

JePilote



*Je pilote mon **entreprise***

Introduction

In the continuity of my formation as a third year student at Epitech Bordeaux, I had to take part in a 4 to 6 months internship to validate my academic year. I rediscovered the professional world since my first internship in 2016 and got used to it.

JePilote is a web accountant service that needed to convert some of their code into an API and make new modules, they chose Symfony4. It's a great alternative since you can deploy it easily and you can play around with it which offer plenty of possibilities.

In this Internship report, I will be explaining to you all the tasks I had to take on and talk about the issue I encountered, and how I got around them.

Table of contents

1. Deontology, introduction and pursuit of the mission
 - 1.0 Introduction.
 - 1.1 Company context
 - 1.2 Company team.
 - 1.3 Context and projects overview.
 - 1.3.1 Crawlers and functional tests.
 - 1.3.2 Bank synchronization.
 - 1.4 Technology used and their usage.
 - 1.5 Work methods.
 - 1.5.1 Daily SCRUM.
 - 1.5.2 TAIGA.
 - 1.6 Conclusion.
2. Prolongation asking
 - 2.1 Preambular.
 - 2.2 Mail.

Part 1:

Deontology, introduction
and pursuit of the mission

1.1 Company context

JePilote is a company made up of accountants and associates, all of whom have made careers in law firms. The decision to develop online tools was the answer to a strong need expressed by business leaders: getting control back to have real-time numbers to manage their business.

With new technologies, it was finally possible to share information with our customers. But it was necessary to create a collaborative platform with which any business feels at ease. This was our starting creed. Jepilote provides maximum flexibility to facilitate the accountant / business man relationship.

It is therefore:



Figure 1 : JePilote statistics



Figure 2 : JePilote logo

1.2 Company team:

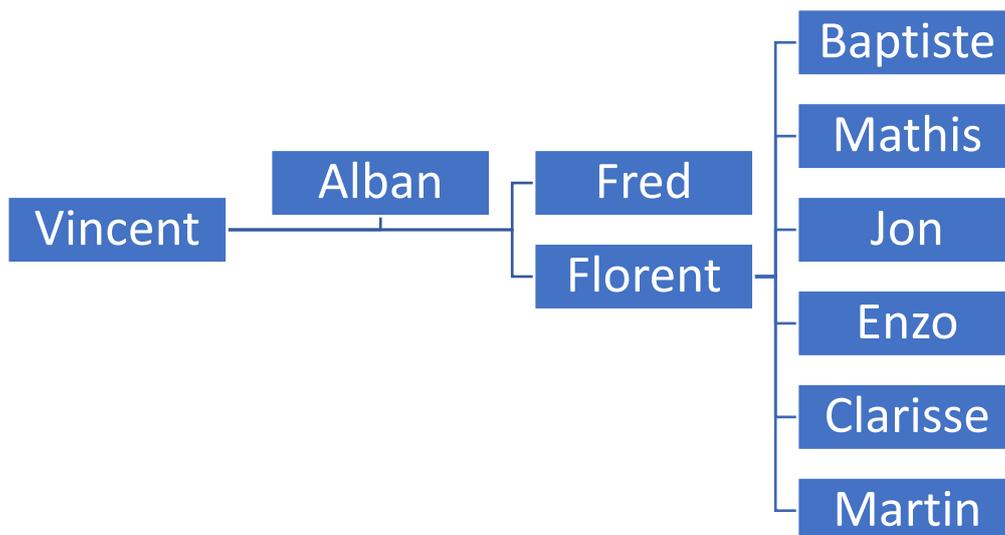


Figure 3 : company organization chart

Vincent as the general director and accountant

Alban as co-founder and accountant

Fred as accountant and after sales service

Florent as IT manager and project manager

Baptiste as an intern developer

Mathis as an intern developer

Jon as an intern developer

Enzo as an intern developer

Clarisse as an intern graphic artist

Martin as an intern developer

1.3 Context and projects overview

During my internship at JePilote, I have been given different tasks to accomplish, all resolving around functional tests and bank synchronization. This working environment being very new to me, I had to start off by getting familiar with all the tools that had been given to me.

1.3.1 Crawler and functional tests

After learning everything I needed to know to take on the project, my mission was to write a crawler that will use a browser (and not a virtualized one) to click on buttons, send forms and more on the website to test if everything is working properly. The purpose of this project is to check if a new update on the website has broken some basic functionalities of the website like buttons, forms, dropdown and links. I used a lot of technology on this project, testing multiple crawler to find the easiest and the most adapted one for this project

For the internship report, I'll use an array to explain my choice for the crawler I finally decided to use.

Name\Capacity	Can crawl an external website ?	Use a "real" browser (not virtualized)	Can read/execute JS.
SymfonyDomCrawler	NO	NO	NO
NightmareJS	YES	NO	YES
CasperJS	YES	NO	YES
FantomJS	YES	NO	YES
Horseman	YES	YES	NO
Panther	YES	YES	YES

Figure 4 : Table of choices for my crawler

I decided to use Panther, a PHP class for Symfony4 who that overloads SymfonyDomCrawler methods, so it was a win-win situation: I already got the experience of SymfonyDomCrawler and he has all the requirement I wanted for my crawler.



Then I begin to "parse" the website with my crawler and manage to make 2 functional tests that test if a button appears when the user selects a specific invoice type and a second one to test if the TVA is indeed decreasing when you pay an invoice in multiple payment.

```
// 1. Create client
$client = \Panthers\Client::createChromeClient();
$crawler = $client->request('GET', 'http://test-melodie.jepilote.com');
```

Figure 5 : Creating a Panthere client using Chrome browser

The tests returns a binary value for success or failure and takes a screenshot for every action made, that is very useful for finding issues on a website and debugging.

Panthere crawler is able to fill form, click buttons, drag/drop elements, use keyboard shortcuts, download videos of a website and much more.



Figure 6 : Screenshot taken by my crawler after filling a form

1.3.2 Bank synchronization

This internship project aims to create a non-free module, based on bank account information and accountant services provided by JePilote and BudgetInsight.

The goal was to fetch users bank information like IBAN, BIC, balance amount, bank establishment and many more and process them before storing them in JePilote's database.

With BudgetInsigh and the consent of the user, we can safely access his or her financial data from their bank in order to provide them other innovative banking services that are not available at a traditional/native bank.

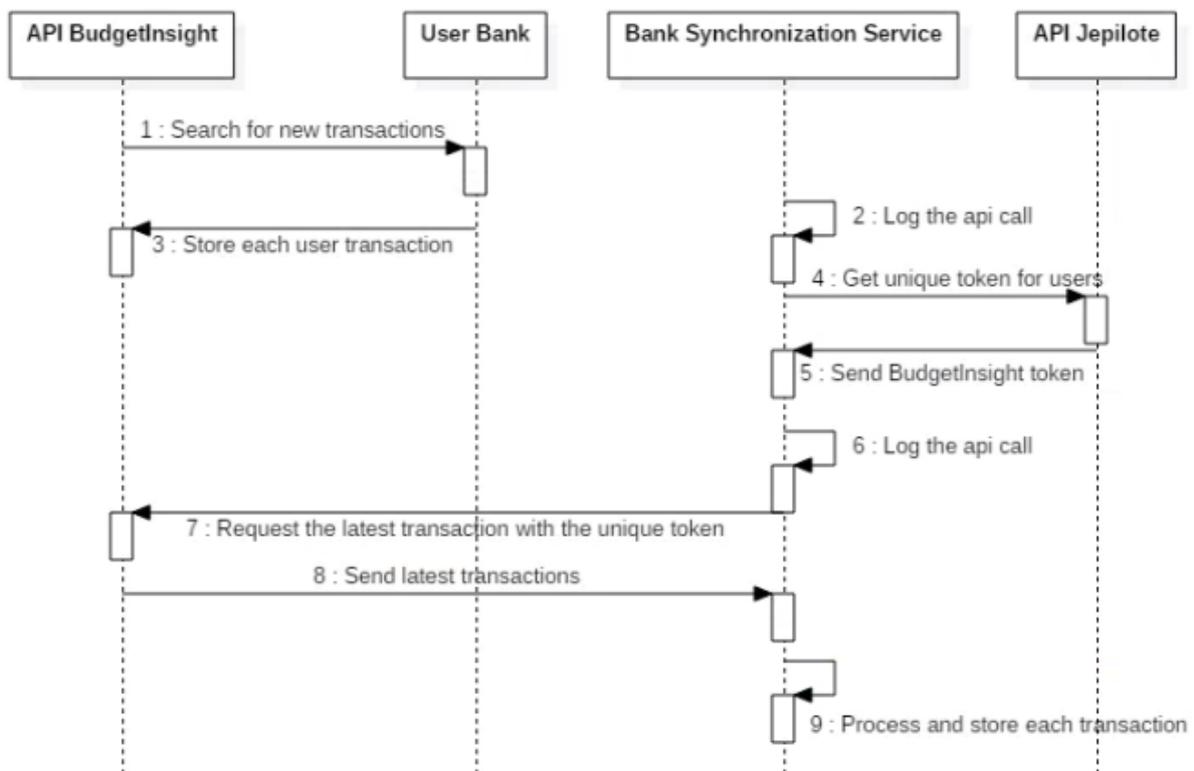


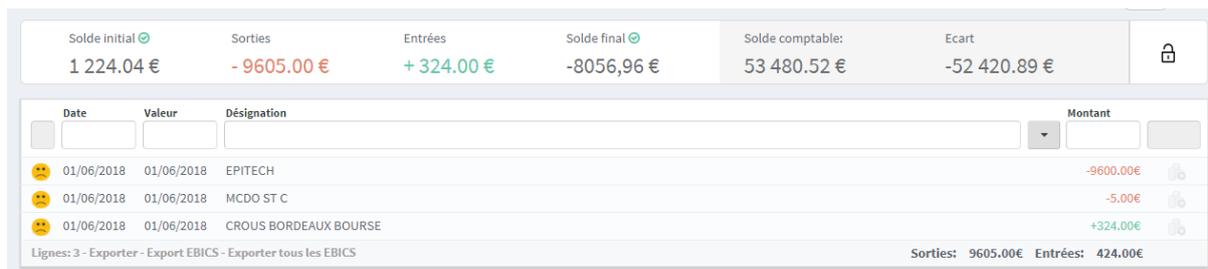
Figure 7 : sequential scheme of a classic request to get user bank data

I provide many methods, functions and commands called by CRON every minute to fetch all the transactions made by users and update the database consequently.

The goal was to fetch all the users data to provide them new services to manage their bank account and facilitate many tasks for company accountants and business men.

I was able to call BudgetInsight's API from my back-office Symfony4 project using Guzzle or Curl and a unique bearer token from the company's API.

The REST API from BudgetInsight returns a JSON with user bank information, then I was able to process each JSON to reach my goal: get the user's bank data for later use from the main website of JePilote.



The screenshot displays a bank account interface with a summary bar at the top and a transaction list below. The summary bar includes fields for initial balance, outgoing payments, incoming payments, final balance, accounting balance, and variance. The transaction list has columns for date, value, designation, and amount. Three transactions are visible, all dated 01/06/2018, with designations EPITECH, MCDO ST C, and CROUS BORDEAUX BOURSE. The interface also includes search filters and export options.

Statut	Date	Valeur	Désignation	Montant	Statut
🟢		1 224.04 €	Solde initial		
🔴		- 9605.00 €	Sorties		
🟢		+ 324.00 €	Entrées		
🔴		-8056,96 €	Solde final		
🟢		53 480.52 €	Solde comptable:		
🔴		-52 420.89 €	Ecart		

Statut	Date	Valeur	Désignation	Montant	Statut
🟡	01/06/2018	01/06/2018	EPITECH	-9600.00€	
🟡	01/06/2018	01/06/2018	MCDO ST C	-5.00€	
🟢	01/06/2018	01/06/2018	CROUS BORDEAUX BOURSE	+324.00€	

Lignes: 3 - Exporter - Export EBICS - Exporter tous les EBICS Sorties: 9605.00€ Entrées: 424.00€

Figure 8 : interface of a bank user account with JePilote services

One of my tasks on the bank synchronization was to send an email to the client's company if we fail to access their banking data.

Most of the time, if a company changes their banking account password, they forget to also change the password on JePilote's services. By checking the logs of all transactions in real time, I am able to verify if any bank sent me back an error like "WrongPass", if so, I call a mailing service from JePilote's API to send an email to the client company.

Thanks to the logs of all transactions from all clients, I was told to search if any transaction was a new client, and if it was one, to create a record into the user database. Then, in a background process, I can have access and get all

transactions of the customer since the opening of his banking account and add them to all the user transactions database.

Since JePilote database fetches all the transactions of their users since the opening of their bank account, they can provide services for new customers, so the company accountants do not lose any data and do not have to type them by hand.



Figure 9 : BudgetInsight logo

1.4 Technology used and their usage

In this internship, I was asked to create and maintain many projects with a lot of new technologies for me.

Here are some of them and their function:

To manage SQL servers and create databases or queries I was told to use SQLyog a heavy SQL GUI client.

To me, it was the best discovery I made during my internship.



Figure 10 : SQLyog logo

For many of my projects, I used a Symfony4 environment, which is a great framework and was very useful when you know how to use it. I was at first very uncomfortable with this framework, but the official documentation is very helpful and complete.



Symfony

Figure 11 : Symfony4 logo

In addition of Symfony4, I was using Doctrine with its ORM and Dbal. I wasn't very comfortable with it until I started to modify table names and columns and manage to recreate a mapping between my query and my database using Doctrine.



Figure 12 : Doctrine logo

Then, I was told to modify and improve some existing code in CodeIgniter, another framework like Symfony4 but the community is not as big as Symfony so it was a bit complicated to debug and work on existing code that you don't know.



Figure 13 : CodeIgniter

For my Crawler, I used 3 technologies named CasperJs, NightmareJs and PhantomJs



Figure 14 : CasperJS logo



Figure 15 : NightmareJS logo



Figure 16 : FantomJs logo

They are frameworks specialized for crawl or scrap website and I was using them to create functional tests.

I finally used a Github project named "Panthere"



Figure 17 : Panthere logo

I programmed mostly in PHP using MVC architecture with a lot of SQL query through Doctrine and SQLyog.

I used JavaScript and Ajax for the crawler project.

1.5 Work method

1.5.1 Daily SCRUM

This is an agile work method for managing work, it is designed for teams of three to nine developers, so as a 5 members developer group, we used it to break our work into actions called sprint. Each sprint contains tasks that have to be complete, and those will be assigned to each developer by the project manager. It's a very useful work method for tracking progress and get a global view and understanding of the project. Every day, all the developers and the project manager regroup in a place, in our case the kitchen, for a 15 minute stand-up meeting, called daily scrums where we expose our problems and successes to other developers who can suggests you a solution.

1.5.2 Taiga

Taiga is a project management platform for agile developers, it's designed for teams who use the SCRUM method, and provide them boards called "Kanban" where each developer can see all the tasks in progress and to whom they have been assigned.

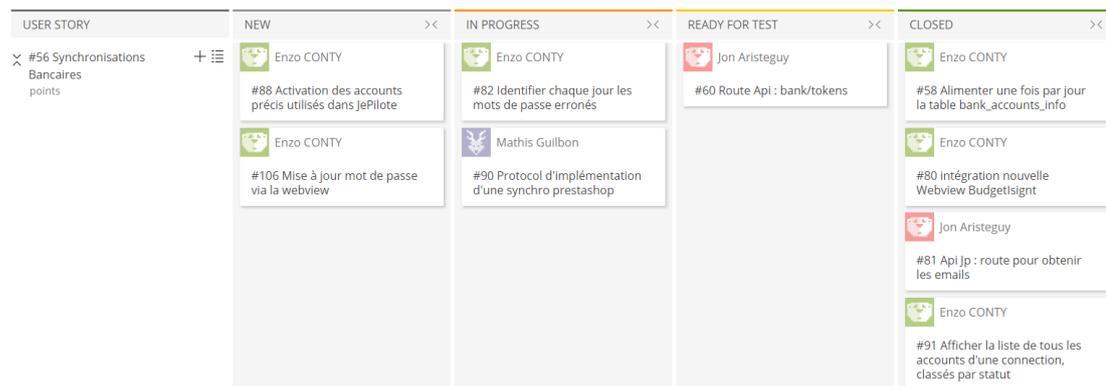


Figure 18 : Kanban of the task I was assigned

The project manager creates and assign tasks to a developer, he/she can also assign issues and more. Taiga is the visualization of the SCRUM method, it provides a global view of the work in progress for each developer.

1.6 Conclusion

It was a very nice experience working at JePilote. I was very well welcomed and it motivated me the first days at the internship. Taking part in this internship gave me the opportunity to know my strengths better as well as the areas in which I had to improve. It's the first time in my life I had to create a new project from scratch and get responsibilities in my work and it gave me a lot of confidence.

My experience was a great one. I got to work with other interns to link our individual projects. I learned a lot in the way and I think it was all worth it. I loved taking on tasks that challenged my abilities.

All in all, the internship was a great experience and confirmed my will to work in the computer science field.

Table of figures

Figure 1 : JePilote statistics	5
Figure 2 : JePilote logo	5
Figure 3 : company organization chart.....	6
Figure 4 : Table of choices for my crawler.....	8
Figure 5 : Creating a Panthere client using Chrome browser	9
Figure 6 : Screenshot taken by my crawler after filling a form	9
Figure 7 : sequential scheme of a classic request to get user bank data	10
Figure 8 : interface of a bank user account with JePilote services.....	11
Figure 9 : BudgetInsight logo	12
Figure 10 : SQLyog logo.....	13
Figure 11 : Symfony4 logo.....	13
Figure 12 : Doctrine logo.....	14
Figure 13 : CodeIgniter.....	14
Figure 14 : CasperJS logo.....	14
Figure 15 : NightmareJS logo	14
Figure 16 : PhantomJs logo	14
Figure 17 : Panthere logo.....	15
Figure 18 : Kanban of the task I was assigned	16

Part 2:

Prolongation asking

2.1 Preambular

The completion of my internship is close, so I ask to my internship master, Florent Cholet, to remain my internship for four more month based on the work I did, the skills I acquired and my personal feelings.

2.2 Mail

From: enzo.conty@epitech.eu

To : florent.cholet@jepilote.com

Subject: Internship Enzo Conty

Dear Mr.Cholet

Currently in internship for more than 4 months in JePilote, I will contact you today in the hope of renewing my intership for the same duration, 4 months.

Indeed, during my internship, I confirmed my desire to work in the field of computer science. I was able to confirm that my professional aspirations are in agreement with your company and I wanted to continue to work in the objective that was fixed at my arrival: create a new module for accountant with BudgetInsight and JePilote banking service.

I was able to work for 4 months on the project and create a large part of his architecture. It is for these reasons that I wanted to work 4 more month with the company to complete and finish the project.

That's why I wanted through the content of this email, my motivation to work on bank synchronization.

Sincerely,

Enzo Conty.