

Marie Espinosa

Projets tests unitaires/e2e/performance

Projets tests

Introduction

L'objectif est de valider le bon fonctionnement d'une application de gestion des utilisateurs en réalisant différents types de tests et en produisant un rapport détaillé des résultats obtenus. Nous verrons les tests unitaires, les tests e2e et enfin des tests de performances.

Tests et résultats

1. Tests unitaires

Changements et Setup

Résultats

2. Test e2e

Configuration et changements

Résultats Sur Firefox

Tests en local utilisant Mocha

Résultats en local utilisant Mocha

3. Tests de non régressions

Configuration et Changements

Résultats test unitaires PHP

Résultats test e2e

4. Tests de Performance avec JMeter ou k6

Configuration

Résultats

Introduction

L'objectif est de valider le bon fonctionnement d'une application de gestion des utilisateurs en réalisant différents types de tests et en produisant un rapport détaillé des résultats obtenus. Nous verrons les tests unitaires, les tests e2e et enfin des tests de performances.

Tests et résultats

1. Tests unitaires

Changements et Setup

- J'ai utilisé un outil de développement en local nommé "Laragon" avec PHP 8.1
- J'ai utilisé le namespace dans le fichier fourni en fonction de mon fichier composer généré.
- J'ai activé Mysql depuis le tableau de bord de Laragon
- Par défaut, Laragon ne met pas de mot de passe pour l'utilisateur root de son système Mysql, j'ai donc changé le mot de passe dans les fichiers tests.
- Je me suis connectée à Mysql avec "mysql -u root" et j'ai copié collé le fichier database.sql (fournit dans le projet de base)

Résultats

```
PHPUnit 9.6.22 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.1.10
Configuration: C:\laragon\www\efrei\tests-rendu\phpunit.xml

User Manager (Mjees\TestsRendu\Model\UserManager)
✓ Add user      28 ms
✓ Add user email exception  39 ms
✓ Update user   33 ms
✓ Remove user   31 ms
✓ Get users     32 ms
✓ Invalid update throws exception  29 ms
✓ Invalid delete throws exception  64 ms

Time: 00:00.258, Memory: 6.00 MB

OK (7 tests, 10 assertions)
PS C:\laragon\www\efrei\tests-rendu>
```

Add user	SUCCÈS
Add user email exception	SUCCÈS
Update user	SUCCÈS
Remove user	SUCCÈS
Get users	SUCCÈS

Invalid update throws exception	SUCCÈS
Invalid delete throws exception	SUCCÈS

2. Test e2e

Configuration et changements

- J'ai dû lancer mon serveur Laragon
- J'ai dû dans api.php ajouter

```
require __DIR__ . '/../../vendor/autoload.php';
```

- Le plugin Selenium était déjà installé sur mon navigateur Firefox

Pour lancer les test :

- npm i
- lancer le serveur et MySQL
- si besoin : changer le lien du serveur dans les fichiers tests/e2e

```
await driver.get("http://localhost/efrei/tests-rendu/src/model/");
```

- npm run test

Résultats Sur Firefox

- Add a user and verify it's displayed

Project: test rendu*

Executing ▾

✓ Untitled

✓ it add a new user*

http://localhost

	Command	Target	Value
1	✓ open	/efrei/tests-rendu/src/model/	
2	✓ set window size	725x672	
3	✓ click	id=name	
4	✓ type	id=name	marie
5	✓ click	id=email	
6	✓ type	id=email	espinosa@gmail.com
7	✓ click	css=button	
8	✓ assert element present	css=li	

Command

- Rename user

Project: test rendu*

Tests +

Search tests... Q

http://localhost/efrei/tests-rendu/src/model/

	Command	Target	Value
3	✓ click	css=button:nth-child(1)	
4	✓ click	id=name	
5	✓ click	id=name	
6	✓ double click	id=name	
7	✓ click	id=name	
8	✓ type	id=name	lucie
9	✓ click	css=button:nth-child(4)	
10	✓ assert element present	css=li	
11	✓ close		

Command: assert element present // [icon]

Target: css=li [icon] [icon]

Value: [input]

Description: [input]

3- Delete user

On peut voir avec le “assert element not present” que la ligne de l'utilisateur a été supprimé

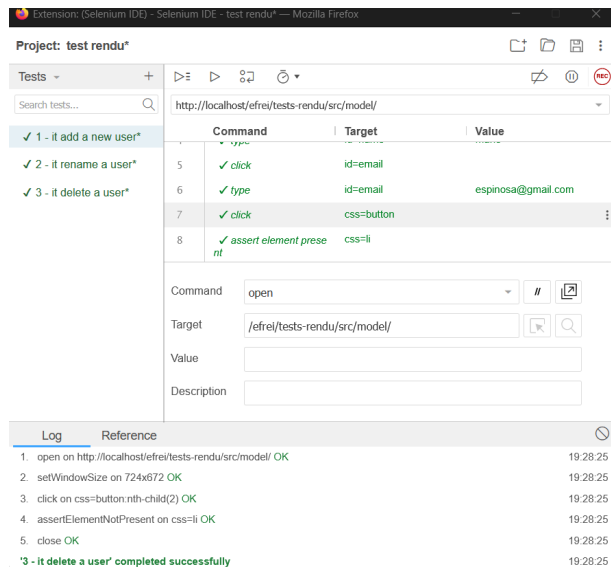
Project: test rendu*

Executing ▾

http://localhost/efrei/tests-rendu/src/model/

	Command	Target	Value
1	✓ open	http://localhost/efrei/test s-rendu/src/model/	
2	✓ set window size	724x672	
3	✓ click	css=button:nth-child(2)	
4	✓ assert element not present	css=li	
5	✓ close		

4- Vue d'ensemble



Tests en local utilisant Mocha

J'ai exporté les tests dans le projet, installé mocha et j'ai dû faire quelques changements car j'ai eu des soucis pour que le driver de selenium affiche bien la page. J'ai du changer aussi le deuxième test e2e (renommage utilisateur) car le test s'effectuait avant que la liste des utilisateurs s'affichait.

```
await driver.wait(until.elementsLocated(By.css("li")), 5000);
```

Résultats en local utilisant Mocha

```

> tests-rendu@1.0.0 test
> mocha tests/e2e/

1 - it add a new user
  ✓ 1 - it add a new user (1246ms)

2 - it rename a user
  ✓ 2 - it rename a user (1847ms)

3 - it delete a user
  ✓ 3 - it delete a user (572ms)

3 passing (17s)
  
```

3. Tests de non régressions

Configuration et Changements

- J'ai drop la base de donnée

```
mysql> DROP DATABASE user_management;  
Query OK, 1 row affected (0.03 sec)
```

- J'ai changé le fichier SQL pour ajouter la colonne created_at, puis j'ai lancé le script SQL.

```
mysql> DESCRIBE users;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default          | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| id         | int           | NO   | PRI | NULL             | auto_increment |  
| name       | varchar(100)  | NO   |     | NULL             |                |  
| email      | varchar(150)  | NO   | UNI | NULL             |                |  
| created_at | datetime      | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.02 sec)
```

- J'ai changé le code de la fonction AddUser pour permettre d'ajouter une date si elle est passée en argument
- J'ai modifié le fichier HTML pour ajouter un champ date
- J'ai modifié le fichier js pour afficher la date et récupérer la valeur de la date dans le formulaire de création d'utilisateur

Résultats test unitaires PHP

```
PHPUnit 9.6.22 by Sebastian Bergmann and contributors.  
  
Runtime:       PHP 8.1.10  
Configuration: C:\laragon\www\efrei\tests-rendu\phpunit.xml  
  
User Manager (Mjees\TestsRendu\Model\UserManager)  
✓ Add user 1127 ms  
✓ Add user email exception 156 ms  
✓ Update user 50 ms  
✓ Remove user 45 ms  
✓ Get users 46 ms  
✓ Invalid update throws exception 42 ms  
✓ Invalid delete throws exception 117 ms  
  
Time: 00:01.861, Memory: 6.00 MB  
  
OK (7 tests, 10 assertions)  
PS C:\laragon\www\efrei\tests-rendu>
```

Conclusion : Le changement n'a apporté aucune régressions sur le backend

Résultats test e2e

```
1 - it add a new user
  ✓ 1 - it add a new user (1170ms)

2 - it rename a user
  1) 2 - it rename a user

3 - it delete a user
  ✓ 3 - it delete a user (564ms)

2 passing (17s)
1 failing

1) 2 - it rename a user
    2 - it rename a user:
      NoSuchElementError: Unable to locate element: button:nth-child(4)
      at Object.throwDecodedError (node_modules\selenium-webdriver\lib\error.js:523:15)
      at parseHttpResponse (node_modules\selenium-webdriver\lib\http.js:524:13)
```

J'ai du modifier cette ligne

```
await driver.findElement(By.css("button:nth-child(5)")).click()
```

Puis les résultats ont changé :

```
> tests-rendu@1.0.0 test
> mocha tests/e2e/

1 - it add a new user
  ✓ 1 - it add a new user (1452ms)

2 - it rename a user
  ✓ 2 - it rename a user (1839ms)

3 - it delete a user
  ✓ 3 - it delete a user (645ms)

3 passing (22s)
```

4. Tests de Performance avec JMeter ou k6

Configuration

Thread Properties	
Number of Threads (users):	100
Ramp-up period (seconds):	1
Loop Count:	<input type="checkbox"/> Infinite 100
<input checked="" type="checkbox"/> Same user on each iteration	
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Specify Thread lifetime	

HTTP Request

Name:

HTTP Request

Comments:

▲ ▼

Basic

Advanced

Web Server

Protocol [http]:

http

 Server Name or IP:

localhost

 Port Number:

HTTP Request

GET

▼

 Path:

efrei/tests-rendu/src/model/

 Content encoding:

☐ Redirect Automatically

☒ Follow Redirects

☒ Use KeepAlive

☐ Use multipart/form-data

☐ Browser-compatible headers

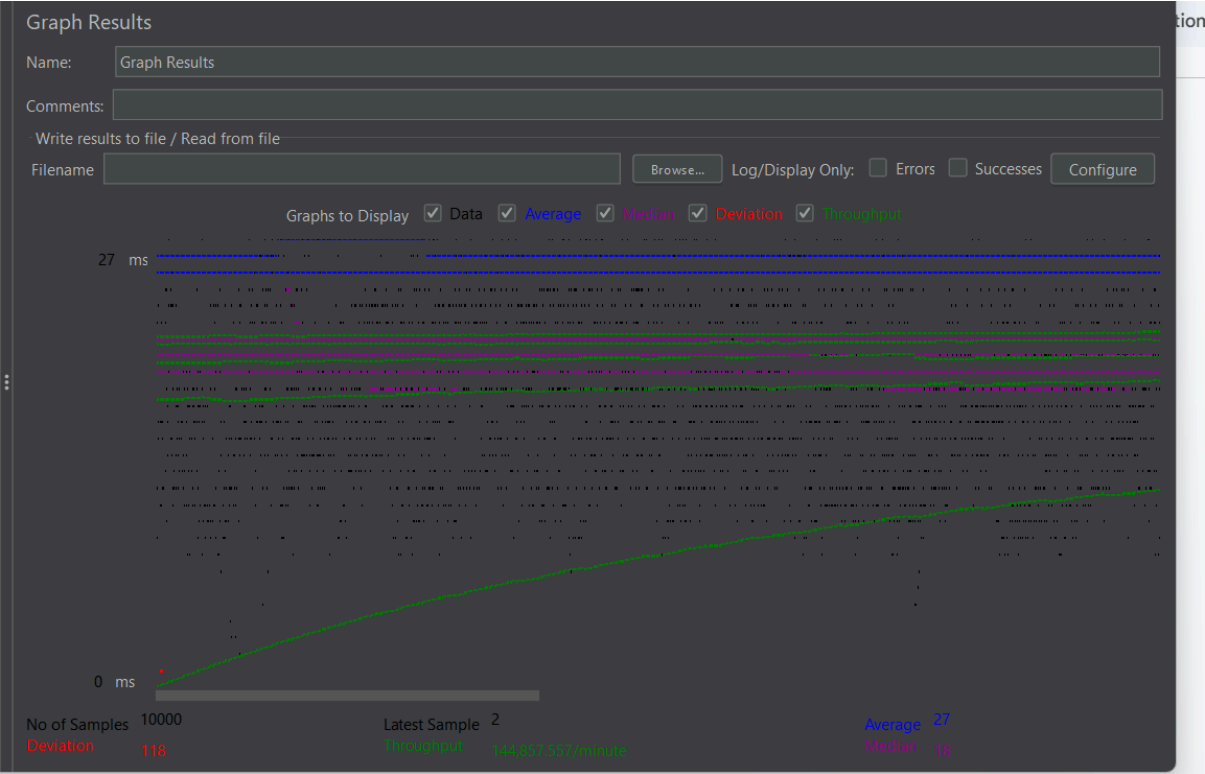
Parameters

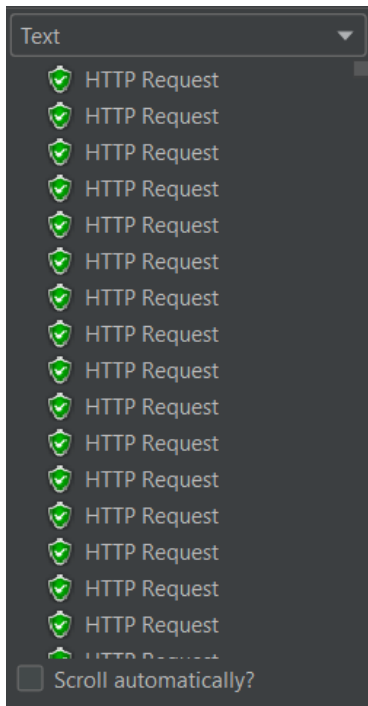
Body Data

Files Upload

Send Parameters With the Request

Résultats





Filename	<div>Browse... Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="checkbox"/> Configure</div>									
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
HTTP Reque...	110000	97	0	17280	917.64	5.60%	194.6/sec	215.41	35.69	1133.6
TOTAL	110000	97	0	17280	917.64	5.60%	194.6/sec	215.41	35.69	1133.6

Le test de performance a permis de simuler 110 000 requêtes HTTP vers l'API. Le temps de réponse moyen observé est de 97 millisecondes. Le débit atteint environ 194 requêtes par seconde. Toutefois, un taux d'erreurs de 5,6 % a été relevé, avec une latence maximale de 17 secondes, ce qui indique des pics de saturation. Le serveur reste performant dans l'ensemble, mais une optimisation du traitement des requêtes pour renforcer sa stabilité serait mieux. Si le système n'est pas préparé à filtrer ou limiter ces requêtes massives, cela peut entraîner une interruption de service. Il montre que sous une pression extrême, l'application pourrait devenir instable.

Réalisé par Marie Espinosa le 06/04/2024