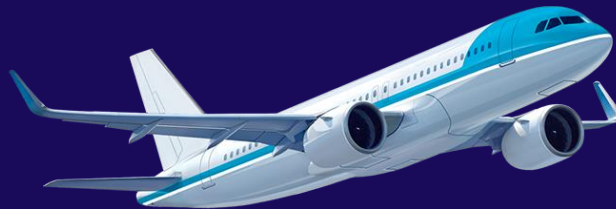


Rapport de Stage AFIDIUM

03.01.2022 au 11.07.2022



epsi

l'école d'ingénierie
informatique

Sommaire :

| | |
|--|-----------|
| Remerciements : | 3 |
| Introduction : | 4 |
| I- Présentation et description de l'entreprise : | |
| - Présentation de l'entreprise : | 5 |
| - Les principaux métiers et la qualification des personnels : | 5 |
| - Produits proposés, cible : | 6 |
| - Clientèle de l'entreprise : | 6 |
| II- Présentation des missions réalisées | |
| - Test de bout en bout : | 7 |
| - Migration de plusieurs tests : | 12 |
| - Réorganisation de la configuration des paramètres : | 13 |
| - Création d'une documentation et échanges avec des alternants : | 17 |
| III- Problèmes à résoudre / contraintes / limites : | 18 |
| IV- Démarche suivie pour effectuer les missions | |
| - Conception d'un test de bout en bout : | 19 |
| - Exécution d'un test de bout en bout : | 21 |
| - Réunion quotidienne : | 21 |
| V- Les outils et services informatiques qu'utilise l'entreprise : | 22 |
| VI- Évaluation des réalisations et des compétences mobilisées | |
| - Adéquation du travail : | 23 |
| - Rétrospective des compétences exploitées : | 24 |
| Conclusion : | 25 |
| Bibliographie : | 26 |
| Annexes : | 27 |

Remerciements

Je tiens dans un premier temps à remercier Patrick Rajaud et David Faveur, pour m'avoir de nouveau permis de faire mon stage au sein d'AFIDIUM.

Je remercie également toute l'équipe de développement pour m'avoir accordé de leur temps si précieux pour répondre à mes différentes questions et m'avoir aidé à résoudre différents problèmes que j'ai pu rencontrer lors de mon stage. Merci de m'avoir fait découvrir leur métier et leur quotidien.

Un grand merci également pour leur confiance, leur patience et toutes leurs explications.

Introduction :

Dans le cadre de ma formation à l'EPSE, j'ai pu réaliser mon stage de nouveau au sein de la société AFIDIUM.

Ce stage a été réalisé afin de compléter mon enseignement en informatique de 2eme année de Bachelor et de BTS.

Ayant eu de grandes difficultés pour trouver un stage, suite à la situation de crise du covid-19, j'ai eu la chance de pouvoir revenir chez AFIDIUM, car mon premier stage chez eux m'avait beaucoup plu. Ça a même été un grand plus car ça m'a permis de parfaire ma découverte de l'entreprise. J'ai pu découvrir encore plus de choses sur mon futur métier en échangeant avec tous les développeurs qui m'ont fait confiance.

Durant ce stage, plusieurs missions m'ont été confiées, j'ai pu refaire des tests de bout en bout afin d'automatiser les applications de l'entreprise sur les nouvelles fonctionnalités qui ont été apportées, ensuite j'ai fait de la documentation puis j'ai participé à la reconfiguration de l'interface des paramètres de l'entreprise.

Mon stage a commencé le 03 janvier 2022 et s'est terminé le 11 février 2022.

Présentation et description de l'entreprise :

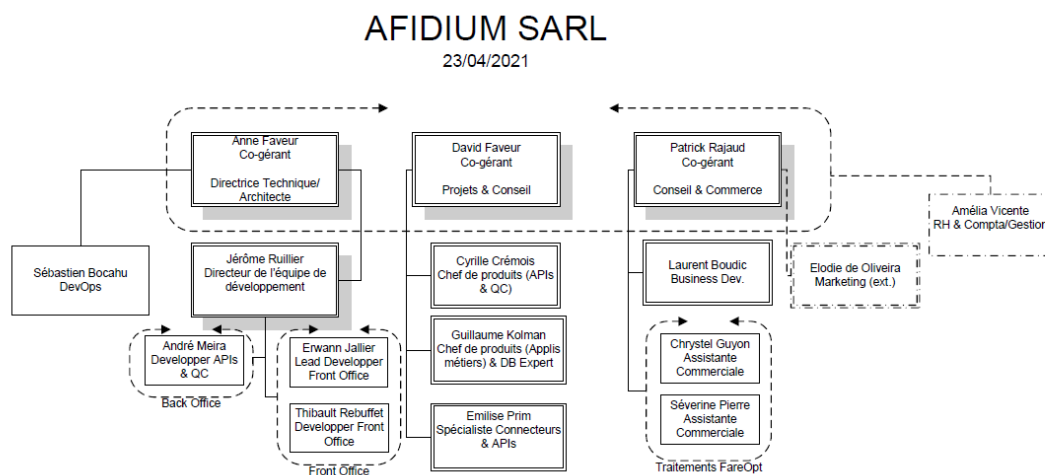
1- Présentation de l'entreprise :

L'entreprise AFIDIUM qui a été fondée en 2005 par David Faveur et Patrick Rajaud.

Elle propose un grand nombre de services et vend des solutions innovantes pour optimiser les revenus, améliorer la productivité et la rentabilité des entreprises du secteur du tourisme telles que les Agences de voyages.

Nombre d'employés et organisme :

L'entreprise AFIDIUM est composée de 16 employés dont voici l'organigramme :



2- Les principaux métiers et la qualification des personnels :

Les salariés d'AFIDIUM sont essentiellement des :

- Back-end développeur
- Sales assistant
- Front-end développeur
- Product manager
- Devops ingénieur
- Product manager

3- Produits proposés, cible :

L'entreprise AFIDIUM, propose six types de services aux entreprises du secteur du tourisme :

- FAREOPT : permet d'améliorer ses marges : être alerté lorsque de meilleurs tarifs deviennent disponibles sur tous ses PNRs (informations liées à la réservation d'un vol, et qui sont contenues dans les dossiers créés par les compagnies aériennes pour chaque vol).
- QUALITY CHECK : permet de synchroniser des données internes avec les PNRs, de faire un contrôle de cohérence et de faire des émissions automatiques à la date attendue.
- WATABI : permet de créer, ajuster et envoyer une proposition pour un voyage sur mesure en quelques minutes à un client.
- IAQOBO : permet de gagner en temps et en efficacité avec l'interface de réservation multi-sources pour l'hôtel et l'aérien.
- SASHA : Aérien, hôtel, mid-office... permet de rechercher et gérer tous les dossiers depuis une même interface.
- OSMAN : permet de maximiser les revenus de stocks aériens (places d'avion réservées à l'avance en grande quantité) grâce à une solution de gestion de stocks.

4- Clientèle de l'entreprise :

La clientèle d'AFIDIUM est uniquement constituée d'entreprises spécialisées dans le tourisme. En effet, on y retrouve notamment un grand nombre d'agences de voyages tel que Carrefour Voyages, Le Cercle des Vacances, Tui, Voyageurs du monde ou encore Les maisons du voyage.

Présentation des missions réalisées

1- Tests de bout en bout

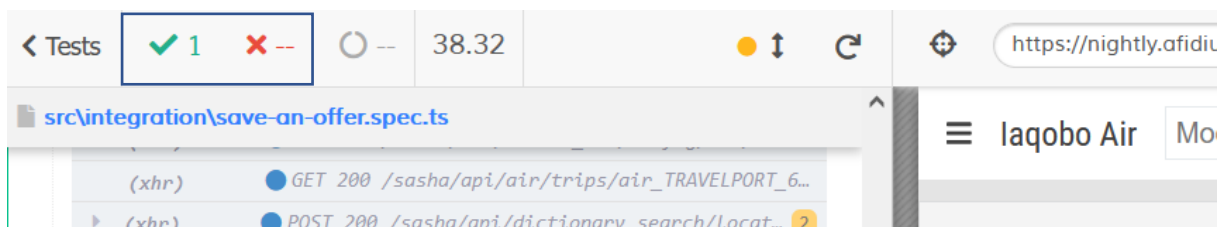
La première mission qui m'a été donnée, était de créer et de conceptualiser des tests de bout en bout. L'objectif de ce type de test est de pouvoir analyser les applications au niveau de leur bon fonctionnement et également au niveau de l'affichage.

En effet, il est possible d'inspecter toutes les informations que les applications web nous proposent.

Pour réaliser ces tests de bout en bout, j'ai utilisé le Framework Cypress, car il est très facile d'utilisation et il permet à la fois de tester les applications AngularJS et Angular mais aussi des sites web non-Angular. J'ai également utilisé les langages de programmations TypeScript et Javascript. L'avantage pour moi était que j'avais déjà fait des tests de bout de bout et que je savais comment cela fonctionnait.

J'ai pu réaliser 3 tests sur l'application laqobo-air. Mon premier test consistait à vérifier si le prix de vente et le prix d'achat était bien affiché sur le billet (Annexe 1). Mon second test était de simuler une sauvegarde d'offre (Annexe 2) et mon troisième test avait pour but de simuler une sauvegarde d'offre puis de réserver l'offre sauvegardée (Annexe 3). Une fois tous les tests fonctionnels terminés, j'ai pu les lancer régulièrement de manière automatisée à l'aide de Jenkins qui est un outil open source de serveur d'automatisation. Il aide à automatiser les parties du développement.

Pour voir si un test échoue ou s'il s'est exécuté avec succès sur Cypress :



On observe un '1' à côté du logo vert cela veut dire qu'il n'y a pas eu d'erreur sinon s'il y a une erreur alors le '1' sera à côté de la croix rouge.

Et sur Jenkins :

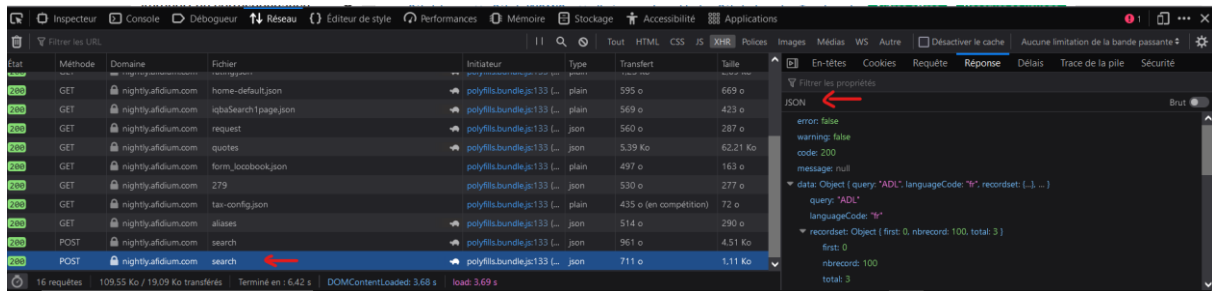
| Jenkins | | | | | | Pipelines | 🔗 | Déconnexion |
|----------------------|------|--------|---------------------------|--------|--------------|-----------|----------|---------------|
| 🔔 / Tests / e2e ☆ ⚙️ | | | | | | Activité | Branches | Pull requests |
| ▶ Lancer | | | | | | 🛑 Disable | | |
| ÉTAT | RUN | COMMIT | MESSAGE | DURÉE | TERMINÉ | | | |
| ✅ | 1106 | — | null (54339+86ff260ccd51) | 3m 29s | 5 hours ago | 🔄 | | |
| ✅ | 1105 | — | null (54329+cad431ef0e08) | 3m 29s | 11 hours ago | 🔄 | | |
| ✅ | 1104 | — | null (54320+0cb87f0f42aa) | 3m 34s | 17 hours ago | 🔄 | | |
| ✅ | 1103 | — | null (54313+3c0fbf1f8f50) | 3m 27s | a day ago | 🔄 | | |
| ✅ | 1102 | — | null (54296+2db6bb8950e1) | 3m 7s | a day ago | 🔄 | | |
| ❌ | 1101 | — | null (54296+2db6bb8950e1) | 2m 51s | a day ago | 🔄 | | |
| ✅ | 1100 | — | null (54296+2db6bb8950e1) | 3m 33s | 2 days ago | 🔄 | | |
| ✅ | 1099 | — | null (54288+cf0f80ee0098) | 4m 27s | 2 days ago | 🔄 | | |
| ✅ | 1098 | — | null (54288+cf0f80ee0098) | 3m 22s | 2 days ago | 🔄 | | |
| ✅ | 1097 | — | null (54288+cf0f80ee0098) | 2m 46s | 2 days ago | 🔄 | | |
| ✅ | 1096 | — | null (54288+cf0f80ee0098) | 2m 40s | 3 days ago | 🔄 | | |
| ✅ | 1095 | — | null (54288+cf0f80ee0098) | 2m 49s | 3 days ago | 🔄 | | |
| ✅ | 1094 | — | null (54288+cf0f80ee0098) | 3m 29s | 3 days ago | 🔄 | | |

Logo vert pour les tests réussis et rouges quand ça échoue.

J'ai également pu créer des « MockId » qui permettent de rendre les tests beaucoup plus fluides lors des échanges de données avec les serveurs de communication aériens dont les GDS. En effet pour créer des « MockId » pour un test de réservation d'un billet d'avion, on récupère tous les échanges de données qui ont lieu entre le navigateur web et l'application. Pour cela il faut d'abord afficher la fonctionnalité « inspecter l'élément » et aller dans la partie « network » et sélectionner les requêtes XHR.

| Inspecteur Console Débugueur Réseau Éditeur de style Performances Mémoire Stockage Accessibilité Applications | | | | | | | | | |
|---|---------|--------------------------|---|--------------------------------|-------|-----------|----------|---------|---------|
| Filtrer les URL | | | | | | | | | |
| État | Méthode | Domaine | Fichier | Initiateur | Type | Transfert | Taille | 0 ms | 40,96 s |
| 200 | POST | nightly.afidium.com | search | polyfills.bundle.js:133 (xhr) | json | 4,08 Ko | 35,80 Ko | 323 ms | |
| 200 | POST | nightly.afidium.com | search | polyfills.bundle.js:133 (xhr) | json | 962 o | 4,51 Ko | 188 ms | |
| 200 | POST | nightly.afidium.com | search | polyfills.bundle.js:133 (xhr) | json | 961 o | 4,51 Ko | 188 ms | |
| 200 | POST | nightly.afidium.com | search | polyfills.bundle.js:133 (xhr) | json | 4,30 Ko | 37,10 Ko | 162 ms | |
| 200 | POST | nightly.afidium.com | search | polyfills.bundle.js:133 (xhr) | json | 682 o | 794 o | 177 ms | |
| 200 | POST | nightly.afidium.com | search | polyfills.bundle.js:133 (xhr) | json | 575 o | 199 o | 2619 ms | |
| 200 | GET | nightly.afidium.com | rating.json | polyfills.bundle.js:133 (xhr) | plain | 1,23 Ko | 2,69 Ko | 48 ms | |
| 200 | POST | nightly.afidium.com | search | polyfills.bundle.js:133 (xhr) | json | 961 o | 4,51 Ko | 227 ms | |
| 200 | POST | nightly.afidium.com | search | polyfills.bundle.js:133 (xhr) | json | 682 o | 794 o | 183 ms | |
| 200 | POST | c308445.ingest.sentry.io | /api/1770665/store/?sentry_key=5f6d57324894162b01a0012b714168&sentry_ver= | polyfills.bundle.js:21 (fetch) | json | 476 o | 41 o | 1274 ms | |
| 23 requêtes 144,69 Ko / 27,69 Ko transférés Terminé en : 37,52 s DOMContentLoaded: 745 ms load: 751 ms | | | | | | | | | |

En sélectionnant cela on peut récupérer les données qui ont été stockées dans un Json comme pour les données de notre recherche.

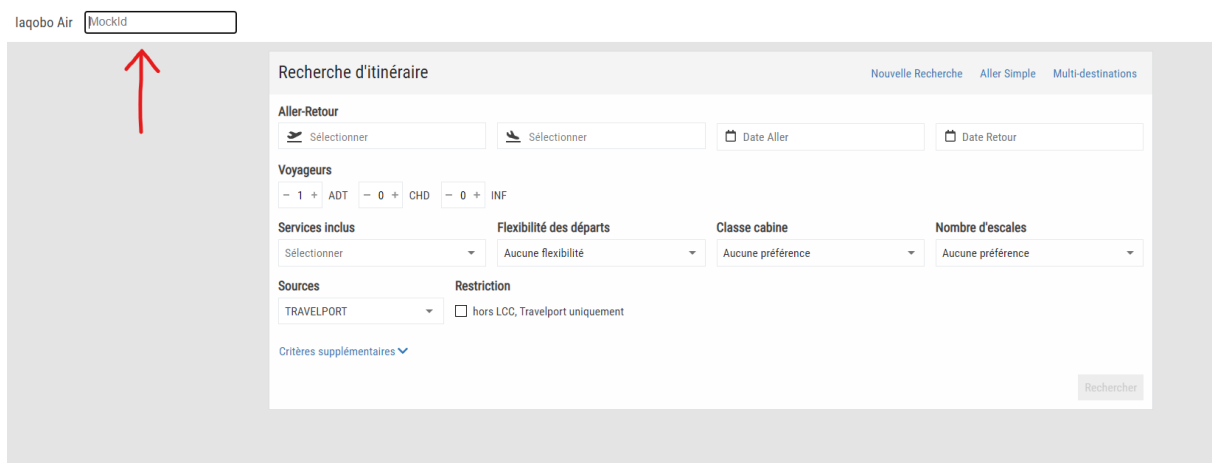


Lors de notre test cela permettra de récupérer directement ces données et de gagner du temps car l'application n'aura pas besoin de communiquer avec les GDS car elle possède déjà la réponse de la recherche.

Toutes les données des Json sont alors stockées dans un fichier qu'on nomme en fonction du test auquel il est associé, exemple : pour le test qui permet de faire une sauvegarde d'offre, son mock s'appelle : IQBA-E2E-MAVED-OFFER-RESERVATION.

On écrit IQBA pour le nom de l'application qui est laqobo-air et E2E pour « test end to end ».

C'est à cet endroit qu'on spécifie l'utilisation d'un mock en entrant le nom du fichier.



Pour un test qui n'utilise pas de mock voici le temps d'exécution :

Tests 1 38.32

src/integration/save-an-offer.spec.ts

(xhr) GET 200 /sasha/api/air/trips/air_TRAVELPORT_6...

(xhr) POST 200 /sasha/api/dictionary_search/Locat...

5 get .fromto

6 -assert expected [<div.fromto>, 65 more...] to be visible

7 get .e2e-list-up

8 -click

9 get .quote-result:nth-child(1) booking .e2e-select-quote-btn

10 -click

(xhr) GET 200 /sasha/api/air/trips/air_TRAVELPORT_6...

11 wait @postQuoteOffer

(xhr) GET 200 /sasha/api/sam/contacts/299

(xhr) GET 200 /sasha/api/sam/companies/4

(xhr) POST 200 /sasha/api/air/quotes/quoteOffer

(xhr) POST 200 /sasha/api/dictionary_search/Locatio...

(req) https://nightly.afidium.com/sasha... postQuoteOffer

(xhr) POST 200 /sasha/api/dictionary_search/Locatio...

12 get .e2e-select-offer

13 -click

(xhr) POST 200 /sasha/api/air/offers/save

(req) https://nightly.afidium.com/sasha... postSaveOffer

14 wait @postSaveOffer

15 get .e2e-success-save-offer .success

16 -assert expected <div.notification.success.ng-star-inserted> to be visible

laqobo Air MockId

Résumé de la recherche

8 mars 2022 PAR → BCN 18 mars 2022 BCN → PAR 1ADT Types de tarif Public, Tour

Sauvegarde d'une offre

Sauvegarde d'une offre réussie

Pour un test qui utilise un mock voici le temps d'exécution :

Tests 1 20.84

src/integration/save-an-offer.spec.ts

9 get .fromto

10 -assert expected [<div.fromto>, 65 more...] to be visible

11 get .e2e-list-up

12 -click

(xhr) GET 200 /sasha/wui/config/home/home-default, j...

(xhr) POST 200 /sasha/api/dictionary_search/Locat...

13 get .quote-result:nth-child(1) booking .e2e-select-quote-btn

14 -click

(xhr) GET 200 /sasha/api/air/trips/air_TRAVELPORT_6...

15 wait @postQuoteOffer

(xhr) GET 200 /sasha/api/sam/contacts/299

(xhr) GET 200 /sasha/api/sam/companies/4

(xhr) POST 200 /sasha/api/dictionary_search/Locatio...

(xhr) POST 200 /sasha/api/air/quotes/quoteOffer

(xhr) POST 200 /sasha/api/dictionary_search/Locatio...

(req) https://nightly.afidium.com/sasha... postQuoteOffer

16 get .e2e-select-offer

17 -click

(xhr) POST 200 /sasha/api/air/offers/save

(req) https://nightly.afidium.com/sasha... postSaveOffer

18 wait @postSaveOffer

19 get .e2e-success-save-offer .success

20 -assert expected <div.notification.success.ng-star-inserted> to be visible

laqobo Air IQBA-E2E-AIR-SEARCH

Résumé de la recherche

15 mars 2022 PAR → NYC 25 mars 2022 NYC → PAR 1ADT Source TRAVELPORT

Sauvegarde d'une offre

Sauvegarde d'une offre réussie

On observe un gain de temps de 20 secondes pour un test qui utilise un mock

J'ai également créé une fonction qui permet de spécifier si l'on souhaite lancer un test avec des mocks ou non.

Voici la fonction :

```
1 export enum EMockableTest {
2     SeeRetailPrice = 0,
3     AirSearchSilent = 1,
4     SavedOfferReservation = 2,
5     RouteSearch = 3,
6 }
7
8
9 export function runTestWithMockOrNot(testId: EMockableTest): string {
10
11     const withMock: boolean = true;
12
13     const mockableTestsMap: { [key in EMockableTest]: string } = {
14         [EMockableTest.SeeRetailPrice]: 'IQBA-E2E-SEE-RETAIL-PRICE-2022-01-11',
15         [EMockableTest.AirSearchSilent]: 'IQBA-E2E-AIR-SEARCH-SILENT-2022-01-07',
16         [EMockableTest.SavedOfferReservation]: 'IQBA-E2E-MAVED-OFFER-RESERVATION-2022-01-11',
17         [EMockableTest.RouteSearch]: 'IQBA-E2E-ROUTE-SEARCH-2022-01-11',
18     };
19
20
21     return (withMock) ? mockableTestsMap[testId] : null;
22 }
23
24
```

Si la constante withMock = false, alors les tests n'utilisent pas les mocks et inversement pour withMock = true.

Voici où est appelée la fonction dans les tests :

```
import {login, password} from "../../sasha-e2e/src/config";
import {LoginPageObject} from "../support/login.po";
import {RouteSearchPageObject} from "../support/route-search.po";
import {EMockableTest, runTestWithMockOrNot} from "../../sasha-e2e/src/support/run-test-with-mock-or-n

describe('title: "Iaqobo air : See if the retail price is displayed correctly"', fn(): void => {
    const loginPage = new LoginPageObject();
    const routeSearchPage = new RouteSearchPageObject();

    beforeEach(fn(): void => {
        routeSearchPage.postSearchJsonWithRetail();
        loginPage.visit();
        loginPage.typeLogin(login);
        loginPage.typePassword(password);
        loginPage.clickLogin();
    });

    it('title: "Iaqobo air : See if the retail price is displayed correctly"', fn(): void => {
        routeSearchPage.visitIaqoboAirSearchHome(runTestWithMockOrNot(EMockableTest.SeeRetailPrice));
        routeSearchPage.assertReservationButtonToBeVisible();
        routeSearchPage.clickAllProposalsListUp();
        routeSearchPage.clickTravelReservation();
        routeSearchPage.assertRetailPriceButtonToBeVisible();
        routeSearchPage.assertPurchasePriceButtonToBeVisible();
    })
})
```

Pour cela j'ai utilisé des Enums que j'ai nommé par le nom des tests et que j'ai associé avec les mocks correspondants.

2- Migration de plusieurs tests

Dans un second temps, j'ai pu migrer plusieurs tests de bout en bout qui ont été faits par moi-même durant mon premier stage avec le framework Protractor, afin de tous les passer sur Cypress, car ce dernier a été laissé un peu à l'abandon par ses créateurs.

Voici le but des différents tests :

- Un test qui permettait de se connecter à une application de réservation de vol qui s'appelle laqobo Air, afin de voir si les prix qui sont affichés dans la partie meilleurs prix correspondent à la liste des vols affichés par l'ordre des prix croissant. De même pour les horaires de vol, pour regarder si le vol le plus rapide correspond à la liste.
- Un test qui permettait d'envoyer des mails via l'application et voir si les dates d'envoi correspondaient bien à l'heure réelle à laquelle ils avaient été envoyés.
- Un test qui permettait d'ajouter des notes dans une réservation de vol.
- Un test qui permettait de regarder si l'affichage des réservations était correct.
- Un test qui permettait de vérifier qu'en cas d'annulation d'une réservation, la réservation était effectivement bien annulée.
- Un test qui permettait de générer des rapports liés au client de l'entreprise avec une exécution, avec une date précise (tous les 1ers de chaque mois) et avec un horaire précis également.
- Et un dernier test qui permettait de voir si les serveurs de l'entreprise étaient bien connectés.

Ces tests de bout en bout représentent des enjeux importants pour l'entreprise AFIDIUM, car ces tests lui permettent d'être informée en temps réel sur la fonctionnalité et les erreurs que rencontrent leurs applications. En effet, cela permet d'éviter un maximum de problèmes techniques lors de la livraison de leur produit aux différents clients.

3- Réorganisation de la configuration des paramètres

Ma deuxième mission a consisté à reconfigurer l'interface des paramètres de l'entreprise. Ma première tâche était de traiter un document Excel et d'ajouter les différents types de champs nécessaires dans le formulaire pour les différents paramètres ajoutés puis d'indiquer les caractères qu'on accepte dans chaque champ.

Ensuite je devais créer des écritures régulières afin de bien spécifier à chaque champ du formulaire les caractères autorisés puis bloquer la soumission du formulaire quand un champ possédait des caractères non autorisés comme un espace ou des majuscules...

Exemple d'expression régulière pour une url :

```
REGULAR EXPRESSION
: / ^https?:\/\/[a-zA-Z0-9\.\:\/\?=\-\_] +$

TEST STRING
http://abbeyservicesprod.com:8080/iCom/servlet/conn
```

Pour une url on spécifie qu'on accepte uniquement si l'url commence par http ou https et si elle possède des /, ., lettres majuscules et minuscules, des nombres, =, : - puis _.

Donc en remarque que l'url est surlignée en bleu, ce qui veut dire que l'url est bien correcte.

Ici nous pouvons voir que l'url n'est pas surlignée en bleu car celle-ci ne commence pas par http ni par https, donc l'url n'est pas acceptée.

```
REGULAR EXPRESSION
: / ^https?:\/\/[a-zA-Z\0-9\.\:\/] +[\/] $

TEST STRING
adz://ws0.bonotel.com/bonotelapps/bonotel/reservation/
```

Voici un exemple de cas réel :

- Ici on voit que l'url ne correspond pas.

The screenshot shows the AFIOIUM Paramétrages unifiés interface. The breadcrumb trail is [Accueil](#) » [Paramètres](#) » [Assurinco](#) » [Modifier](#). The title is "Modifier Connecteurs : Assurinco". A message states "Les champs marqués * sont requis." Below this, a red box contains the error: "Veuillez corriger les erreurs de saisie : • Endpoint (Test) est invalide." The "Endpoint (Test) *" field contains the invalid URL "adz://wsi.cabinetchaubet.com/?v=2" and is highlighted with a red border. The "Endpoint (Production) *" field contains the valid URL "http://ws.cabinetchaubet.com/v=2". An "Enregistrer" button is at the bottom.

- Ici on observe que l'url est bien conforme.

This screenshot shows the same AFIOIUM Paramétrages unifiés interface, but with valid URLs. The breadcrumb trail is [Accueil](#) » [Paramètres](#) » [Assurinco](#) » [Modifier](#). The title is "Modifier Connecteurs : Assurinco". The message "Les champs marqués * sont requis." is present. Both the "Endpoint (Test) *" field and the "Endpoint (Production) *" field now contain the valid URL "http://wsi.cabinetchaubet.com/?v=2" and "http://ws.cabinetchaubet.com/v=2" respectively. The "Enregistrer" button remains at the bottom.

Une fois cela fait, je devais coder en Php et implémenter ces expressions régulières dans le code de la configuration des paramètres. J'ai également testé si les formulaires acceptaient bien les caractères autorisés puis bloquaient bien aussi les mauvais.

Puis je devais mettre en place des tests unitaires (environ plus de 100 tests) afin de tester plus rapidement chaque formulaire possédant les nouveaux paramètres en insérant des données erronées et correctes.

Pour les tests unitaires je devais faire quatre insertions pour chaque valeur, c'est-à-dire 3 insertions erronées et une correcte, afin de bien vérifier si l'expression régulière était bien correcte.

```
function testValidate() {
    $params = new ABIParams();

    //missing http
    $params->{'wenji.abi.urltest'} = '://abbeyservicesuat.com:7080/testiCom/servlet/conn';
    $params->validate();
    $this->assertArrayHasKey( key: 'wenji.abi.urltest', $params->getErrors());

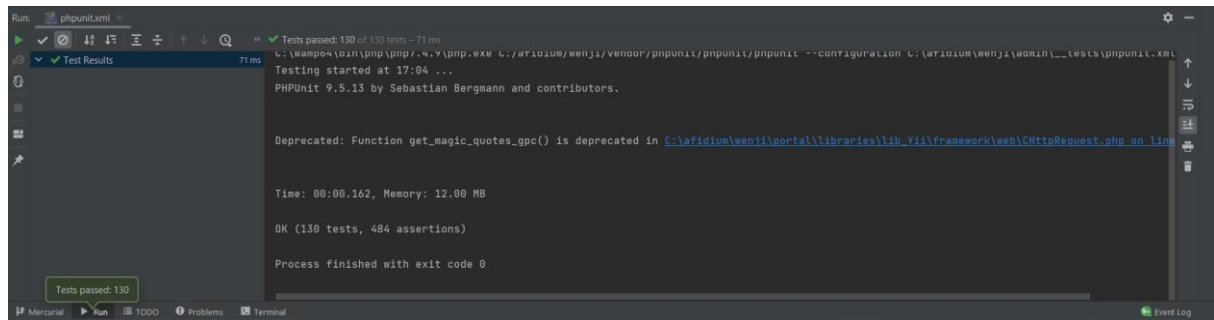
    //space
    $params->{'wenji.abi.urltest'} = 'http://abbeyservicesuat.com:7080/testiCom/servlet/conn ';
    $params->validate();
    $this->assertArrayHasKey( key: 'wenji.abi.urltest', $params->getErrors());

    //malformed line
    $params->{'wenji.abi.urltest'} = 'cesuhttp://abbeyserviat.com:7080/testiCom/servlet/conn';
    $params->validate();
    $this->assertArrayHasKey( key: 'wenji.abi.urltest', $params->getErrors());

    $params->{'wenji.abi.urltest'} = 'http://abbeyservicesuat.com:7080/testiCom/servlet/conn';
    $params->validate();
    $this->assertArrayNotHasKey( key: 'wenji.abi.urltest', $params->getErrors());
}
```

- Dans la première insertion, il manque http.
- Dans la deuxième, il y a un espace à la fin.
- Dans la troisième, l'url est mal formée.
- Dans la quatrième, l'url est correcte.

Voici l'exécution des tests unitaires :



```
Run: phpunit.xml
Tests passed: 130 of 130 tests - 71 ms
C:\wamp64\bin\php\php7.4.3\php.exe C:\aridium\wenji\vendor\phpunit\phpunit\phpunit --configuration C:\aridium\wenji\admin\__tests\phpunit.xml
Testing started at 17:04 ...
PHPUnit 9.5.13 by Sebastian Bergmann and contributors.

Deprecated: Function get_magic_quotes_gpc() is deprecated in C:\aridium\wenji\portal\libraries\lib_vii\framework\web\HttpRequest.php on line

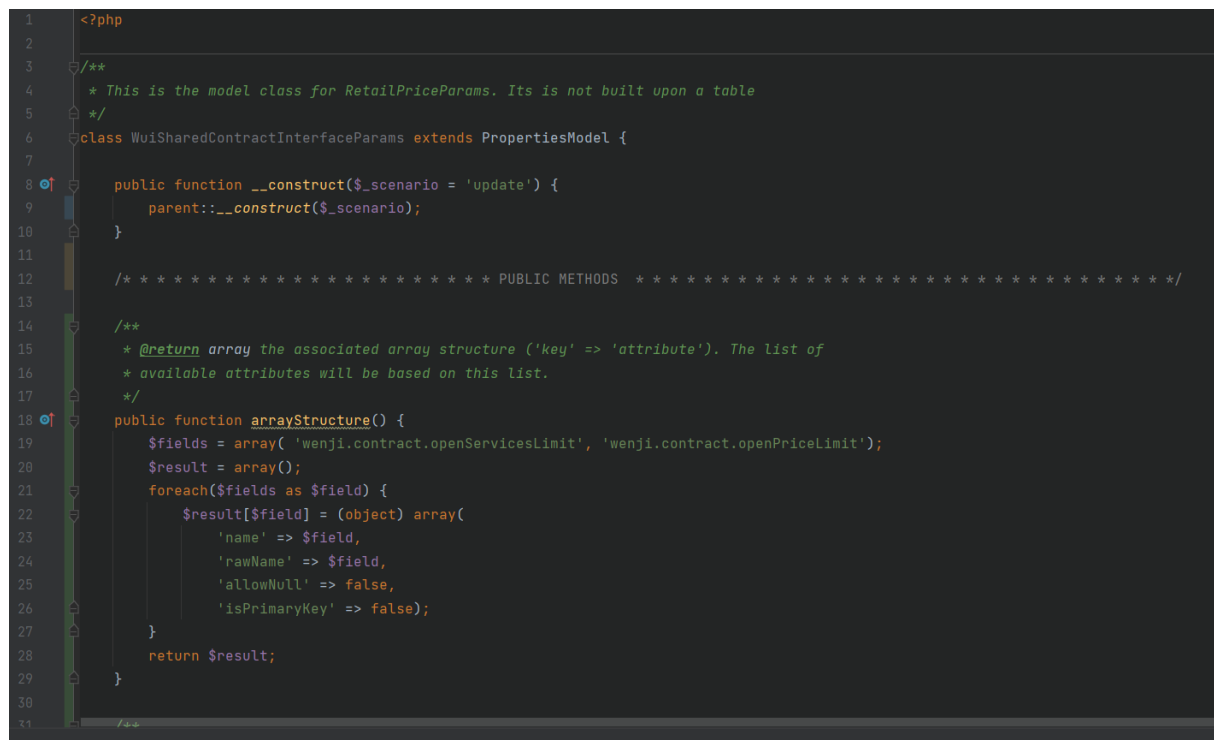
Time: 00:00:162, Memory: 12.00 MB

OK (130 tests, 484 assertions)

Process finished with exit code 0
```

Une fois terminé je devais mettre à jour la structure de tous les fichiers de configuration. En effet, l'entreprise, avec le temps, a modifié la structure de ces fichiers. J'ai donc dû rechercher tous les anciens fichiers et les mettre sous la bonne forme.

Ancien modèle de structure :



```
1 <?php
2
3 /**
4  * This is the model class for RetailPriceParams. Its is not built upon a table
5  */
6 class WuiSharedContractInterfaceParams extends PropertiesModel {
7
8     public function __construct($scenario = 'update') {
9         parent::__construct($scenario);
10    }
11
12    /* ***** PUBLIC METHODS ***** */
13
14    /**
15     * @return array the associated array structure ('key' => 'attribute'). The list of
16     * available attributes will be based on this list.
17     */
18    public function arrayStructure() {
19        $fields = array( 'wenji.contract.openServicesLimit', 'wenji.contract.openPriceLimit');
20        $result = array();
21        foreach($fields as $field) {
22            $result[$field] = (object) array(
23                'name' => $field,
24                'rawName' => $field,
25                'allowNull' => false,
26                'isPrimaryKey' => false);
27        }
28        return $result;
29    }
30
31    /**
```


Nouveau modèle de structure :

```
1 <?php
2
3 /**
4  * This is the model class for RetailPriceParams. Its is not built upon a table
5  */
6 class WuiSharedContractInterfaceParams extends PropertiesModel {
7
8     public function __construct($scenario = 'update') {
9         $config = [
10
11             'wenji.contract.openServicesLimit' => [
12                 'label' => Yii::t('yii-admin-content', "Nombre max de prestations ouvertes"),
13                 'hint' => Yii::t('yii-admin-content', "Nombre de prestations à partir duquel elle sont fermées. EX: '2' => les prestations sont fermées si on a 2 prestations ou plus."),
14                 'default' => '2',
15                 'rules' => [
16                     ['numerical'],
17                 ],
18             ],
19
20             'wenji.contract.openPriceLimit' => [
21                 'label' => Yii::t('yii-admin-content', "Nombre max de prix ouverts"),
22                 'hint' => Yii::t('yii-admin-content', "Nombre de prix à partir duquel ils sont fermés. EX: '2' => les prix sont fermés si on a 2 prix ou plus."),
23                 'default' => '2',
24                 'rules' => [
25                     ['numerical'],
26                 ],
27             ],
28         ];
29
30         parent::__construct($scenario, $config);
31     }
32 }
```

4- Création d'une documentation et échanges avec des salariés en alternance

J'ai pu créer une documentation afin d'expliquer comment créer correctement des tickets d'incident sur Redmine (qui est une application web libre de gestion de projet) pour avoir un suivi du travail en cours de chaque salarié (Annexe 4).

J'ai également pu échanger avec des alternants afin qu'on puisse s'entre aider sur nos tâches respectives.

Problèmes à résoudre / contraintes / limites

Au niveau de l'élaboration des tests, il y avait un grand nombre de problèmes à résoudre.

En effet, au niveau de l'ajout de classe CSS (Les sélecteurs de classe CSS permettent de cibler des éléments d'un document en fonction du contenu de l'attribut class de chaque élément), trouver le bon élément de la page web auquel on souhaite interagir n'était pas souvent simple.

Lors de la fin de l'exécution d'un test, Cypress peut enregistrer une vidéo de l'exécution ou prendre une capture d'écran pour montrer où le test s'est terminé. Cela m'était très utile pour voir les différents problèmes qui étaient à corriger.

De même, les tests qui étaient exécutés en local fonctionnaient très bien, mais lorsqu'ils étaient exécutés sur le serveur de l'entreprise certains tests ne fonctionnaient plus du tout sans trop que l'on comprenne pourquoi.

Pour cela, j'ai donc dû faire un grand nombre de modifications afin de rendre tous les tests correctement exécutables.

Démarche suivie pour effectuer les missions

Chaque jour avant de commencer, je regardais sur Redmine, qui est une application web libre de gestion de projet, les tickets qui m'avaient été attribués, afin de voir les tâches que je devais faire.

1- Conception d'un test de bout en bout

Pour réaliser les tests de bout en bout, je me suis d'abord documenté sur le Framework Cypress afin d'apprendre la syntaxe et les différentes fonctionnalités que propose ce dernier.

Pour réaliser un test de bout en bout avec Cypress, il faut donc créer un fichier PO et un fichier spec. Le fichier PO est un fichier de type Page Object qui correspond aux fonctions que nous allons créer pour réaliser des actions sur notre test et que nous pourrions réutiliser pour d'autres tests.

Puis le second fichier spec correspond au scénario que l'on souhaite donner à notre test.

Voici à quoi ressemblent les instructions d'un test de bout en bout : (fichier spec).

```
import {login, password} from "../../sasha-e2e/src/config";
import {LoginPageObject} from "../support/login.po";
import {RouteSearchPageObject} from "../support/route-search.po";
import {EMockableTest, runTestWithMockOrNot} from "../../sasha-e2e/src/support/run-test-with-mock-or-not";

describe( title: "Iaqobo air : See if the retail price is displayed correctly", fn(): void => {
    const loginPage = new LoginPageObject();
    const routeSearchPage = new RouteSearchPageObject();

    beforeEach( fn(): void => {
        routeSearchPage.postSearchJsonWithRetail();
        loginPage.visit();
        loginPage.typeLogin(login);
        loginPage.typePassword(password);
        loginPage.clickLogin();
    });

    it( title: "Iaqobo air : See if the retail price is displayed correctly", fn(): void => {

        routeSearchPage.visitIaqoboAirSearchHome(runTestWithMockOrNot(EMockableTest.SeeRetailPrice));
        routeSearchPage.assertReservationButtonToBeVisible();
        routeSearchPage.clickAllProposalsListUp();
        routeSearchPage.clickTravelReservation();
        routeSearchPage.assertRetailPriceButtonToBeVisible();
        routeSearchPage.assertPurchasePriceButtonToBeVisible();

    })
})
```

Voici à quoi ressemble les fonctions d'un test de bout en bout : (fichier PO).

```
public assertReservationButtonToBeVisible(): void {
    cy.wait('@postSearch');
    this.elements.reservationSpan().should( chainer: 'be.visible');
}

public clickTravelReservation(): void {
    this.elements.travelReservationButton().click();
}

public assertLastNameInputIsNotEmpty(): void {
    cy.wait('@postQuoteOffer')
    this.elements.lastNameInput().should( chainer: 'be.empty')
}

public clickLastTravelReservation(): void {
    cy.wait(5000);
    this.elements.lastTravelReservationButton().click();
}

public assertBodyConfirmationReservationDivToBeVisible(): void {
    this.elements.bodyConfirmationReservationDiv().should( chainer: 'be.visible');
}

public assertStatusColumnToBeVisible(): void {
    cy.wait('@postBookingsCreate');
    this.elements.statusColumn().should( chainer: 'be.visible');
}
```

Ensuite, lorsqu'on souhaite interagir avec un élément de la page web, il faut trouver les classes CSS de chaque élément, ou accéder au code de la page et ajouter une classe CSS à l'élément qu'on recherche. (Annexe 5).

Quand on ajoute une classe, on la nomme la plupart du temps par l'action que l'élément représente en mettant au début de la classe « e2e » qui correspond à end to end, pour bien comprendre que c'est une classe CSS qui est utilisée par les tests de bout en bout.

Ensuite après avoir récupéré les classes CSS de chaque élément avec lesquelles on souhaite interagir, on crée des fonctions dans notre code donc le fichier PO qui va être capable d'interagir avec les classes CSS de la page web.

De plus, lorsqu'on nomme nos fonctions, on choisit le nom en fonction de l'action qu'il représente comme cliquer, récupérer une valeur, attendre que la page se charge etc.

Ensuite lorsque le test est terminé et qu'il fonctionne correctement, il faut transférer le code sur le serveur qui possède toutes les applications et développement de l'entreprise.

Après avoir fait cela, le test va pouvoir être exécuté automatiquement sur Jenkins qui va pouvoir le lancer quotidiennement, environ 3 fois par jour pour voir si tout fonctionne correctement.

2- Exécution d'un test de bout en bout

Pour lancer un test cypress il faut exécuter plusieurs commandes :

Par exemple pour lancer un test en local il faut d'abord exécuter la commande,

```
npm run serve sasha
```

Puis,

```
npx nx e2e sasha-e2e --baseUrl=http://localhost:3000/nightly/sasha --watch
```

Et si on veut lancer les tests directement sur la vraie application qui est en ligne il faut juste exécuter,

```
npx nx e2e sasha-e2e --baseUrl=https://nightly.afidium.com/sasha --watch
```

Je me suis également aidé de plusieurs vidéos YouTube pour apprendre les expressions régulières car c'était quelque chose de tout nouveau pour moi. Le site regex 101 m'a aussi beaucoup aidé pour simuler des expressions régulières.

3- Horaire de travail et réunion quotidienne

Durant mes 6 semaines de stage, mon horaire de travail du lundi au vendredi était 9h00 - 17h00 avec une heure de pause le midi.

Chaque jour, à partir de 16h00 j'étais convié à participer à la réunion quotidienne de l'entreprise. En effet, dans cette réunion tout le monde devait parler chacun son tour, afin d'expliquer ce que chacun avait fait durant sa journée et parler des problèmes rencontrés.

J'ai également été convié à participer aux réunions de début de sprint qui permet de voir les objectifs fixés durant le mois (un sprint dur 4 semaines), puis j'ai participé aux réunions de fin de sprint afin de présenter le travail réalisé.

Les outils et services informatiques qu'utilise l'entreprise

L'entreprise utilise plusieurs logiciels, comme Slack qui est une messagerie instantanée afin de discuter avec tous les membres de l'entreprise (Annexe 6). Il y a également PhpStorm qui est un IDE très utilisé par l'entreprise, pour le développement.

Il y a BitWarden qui est une extension pour navigateur web afin de garder tous les mots de passe dans une même application, cela a pour but de se connecter beaucoup plus rapidement aux différentes interfaces web qu'utilise l'entreprise. Il y a aussi OpenVPN GUI qui est un VPN pour se connecter aux différents serveurs de l'entreprise.

Pour voir les différentes missions à faire, l'entreprise utilise Redmine qui est une application web libre de gestion de projet, cette application est très utile car elle permet de voir où se situe l'entreprise au niveau de l'avancement des différentes missions.

Et pour exécuter les différents tests de bout en bout sur les différentes applications, l'entreprise utilise Jenkins. Afin de réaliser des réunions, l'entreprise utilise l'application Zoom qui est très simple d'utilisation et qui propose un grand nombre de services très utiles.

Pour finir, il y a TortoiseHg qui est un client du logiciel de gestion de versions Mercurial afin de se retrouver dans les différentes versions des applications et au niveau des modifications. On peut voir les différents commit qui ont été fait avec le nom de la personne puis avec une description du travail. (Annexe 7).

Évaluation des réalisations et des compétences mobilisées

1- Adéquation du travail

Après avoir terminé les différentes missions qui m'ont été confiées, l'équipe de développement était très intéressée par mes réalisations qui leur permettaient de gagner énormément de temps au niveau de leur quotidien. En effet mes tests étaient capables d'analyser les applications web et de vérifier si tout était en ordre et qu'il n'y avait aucun problème technique.

Cependant, il y a en effet des améliorations à apporter au niveau des tests que j'ai pu réaliser. Car pour certains tests, il faudrait les optimiser afin qu'ils ne « plantent » pas si les développeurs de l'entreprise décidaient d'apporter des modifications aux applications.

2- Rétrospective des compétences exploitées

Afin de mener les missions qui m'ont été confiées, j'ai dû mobiliser un grand nombre de compétences. En effet, il faut d'abord être organisé ce qui est très important pour être le plus productif possible. J'ai également dû faire preuve de rigueur, être à l'écoute, être sérieux et savoir rester professionnel.

Les missions que j'ai réalisées m'ont permis d'améliorer mes connaissances dans le domaine de l'informatique, j'ai appris beaucoup de choses sur des langages, comme le Type Script (TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un sur ensemble syntaxique strict de JavaScript).

J'ai pu apprendre Cypress (Framework de test de bout en bout, qui exécute des tests sur notre application s'exécutant dans un vrai navigateur, interagissant avec elle comme le ferait un utilisateur), puis revoir le Php qui est un langage dynamique via un serveur HTTP.

Pour mettre en place des propositions d'amélioration il n'y a pas vraiment de compétence spécifique à avoir, il suffit d'être clair dans ce que nous souhaitons expliquer pour améliorer quelque chose. Il suffit également de maîtriser le sujet concerné et de se documenter, afin de trouver de nouvelles méthodes d'utilisation d'un langage.

Conclusion :

Ce stage m'a énormément apporté et m'a conforté dans ce choix de carrière.

J'ai pu discuter avec un grand nombre de personnes faisant partie du domaine de l'informatique.

L'équipe de développement était très sympathique avec moi, je n'ai eu aucun problème pour m'intégrer.

Les missions qui m'ont été confiées m'ont permis d'obtenir de nouvelles connaissances, ce qui est très important dans l'informatique car c'est un domaine qui ne cesse d'évoluer en permanence.

Cette expérience a été encore une fois très enrichissante car j'ai pu participer à plusieurs projets très intéressants.

J'ai pu apprendre au fur et à mesure, que ce soit à travers des recherches personnelles, ou en échangeant avec l'équipe de développement.

J'ai développé mes capacités d'adaptation et d'autonomie dans mon travail et le fait de travailler en distanciel ne m'a posé aucun problème j'ai su m'organiser afin de ne pas être perdu.

J'ai eu la chance d'évoluer dans un environnement très favorable, avec une très bonne équipe et une très bonne ambiance.

De manière générale, l'organisation de l'entreprise m'a permis d'évoluer positivement et rapidement.

Bibliographie

- <https://stackoverflow.com/>
- <https://www.youtube.com/>
- <https://docs.cypress.io/guides/references/best-practices>
- <https://angular.io/>
- <https://angularjs.org/>
- <https://www.typescriptlang.org/>
- <https://developer.mozilla.org/fr/docs/Web/JavaScript>

Annexes

Annexe 1 :

| | | | | | | | |
|--------------|-------------------------------|---|-------------------------------|---|--|--------------------|-------------|
| 1 juin 2022 | CDG → EZE 13:40 → 06:30 +1 |  | 🕒 21h50min 🛫 Soute 1 PC | 1 escale CO ₂ : 2 670 kg | YUCANGoo •5,69/10 (5,88 moy.) | 3 501,65 € | Total achat |
| 12 juin 2022 | EZE → CDG 21:15 → 06:10 +2 |  | 🕒 27h55min 🛫 Soute 1 PC | 1 escale Main Cabin CO ₂ : 2 907 kg | | 10 861,65 € | Total vente |

Détail des prix ▼ Détails YUCANGoo ▼ Itinéraire au format texte Détails des vols

[Autres offres](#) [Voir cette proposition](#)

Public
TRAVELPORT
CO₂: 5 577 kg
Annulation : 100%
Échange : 180,00 €

Annexe 2 :

Résumé de la recherche

15 mars 2022 PAR → NYC 25 mars 2022 NYC → PAR 1ADT Source TRAVELPORT

Sauvegarde d'une offre

Sauvegarde d'une offre réussie

Annexe 3 :

Résumé de la recherche

11 mars 2022 PAR → BCN

21 mars 2022 BCN → PAR

Types de tarif Public, TourOperator, NegoFare

Source TRAVELPORT

Réservation d'une offre sauvegardée

La réservation a été effectuée avec succès.

Annexe 4 :

Voici un extrait de la documentation que j'ai pu faire sur Redmine.

Tickets Redmine et commits HG

Pour créer un ticket :

Se rendre sur le Kaban du sprint actuel, puis cliquer sur le « + » situé en haut à gauche de l'écran, puis cliquer sur « Nouvelle demande »

Technique

Accueil Ma page Projets Easy Gantt Liste des contacts Affaires

Nouvelle demande

Search by subject

Nouvelle catégorie

Nouvelle version

Saisir temps

Nouvelle annonce

Nouvelle page wiki

Effacer

Plaisir écran

Nouveau - New (38) 219.00h Candidat (7) 49.00h En cours - Work in progress (19/6) 159.00h A Valider - Ready for test (6/6) 5.50h A Patcher (2/6) 2.00h Fermé - Closed (15) 48.00h

Project #28406: Tests E2E Iqabo/Sasha 202112

Sasha

Evolution #29949

Test e2e Iqabo Air

Mis-à-jour: 20/01/2022 11:23

Sprint: sprint-2022-A-Technique

☒ sprint-e2e-specs

☒ test-quick-e2e-specs

☐ create-cancel-booking-lufthansa.e2e-specs

☐ create-cancel-booking-a2e-specs

☐ create-cancel-booking-osman.e2e-specs

Eleonore Ayari

Sasha

Evolution #29954

Test e2e Sasha

Mis-à-jour: 19/01/2022 14:43

Sprint: sprint-2022-A-Technique

☒ sasha-booking-display.e2e-specs

☐ sasha-booking-section-booking.e2e-specs

Lucas Aguesse

Appli. Air

Maintenance #28637 (7.00h)

Splitter les tests E2E par projet

Forfait: IJ

Mis-à-jour: 19/01/2022 09:43

Sprint: sprint-2022-A-Technique

Lucas Aguesse

Appli. Air

Project #29936 (8.00h)

Intégration tests E2E cypress dans Jenkins

Mis-à-jour: 20/01/2022 17:00

Sprint: sprint-2022-A-Technique

Lucas Aguesse

Technique

Evolution #29990 (1.50/2.00h)

[e2e] test de login sur Osman

Mis-à-jour: 20/01/2022 17:00

Sprint: sprint-2022-A-Technique

Jérôme Ruillier

Sasha

Evolution #29956

Test Admin

Mis-à-jour: 19/01/2022 09:38

Sprint: sprint-2022-A-Technique

☒ admin-title.e2e-specs

Eleonore Ayari

#29443 #29891 #29914

#29858 #29874

Suivi de la demande

Lorsque le ticket est en cours de réalisation vous devez indiquer votre temps passé et le % réalisé au fur et à mesure. Cliquer sur saisir temps situé en haut à gauche du ticket (indiqué par la flèche bleue).

Evolution #29957

Modifier Saisir temps Surveiller Copier

[doc] [hg] Doc après réu sur la liaison commits / HG

Ajouté par Erwann Jallier il y a 3 jours. Mis à jour il y a moins d'une minute.

Statut: En cours - Work in progress

Priorité: Bas - Low

Assigné à: Lucas Aguesse

Versions cible: -

Début: 18/01/2022

Échéance: -

% réalisé: 10%

Lorsque le ticket est en cours vous devez indiquer votre temps passé et le % réalisé au fur et à mesure.

Evolution #29990

Appli. Air - Project #28406: Tests E2E Iqabo/Sasha 2021

[e2e] test de login sur Osman

Ajouté par Lucas Aguesse il y a environ 18 heures. Mis à jour il y a 14 minutes.

Statut: A Valider - Ready for test

Priorité: Normal

Assigné à: Jérôme Ruillier

Catégorie: -

Versions cible: Technique - STOCK1.16 (next)

Début: 20/01/2022

Échéance: -

% réalisé: 70%

Temps estimé: 2:00 h

Temps passé: 1:30 h

Patched Version: -

Owner: -

Focus: -

Technique - STOCK1.16 (next)

70%

0%

10%

20%

30%

40%

50%

60%

70%

80%

90%

100%

Non

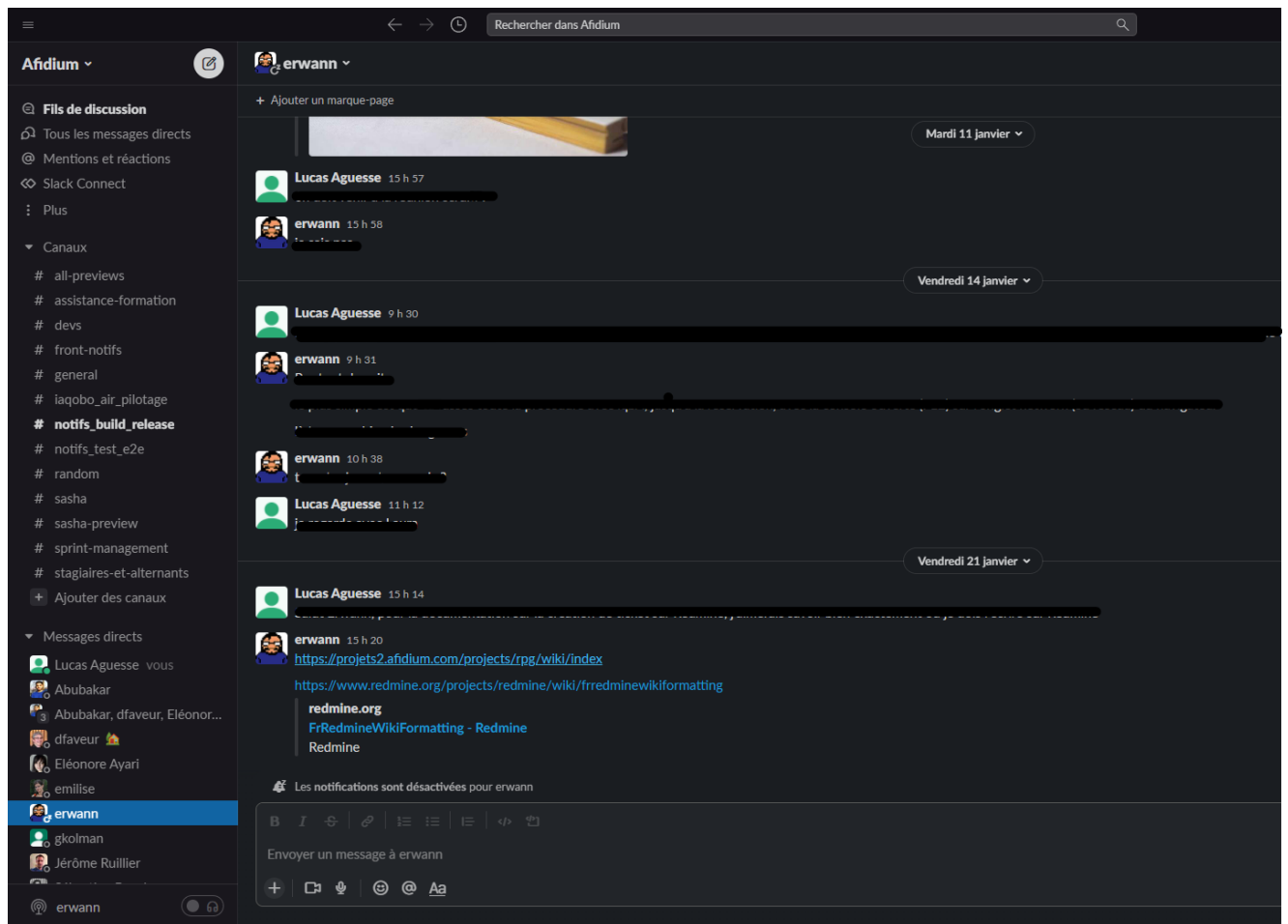
Annexe 5 :

```
export class RouteSearchPageObject {
  private elements = {
    iaqoboAirSearchHomeButton: () => cy.get('home-app .app-links a.iqbAir'),
    reservationSpan: () => cy.get('.fromto'),

    travelReservationButton: () => cy.get('.quote-result:nth-child(1) booking .e2e-select-quote-btn'),
    lastNameInput: () => cy.get('.e2e-last-name-field input'),
    lastTravelReservationButton: () => cy.get('.e2e-booking-btn'),
    bodyConfirmationReservationDiv: () => cy.get('body'),
    statusColumn: () => cy.get('.status-column'),
    retailPrice: () => cy.get('.e2e-booking-price-retail'),
    purchasePrice: () => cy.get('.e2e-quote-result-price-cost'),
    selectThisProposalButton: () => cy.get('.e2e-select-offer'),
    selectAllProposalsListUp: () => cy.get('.e2e-list-up'),
    TheOfferId: () => cy.get('.e2e-input-booking-id'),
    successSaveOffer: () => cy.get('.e2e-success-save-offer .success'),
    successSaveBookingOffer: () => cy.get('.e2e-success-save-booking-offer'),
    mockIdInput: () => cy.get('.e2e-mock-input'),
    sashaInMainNavMenu: () => cy.get('.sashaHome'),
    clickFirstBooking: () => cy.get('.e2e-list-booking mat-list-option:nth-child(2) a.e2e-confirm-first-booking'),
    getFirstBookingCode: () => cy.get('.e2e-list-booking mat-list-option:nth-child(2) a.e2e-confirm-first-booking'),
  }
}
```

Les classes CSS sont représentées en vert

Annexe 6 :



Annexe 7 :

Depôt Aide

| Grphe | Rev | Branche | Description |
|-------|--------|--------------------|---|
| | 54450+ | default | ★ Répertoire de travail ★ |
| | 54450 | default | default tip #28462 config changing old files as newer files : AirParams.php |
| | 54449 | default | #28462 config changing old files as newer files : WenjiUIYiGlobalParams.php, WenjiUIAirFareOptDashboardParams, |
| | 54448 | default | Étiquette WTB1.21 déplacée à la révision 62917c86f03f (de la révision 6597a9866bf3) |
| | 54447 | watabi-stable-1... | watabi-stable-1.21 WTB1.21 #29940: remove modal with dom and JS, on sequence AND block |
| | 54446 | default | #30120: fix errors *attributes on same element |
| | 54445 | default | #29940: remove modal with dom and JS, on sequence AND block |
| | 54444 | default | Updates translation messages |
| | 54443 | default | #30120: Ajouter une permission pour permettre de ne pas afficher les encours |
| | 54442 | default | Étiquette AIR1.57 déplacée à la révision 78eda660fb1c (de la révision 366dafd50ffd) |
| | 54441 | air-stable-1.57 | air-stable-1.57 AIR1.57 #30117: check eligibility after prov stus during issuance ... |
| | 54440 | default | #30117: check eligibility after prov stus during issuance |
| | 54439 | default | Étiquette AIR1.58 déplacée à la révision a61ee5a9fd52 (de la révision 809384b4a053) |
| | 54438 | air-stable-1.58 | air-stable-1.58 AIR1.58 Updates install/composer.json to use "slim/pdo": "v2.2.0" instead of "dev-master". (Build w |
| | 54437 | default | Étiquette AIR1.58 déplacée à la révision 809384b4a053 (de la révision 809384b4a053) |
| | 54436 | default | Étiquette AIR1.57 déplacée à la révision 366dafd50ffd (de la révision 366dafd50ffd) |
| | 54435 | default | Étiquette AIR1.58 déplacée à la révision 809384b4a053 (de la révision 48dfc56f776) |
| | 54434 | air-stable-1.58 | #30107: Problèmes sur la synchro des billets ... |
| | 54433 | default | Étiquette AIR1.57 déplacée à la révision 366dafd50ffd (de la révision bfb79577832c) |
| | 54432 | air-stable-1.57 | #30107: Problèmes sur la synchro des billets ... |
| | 54431 | default | Updates tag IQBAD4T2021.1M12 to changeset 4ffae20551e9255cc16f178168e2b4f5c5ae7fe4 |

| Auteur | Âge | Étiquettes | Phase |
|-----------------|------------|------------|--------|
| Lucas Aguesse | maintenant | | |
| Lucas Aguesse | 2 jours | tip | public |
| Lucas Aguesse | 2 jours | | public |
| Erwann Jallier | 2 jours | | public |
| Erwann Jallier | 2 jours | WTB1.21 | public |
| Erwann Jallier | 2 jours | | public |
| Erwann Jallier | 11 heures | | public |
| JenkinsCI | 12 heures | | public |
| ccremois | 16 heures | | public |
| Emilise Prim | 19 heures | | public |
| Emilise Prim | 19 heures | AIR1.57 | public |
| Emilise Prim | 19 heures | | public |
| Emilise Prim | 20 heures | | public |
| Emilise Prim | 20 heures | AIR1.58 | public |
| Emilise Prim | 20 heures | | public |
| Emilise Prim | 20 heures | | public |
| Emilise Prim | 20 heures | | public |
| Emilise Prim | 20 heures | | public |
| Emilise Prim | 20 heures | | public |
| Emilise Prim | 20 heures | | public |
| Jerome Ruillier | 21 heures | | public |