâches et les jalons du projet.

En utilisant Notion pour la gestion de projet, l'équipe peut travailler de manière collaborative, en temps réel, et avoir une vue d'ensemble de l'avancement du projet. Les mises à jour sont synchronisées en temps réel, ce qui permet à chaque membre de l'équipe de rester informé des dernières modifications apportées au projet.

En somme, Notion est un outil de gestion de projet efficace pour l'équipe de développement d'Immozia. Il permet de stocker une base de données de tâches, de suivre l'avancement du projet et d'utiliser différentes visualisations pour la gestion de projet. En utilisant Notion, l'équipe peut travailler de manière collaborative et avoir une vue d'ensemble de l'avancement du projet.





Au bureau des développeurs, nous avons également un grand tableau que j'utilise très souvent pour expliquer les prochains aspects du projet. Je trouve que le schéma est la meilleure façon d'expliquer quelque chose de technique. Cela force à expliquer de façon simple avec des traits simples.

4. Mise en oeuvre

4.1 Recherche des sources

Lorsque je recherche des sources de données immobilières, j'ai toujours tendance à privilégier les solutions open source. En effet, non seulement ce choix me permet d'avoir un accès direct au code, mais cela correspond également à une conviction personnelle. Si deux projets sont disponibles et que l'un d'entre eux propose de mettre son code en ligne pour permettre aux autres contributeurs de l'améliorer, cela signifie qu'il est motivé par une réelle volonté de collaboration et d'innovation. Dans ce cas, je préfère choisir cette solution car elle reflète davantage mes valeurs et mes préférences en matière de développement de logiciels.



Les sources de ce projet doivent rester confidentielles car elles sont l'essence même du projet. La valeur ajoutée d'ImmoZia réside dans la recherche de sources viables. Si je partage ces listes, nous perdrons cette valeur ajoutée.

A titre d'exemple, à droite est une capture d'écran du processus utilisé pour collecter et organiser les sources pour le projet.

Voici quelques exemples:

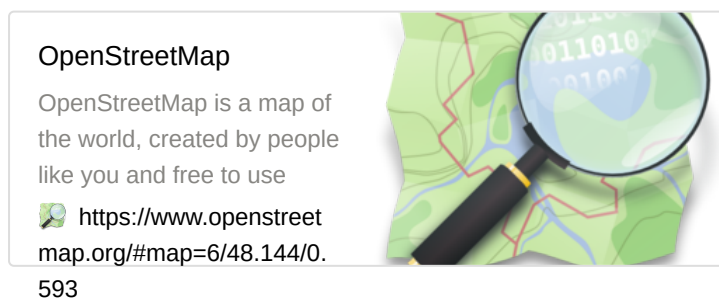
Petites annonces immobilières | 1er site immobilier français | Portail immo

Trouvez un appartement ou une maison en vente ou en location. Une offre de plus de 1 208 631 annonces immobilières actualisée en temps réel de logements en france.

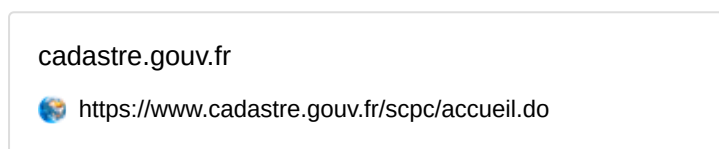
 <https://www.seloger.com/>



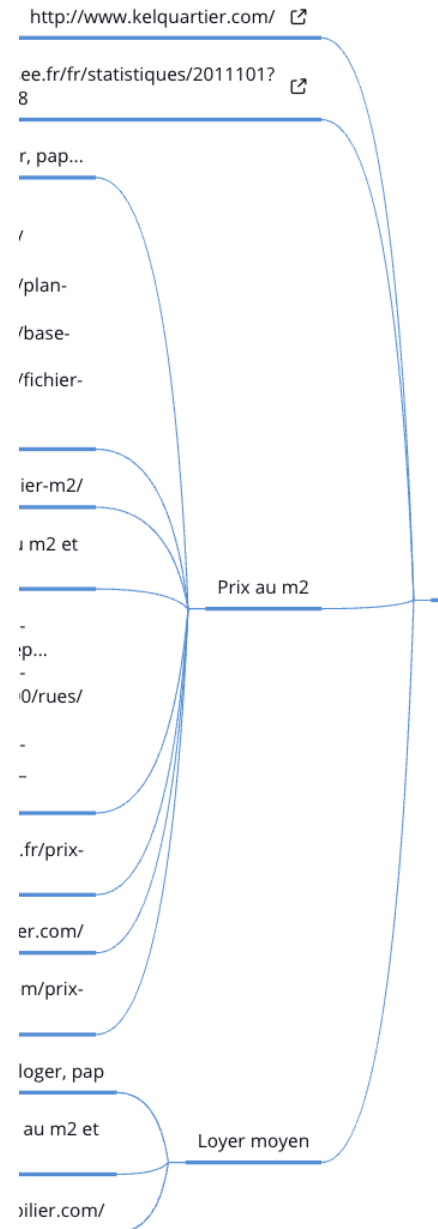
- Se Loger:
 - Se Loger est un site web de recherche de propriétés.
 - Se Loger propose une API pour accéder à ses données immobilières. L'API offre des fonctionnalités telles que la recherche de biens immobiliers, la récupération de détails sur les biens immobiliers et la récupération de photos pour les biens immobiliers.



- Open Street map
 - Open Street Map est un projet collaboratif de cartographie libre et ouverte.
 - Les données sont collectées par des contributeurs volontaires du monde entier et sont disponibles gratuitement sous une licence ouverte.



- Cadastre
 - Le cadastre est un service public français qui gère l'ensemble des plans et des données cadastrales du territoire français. Il fournit des informations sur les propriétés foncières, les parcelles cadastrales, les bâtiments et les propriétaires. Il est géré par la Direction



générale des finances publiques (DGFIP)
et est accessible au public via un site web
dédié.

- Les informations contenues dans le cadastre sont utilisées pour déterminer les impôts fonciers et pour d'autres raisons administratives.

4.2 Recherche des bonnes passerelles

Lors de la récupération de données depuis des pages sur internet, l'on a souvent à faire à des sites internet qui mettent en place des protections contre des potentielles surcharges de leur site. Il est donc souvent difficile de faire des requêtes trop rapides, trop souvent, de façon trop robotique ou depuis la même adresse IP ou origine géographique.

Devant la complexité de tout ces paramètres, il est important de faire des schémas clairs pour la maintenance évolutive.

Après une recherche approfondie, j'ai construit un tableau comparatif des prix des services. Je les trie selon certains critères. Je priorise par exemple des services qui proposent une première offre gratuite pour des tests puis qui proposent des offres évolutives.

nom	prix	Fonctionnalités
-----	------	-----------------

Les colonnes du tableau de tri

\$29 Hobby; \$99 Pro; Différences: Nombres de requêtes simultanées (5 vs 15) et d'appels API réussies par mois (250.000 vs 1.250.000)	Une API; rotating proxies; keep alive sessions; javascript rendering;
---	---

Un exemple de remplissage (sans le nom)

4.3 Exécution asynchrone des récupérations de données

C'est un point crucial. Quand on récupère une grosse quantité de données en même temps, on se rend compte très vite que l'on est bridé par la dimension synchrone de certains langages de programmation. Le PHP ne fait pas exception.



Notez que le PHP n'est pas un langage asynchrone de base. Cela signifie que les requêtes de récupération de données sont exécutées de manière synchrone, ce qui peut entraîner des temps d'attente importants lors de la collecte de grandes quantités de données.

Une solution courante pour exécuter des requêtes de récupération de données de manière asynchrone est d'utiliser `curl_multi_exec`. Avec cette méthode, vous pouvez exécuter plusieurs requêtes en même temps, ce qui permet de récupérer des données plus rapidement.

`curl_multi_exec` est une fonction de la bibliothèque cURL de PHP qui permet d'exécuter plusieurs requêtes simultanément. Cette fonction prend un tableau de gestionnaires de ressources cURL et exécute toutes les requêtes en parallèle. Vous pouvez ensuite récupérer les résultats de chaque requête à l'aide de `curl_multi_getcontent`.

4.4 Organisation du code

Mon code est organisé en approche orientée objet avec des interfaces, des classes abstraites et des classes en PHP. Cette méthode me permet de séparer les différentes fonctionnalités de mon application en modules distincts, ce qui facilite la maintenance et la mise à jour du code.

Les interfaces définissent les méthodes que chaque classe qui implémente l'interface doit implémenter. Les classes abstraites fournissent une implémentation par défaut pour ces méthodes, qui peuvent ensuite être surchargées par les classes qui les étendent. Les classes en PHP sont les classes qui implémentent réellement le code et les fonctionnalités de mon application.

En utilisant cette approche, j'organise mon code en modules distincts, chacun implémentant une fonctionnalité spécifique de mon application. Cela facilite la maintenance et la mise à jour du code, car chaque module peut être modifié sans affecter les autres modules.



L'utilisation d'une approche orientée objet avec des interfaces, des classes abstraites et des classes en PHP facilite l'organisation et la maintenance de mon code.

4.5 Vérification des données récupérées



Les **"guard clauses"** sont des structures de contrôle de flux qui permettent de vérifier si une condition est vraie ou fausse avant de poursuivre l'exécution d'un programme. Ils sont souvent utilisés pour prévenir les erreurs ou les comportements inattendus dans le code.

Dans le contexte de la récupération de données, les "guard clauses" sont souvent utilisées pour vérifier si les données récupérées sont valides avant de les traiter. Par exemple, on peut vérifier si les données sont bien au format attendu, si elles ne sont pas vides ou si elles contiennent des caractères non valides.

En utilisant des "guard clauses", on peut s'assurer que les données récupérées sont valides avant de les traiter, ce qui permet d'éviter les erreurs ou les comportements inattendus dans le reste du code. Cela contribue également à améliorer la qualité et la fiabilité de l'application en évitant les bugs ou les erreurs de traitement des données.

5. Gestion de la maintenance (corrective/évolutive)

La maintenance d'un projet informatique est essentielle pour assurer son bon fonctionnement à long terme. Elle peut être corrective, c'est-à-dire qu'elle vise à corriger des bugs ou des erreurs dans le code existant, ou évolutive, c'est-à-dire qu'elle vise à ajouter de nouvelles fonctionnalités ou à améliorer les fonctionnalités existantes.

5.1 Mise à jour de la documentation du SI

Pour la gestion de la maintenance corrective et évolutive du SI, il est important de maintenir à jour la documentation du SI. Cela permet de garantir que les développeurs ont accès à toutes les informations nécessaires pour effectuer des modifications et des mises à jour en toute sécurité et en toute efficacité.

En maintenant une documentation à jour, l'équipe de développement peut travailler de manière collaborative et efficace pour résoudre les problèmes et maintenir le SI à jour. Cela permet également de garantir que le système est toujours en mesure de répondre aux besoins de l'entreprise et de ses utilisateurs.

5.2 Évaluation de la qualité de la solution

La qualité de la solution a été évaluée en utilisant une suite de tests a été effectuée pour garantir que toutes les fonctionnalités fonctionnent correctement et que les données sont traitées de manière fiable.

Ensuite, des retours ont été recueillis auprès des futurs utilisateurs, en particulier du pôle de communication, afin de s'assurer que l'interface utilisateur est conviviale et que toutes les fonctionnalités sont utiles et faciles à utiliser.

Enfin, plusieurs versions ont été publiées pour permettre aux utilisateurs de tester la solution et de fournir des commentaires. Ces commentaires ont été pris en compte pour améliorer la solution et garantir qu'elle répond aux besoins des utilisateurs et de l'entreprise.

5.3 Procédure de correction d'un dysfonctionnement

Lorsqu'un dysfonctionnement est signalé dans le code, il est important de le corriger rapidement pour garantir le bon fonctionnement de l'application. Pour faciliter la gestion des dysfonctionnements, l'équipe utilise GitHub et les **pull requests**.



En utilisant GitHub et les pull requests, l'équipe peut gérer efficacement les dysfonctionnements et les corrections dans le code. Cette méthode permet de garantir que le code est de qualité et que les corrections sont apportées rapidement pour assurer le bon fonctionnement de l'application.

La procédure pour signaler les problèmes et résoudre les "pull requests" dans l'équipe est la suivante :

1. Tout d'abord, lorsque qu'un dysfonctionnement est détecté dans le code, un membre de l'équipe peut créer une "issue" sur GitHub. Cette "issue" doit contenir une description détaillée du dysfonctionnement, ainsi que les instructions nécessaires pour reproduire le problème.
2. Ensuite, un membre de l'équipe peut se porter volontaire pour résoudre le dysfonctionnement signalé. Le membre de l'équipe crée alors une nouvelle branche dans le dépôt GitHub pour corriger le dysfonctionnement.
3. Une fois la correction apportée, le membre de l'équipe crée une "pull request" sur GitHub. Cette "pull request" comprend les modifications apportées au code

ainsi que des commentaires détaillés expliquant les raisons de chaque modification.

4. La "pull request" est ensuite examinée par un autre membre de l'équipe, qui vérifie que les modifications sont pertinentes et que le code est de qualité. Si nécessaire, des commentaires sont ajoutés à la "pull request" pour suggérer des améliorations ou des modifications.
5. Une fois que la "pull request" a été approuvée, le membre de l'équipe peut fusionner la branche dans le dépôt principal. La correction est alors disponible pour tous les utilisateurs de l'application.

Quelle que soit la méthode utilisée, l'équipe de développement d'Immozia s'engage à diagnostiquer et à résoudre les problèmes signalés dans les plus brefs délais. Les utilisateurs seront informés de l'avancement de la résolution du problème et de la mise à disposition de correctifs ou de mises à jour pour résoudre le problème.

En cas de dysfonctionnement critique, l'équipe de développement prendra des mesures immédiates pour résoudre le problème et en informer les utilisateurs le plus rapidement possible.

6. Bilan du projet

6.1 Validation des exigences point par point

Pour ce projet:

- J'ai géré la mise en production du projet par le **parc informatique** qui m'a été fournis.
- Un **suivi des demandes** et de la correction évolutive a été faite par mes soins.
- Ce projet est partie prenante du futur **site phare en ligne** de l'entreprise.
- Pour toutes mes missions je **travaille en mode projet** avec un suivi ainsi qu'une organisation par paliers.
- J'ai suivi l'évolution et l'utilisation de mon moteur par le reste de l'équipe en fournissant **la documentation et l'aide nécessaire** pour une bonne compréhension.

- Grâce à ce projet j'ai **perfectionné ma programmation orienté objet**. Ce qui, **dans le cadre de mon objectif professionnel**, est crucial.

6.2 Axes d'amélioration

La complexité du projet ainsi que mon apprentissage continu pendant le projet font que j'ai forcément des choses que j'aurais faites différemment. En tant que développeur, notre travail est de savoir évoluer rapidement donc de constamment apprendre de nouvelles façons de travailler. Il est donc naturel que le code que j'ai produit il y a 3 mois ne soit pas mon idéal de programmation actuel.

Lors de l'utilisation du paradigme de programmation orienté objet, j'ai choisi certaines méthodes. Aujourd'hui, je choisirais d'autres méthodes car je sais ce que le projet attend. Tout est question d'anticipation.

L'héritage dans ce projet

Le noyau de ce projet est devenu trop compliqué pour être modifié intégralement. Initialement, le projet a été construit avec une pensée orientée objet. Mais si je devais refaire le noyau, j'augmenterais l'héritage selon un meilleur diagramme de classe.

S'il n'y a pas de diagramme de classe dans cette documentation, c'est parce que j'ai pensé à en créer un beaucoup trop tard. À l'avenir, je m'appuierai donc beaucoup plus sur des diagrammes préalablement réalisés pour l'organisation de mes classes et des différents héritages.

6.3 Compétences acquises

Ce projet m'a permis d'acquérir de nombreuses compétences en développement de logiciels. J'ai appris à travailler de manière collaborative avec une équipe, à gérer des projets de grande envergure, à collecter et organiser des données immobilières, et à utiliser des outils tels que Notion pour la gestion de projets.

J'ai également amélioré mes compétences en programmation orientée objet en utilisant des interfaces, des classes abstraites et des classes en PHP pour organiser mon code en modules distincts. J'ai appris à utiliser les "guard clauses" pour vérifier les données récupérées et à exécuter des requêtes de récupération de données de manière asynchrone à l'aide de `curl_multi_exec`.

Enfin, j'ai acquis une meilleure compréhension de la gestion de la maintenance corrective et évolutive du SI, de la mise à jour de la documentation du SI et de la procédure de correction des dysfonctionnements. J'ai également perfectionné ma

programmation orientée objet, ce qui est crucial pour atteindre mes objectifs professionnels.



Cette documentation est la propriété intellectuelle de Grimaldi Baptiste.
portfolio.baptistegrimaldi.info/legal

Avis de droits d'auteur

Informations légales sur les droits d'auteur de cette documentation

Cette documentation a été créée par moi, Grimaldi Baptiste, en tant qu'étudiant. Tous les droits d'auteur sur cette documentation sont réservés. Aucune partie de cette documentation ne peut être reproduite, stockée dans un système de récupération ou transmise sous quelque forme ou par quelque moyen que ce soit, électronique, mécanique, photocopie, enregistrement ou autre, sans l'autorisation préalable écrite de l'auteur.

Toute utilisation non autorisée de cette documentation peut constituer une violation des lois sur les droits d'auteur et entraîner des poursuites judiciaires.

Si vous souhaitez utiliser cette documentation à des fins éducatives ou autres, veuillez contacter l'auteur pour obtenir l'autorisation écrite nécessaire.

Copyright Grimaldi Baptiste, 2023.

Par Baptiste Grimaldi