MEILHAC
Vivien

Greenbureau

Third year intership

Year (2011/2012)

Contents

A. First Part / The company

Introduction

I. Presentation of Greenbureau

- The startup
- The technical environment
- The server architecture

II. Introduction to the scraping

- The scraping
- Creation
- Maintenance

III. The website

B. Second Part/ New project

A) First Part / The company

Introduction

The purpose of this document is to give a quick overview of the startup Greenbureau. Some of the following informations might involve by the time of the writing. First a presentation of the startup will be made, next a listing of the technological environment, then a description of the global architecture. In another point a brief description of the scraping at Greenbureau will be done and finally a quick overview of the front part. This document is oriented on the technological side of the project.

I Presentation of Greenbureau

The startup

Greenbureau was created on the 15th of july 2011 by ex employees of SFR. Greenbureau is also the first free service to centralize bills in a secure way for personal individuals. Founded by Mathieu Delobelle, Yan Tamalet, Florian Nicourt and Frédéric Dermer Greenbureau raised 300 000 Euros in september 2011 thanks to a private investor. Greenbureau has currently several partnerships with famous brands such as SFR, Direct Energie, Numéricable and more soon.

The Greenbureau team is:

- Mathieu Delobelle, cofounder, in charge of the communication and customer relatioship
- Yan Tamalet, cofounder, in charge of the partnerships
- Florian Nicourt, cofounder, technical director
- Frédéric Dermer, cofounder, in charge of the user experience
- Goulwen Reboux, lead developer
- Yann Guibet, developer

Moreover Greenbureau also hired interns: Augustin Pasquini, Ewa Nowak.

Since Greenbureau is a young company, there is a good atmosphere and dynamic that we could not find in a classic company.

The technical environment

The technical environment is exclusively composed of python solutions.

The tools used are:

- Django (framework)
- Selenium (automates browser, used to scrape)
- Celery(asynchron tasks)
- RabbitMQ (backend for Celery)
- Postgresql (relational database)
- MongDB(scalable database)
- Gunicorn(application server)
- Nginx (Http server)
- HAProxy (proxy & load balancing)
- Stunnel(proxyfier SSL)
- Varnish (cache)

Server architecture

The current architecture of Greenbureau is composed of a proxy server which dispatched the incoming http connections on two different front servers. These servers communicate with a server containing the Postgrsql database. The front server also communicates with the back server via Celery which contains the Mongo database (for the file storage). The back server launches the scrapings on two specific servers used for this task.

II Introduction to the scraping

The scraping

The scraping is a technique to extract informations from a website, therefore the programs will simulate a human browsing of the website. The scraping technical is relatively new, so the software solutions are limited either in number and features.

After several tried on many solutions,Greenbureau chosed to use Selenium. Webdriver (selenium2) gives you the possibility to control the Webdriver. It was at first made to test the web interfaces but can be also an interesting tool to scrap websites.

The primary feature of Selenium is the webdriver API available in different languages (including Python). To control your Browser, Firefox for instance, Selenium injects an .xpi extension in Firefox named Webdriver to control the browser. However firefox must be launched in a graphic environment, so we use xvfb to create a buffer imiting a graphic session, we finally only need to redirect Firefox in the xvfb to obtain a manageable firefox in command lines.

This is the different steps to scrap a website at Greenbureau:

- Connection to the website and successfully sign in
- Collect the personal information (user informations)
- Collect the user's contract(s)
- Download each bill(s) of each contract(s)

Most of the time the authentication systems are easy to pass. They use only two input fields (username and login). We only need to set the login and password in the right fields and simulate a click to submit the connection. However some authentication systems use what we call a numeric pad. A numeric pad is an array implementing the Turing test within images randomly set. Unfortunately this system is not perfect, and there are still some ways to sign in.

When this security is badly implemented, the only thing you need to do is to spot the Javascript code to understand how it works and reproduce it to simulate the user clicking on the different images.

When a numeric pad is properly implemented, we have to download the image which makes the numeric pad. Then the script, thanks to the library Imagemagick we will be able to make the images match with the numbers.

To collect the personal informations, the contract informations and the bills we have to simulate the actions of the user thanks to the css selector or the xpath. The challenge of this task is to try to find a way to make the auto-navigation stable. Indeed, a lot of website are old or do not respect the rules of web developing, so it might be hard to find a solution which can last if one of the selector is update.

Creation of a scraping script

This is the different steps to scrap a website:

- getting the logins
- Studying the website
- writing the script
- add the new provider in the tests

Before creating a new script we must obtain some logins to be able to sign into the website and study it. To do so, usually users give us their personal informations when they want us to add a new provider.

The step “studying the website” which is reproducing the actions of the user and get the more accurate selectors for an efficient script can more or less take time. Indeed, it will depend of the level of protection, the quality of the html and css, the ergonomy of the website. For instance, in general, health insurance websites will take more time than providers’ websites.

When the programmer knows the way of the websites works and where to go to collect the informations he can write the script. The script is a Python program using Selenium and a Greenbureau’s abstraction called Moonchild. With the Python package pdb (which is a debugger) it is easy to put the program in “stand by” step by step look for the right selector. Pdb will provide a Python shell, truly useful to make a lot of test and see the result on the Firefox window. This window is launched automatically when a script is created and launched. For the same reasons enumerate above it can take more or less time to write the script.

When the script is done and ready we have to try with different user account to see if it is generic. Once this step is validated, we set the logins we used in a personal secure database. This database will help us to do tests or if a problem is detected for a scraping to launch a test only for the website we want to work on.

This is few lines from a script:

```
def scraping_task(keyword):
driver = webdriver.Firefox()
driver.get("http://www.google.com")
driver.find_element_by_css_selector('#gbqfq').send_keys(keyword)
driver.find_element_by_css_selector('#gbqfb').click()
driver.implicitly_wait(4)
res = driver.find_element_by_css_selector('#ires li.g a').text
driver.close()
return res
```

Maintenance of the scripts

A script is written to follow exactly the action we wanted to do on the website. To find some html elements we use the css and html tags. So if the website has been changed (change the name of a css element or small changes on the html) the script will not for the element and will crash.

Greenbureau set a strategy to fix these errors in the smallest delays. Every morning, the errors of scraping are collected, counted and ordered in a daily report, which is send to the team. The person in charge of the exploitation will create tickets and assign them to a developer.

When a developer got a new ticket, he tries from his own machine to launch the scraping in the login of the user. The scraping is supposed to fail, it will also give more details to the developer about the error. The developer just needs to fix the problem working step by step with pdb. Usually It take between 5 minutes and 2 hours to fix a script.

III The website

The website is made thanks to the framework Django (also made in Python). Django is a perfect example of a project using the MVC (Model View Controller) pattern even though Django's referred itself as a MTV (Models Template Views) pattern. The models are used to get the informations from the database. The models are called by the views. The Views contains the "real" informations about the html page such as useful containers, or the result of a model. The view will call a template which contains only html code. Django allows to transfers variable from views to template.
Greenbureau website follows the MTV pattern. The file static at the root of the project is used for the javascript, the css and all the images of the websites. The views are located in the folder views, and the models are the folder model.

B)Second Part/ New project

During these last 4 months I have been working on the Greenbureau project, doing scripts to collect the bills from websites. Furthermore, I also have been working on the company website to fix some problems, and make some improvements. This part of the job gave me the opportunity to have a global overview of the project, excluding the server part. The team will start a new project, to make a smart phone version of Greenbureau.

As my mentor you have been following my progress on this project. First, you made me create and fix scripts; which is a task requiring research and finding optimal solutions in making the scripts generic and as fast as possible. Then you made me work on the website, which requires a better understanding of the project, from the databases, through the servers to the html codes. During these 4 months you have been validating every issue ticket that has been assigned to me. During our mid-internship interview you told me to find a project I will be interested in. I think I could be a major asset for Greenbureau by working on the android project. My knowledge of the website will help to build an ergonomic navigation. I have a good understanding of how to make requests to the server; this skill will also be useful. Finally, before joining the Greenbureau team, I did some personal android projects. My background in android programming makes me the most qualified person to take care of the project allowing Greenbureau to save time and money.
I hope you will take my request into consideration and share my point of view.