Donut on Demand

Test Plan for Made to Order Doughnuts Online Shop Application

Titouan Le Mao
Computer Science
Asian Institute of Technology
st122014@ait.asia

Cedric Le Mercier
Computer Science
Asian Institute of Technology
st122004@ait.asia

Version 1.0

Table of Contents

1. Objectives

Basic tests have been done in JMeter for Database Performance, as well as more user interface testing using **Selenium**. **Selenium** is a collection of tools & libraries for automation of web browsers, useful for Web UI Testing. In this project, we focused on **Selenium Webdriver, the** API to control the behavior of browsers. The language used was Python while the testing framework is Python's integrated **unittest.**

Within selenium, because our project does not incorporate advanced e-commerce features, we are focusing on security and usability aspects, including:

- Browsing on public pages
- Trying accessing authorised content
- Trying accessing restricted content
- Trying admin actions, such as creating recipes or employees

2. Test Strategy

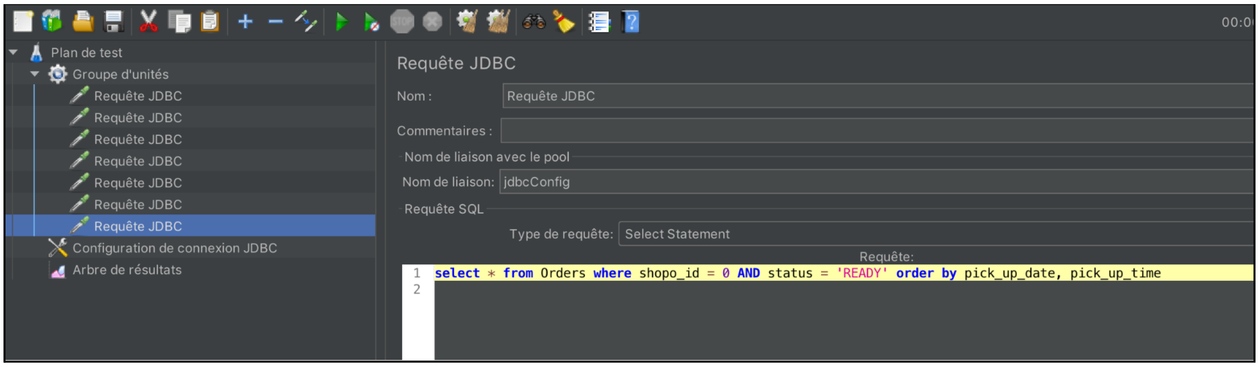
2.1. Database Performance Testing with JMeter

Perform our postgreSQL database performance testing. We used Apache JMeter, a free software that allows you to perform performance tests of applications and servers according to different protocols as well as functional tests. So, once we managed to connect JMeter to our database with the correct URL, we could test its performance.

We created a Thread group to simulate several users wanting to send requests to our database at the same time and thus verify that its performance remains correct.

Then to test that, we implemented JDBC queries for each query that our application uses to measure their performance.

As you can see below, it is an example of queries that our application uses when it accesses our database.



Requête JDBC

Nom : Requête JDBC

Commentaires :

Nom de liaison avec le pool

Nom de liaison: jdbcConfig

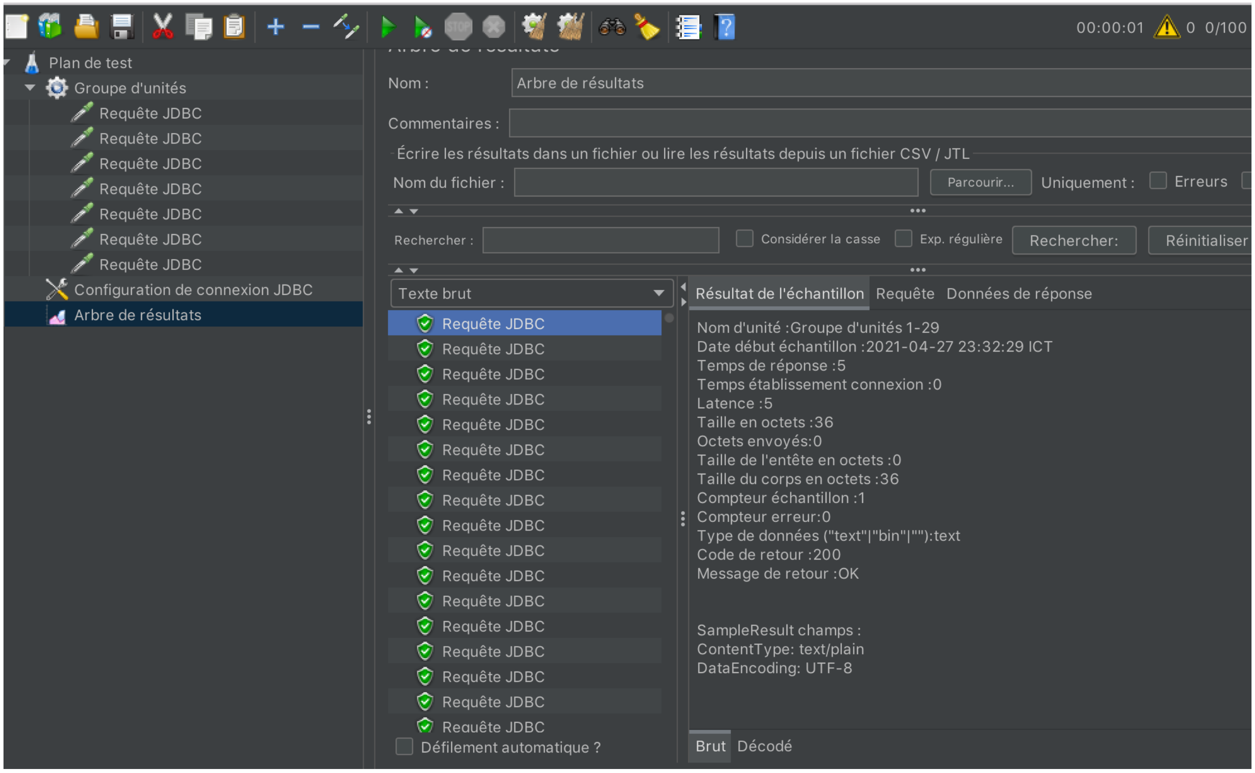
Requête SQL

Type de requête: Select Statement

Requête:

```
select * from Orders where shopo_id = 0 AND status = 'READY' order by pick_up_date, pick_up_time
```

We have tested our database with different parameters and we have concluded after different tests that it is very efficient for our use. Indeed, as the queries we make are not too complex, our database can handle at the same time the queries of 100 users, which is in our case very efficient. Beyond 100 users at the same time, the database needs an additional delay.



Nom : Arbre de résultats

```
Nom d'unité :Groupe d'unités 1-29
Date début échantillon :2021-04-27 23:32:29 ICT
Temps de réponse :5
Temps établissement connexion :0
Latence :5
Taille en octets :36
Octets envoyés:0
Taille de l'entête en octets :0
Taille du corps en octets :36
Compteur échantillon :1
Compteur erreur:0
Type de données ("text"|"bin"|""):text
Code de retour :200
Message de retour :OK


SampleResult champs :
ContentType: text/plain
DataEncoding: UTF-8
```

2.2. User Interface Testing with Selenium

2.2.1. Customer testing

Test Objective:	Ensure a customer, or non-logged in user can access all public pages as wanted
Technique:	● The MainPage class has methods to access public pages, including a method to run all other methods ● Checking if the page title matches what is expected ● If the page is not accessible, the app will either show a 404 error or login page, which marks the test as

	unsuccessful
Completion Criteria	● All tests passed
Special Considerations	● In the future, add all public urls in a sheet or array and loop through it

Test Objective:	Checking if a customer can add a product to the cart
Technique:	● Calling webdriver to browse to the Doughnut list page ● Checking if the page title matches with the correct page ● Calling webdriver to click on the first “Add Now” button ● This action will lead to the shop page
Completion Criteria	● If the webdriver can find the button to remove product from cart, it means adding was successful
Special Considerations	● Once we implement sessions, we will need a more advanced way to check. What if the user already has items in the cart?

Test Objective:	Checking if a customer / non-logged in user can access restricted pages reserved for admins and employees
Technique:	● Browsing to various restricted, e.g /homeEmployee, /createEmployee or /manageOrdersPicking ● If the title does not match the url, assert true
Completion Criteria	● All tests successful
Special Considerations	● In the future, add all public urls in an sheet or array and loop through it

2.2.2. Manager Testing

Test Objective:	Testing if the manager can log in
Technique:	● Browse to login page ● Enter administrator (or manager) details and click submit button ● This will redirects to Employee Homepage, if page title matches, the test is successful
Completion Criteria	● Title matches what is expected
Special Considerations	● None

Test Objective:	Being able to create recipes as a manager
Technique:	● Run the same test as above and login as administrator ● Browse to shop page, and click on “Add Recipe” button found by ID ● If adding a product is successful, this will redirect to adding recipe pages ● Get the number of table rows (<td>) displayed on recipe list, as each rows equals a recipe ● Enter recipe details on each form field respectively and click on submit ● Get the new table row number, and if higher than the previous row number, test is successful
Completion Criteria	● Test is successful if the number of table rows change
Special Considerations	● In the future, implement a check if the inputs are displayed correctly on the last row (newest recipe)

2.2.3. Order Supervisor Testing

Test Objective:	Accessing authorised pages as order supervisor
Technique:	● Same as administrator, but login as supervisor ● Browse to manage orders urls such as /manageOrdersPicking ● If the title matches what is expected, test is successful. A 404 page would fail the test
Completion Criteria	● Accessing /manageOrdersPicking successful
Special Considerations	● None

Test Objective:	Accessing restricted pages reserved for administrator as order supervisor
Technique:	● Login as supervisor ● Browse to admin urls such as /createEmployee ● This should return a 404 page, if the page title matches the URL we are trying to access, test is unsuccessful
Completion Criteria	● Title does not match what is expected
Special Considerations	● None

2.2.4. Cashier Testing

Testing for the cashier will be straightforward, as the only reserved URL for cashier is /manageOrdersWithdrawn