

# Projet Test Unitaire

*Par S.Mahboubi  
F.Hammouch  
J. Takoutsing  
P.Zozor*

Master 1 Data Engineer/Science

09/12/21



## Sommaire :

- 1- Présentation du sujet
- 2- Arborescence de la WebApp
- 3- Mise en place des tests
- 4- Coverage & Locust
- 5- Demo





## Présentation du sujet

Nous travaillons au sein d'une DSI d'entreprise qui propose une plateforme numérique pour réserver des places aux compétitions deadlifting, strongman en Amérique du Nord et en Australie.

En arrivant au bureau ce matin nous avons reçu un mail de notre responsable:

- Il y a plusieurs bugs qui font planter la WebApp de l'entreprise
- Il faut debuguer la Webapp en utilisant la méthode de Test driven development

## Welcome to the GUDLFT Registration Portal!

Please enter your secretary email to continue:

Email:



# Arborescence de la WebApp

## ▼ WEBAPP\_TESTING-MASTER

> \_\_pycache\_\_

> .pytest\_cache

### ▼ .vscode

{ } settings.json

### ▼ templates

<> booking.html

<> index.html

<> score.html

<> welcome.html

### ▼ tests

> \_\_pycache\_\_

> performance\_tests

+ test\_server.py

≡ .coverage

{ } clubs.json

{ } competitions.json

≡ requirements.txt

+ server.py

<> test\_report.html



# Server.py

```
1 import json
2 from flask import Flask, render_template, request, redirect, flash, url_for
3
4
5 def loadClubs():
6     with open('clubs.json') as c:
7         listofClubs = json.load(c)['clubs']
8         return listofClubs
9
10
11 def loadCompetitions():
12     with open('competitions.json') as comps:
13         listofCompetitions = json.load(comps)['competitions']
14         return listofCompetitions
15
16
17 app = Flask(__name__)
18 app.secret_key = 'something_special'
19
20 competitions = loadCompetitions()
21 clubs = loadClubs()
22
23 @app.route('/')
24 def index():
25     return render_template('index.html')
26
27 @app.route('/showSummary', methods=['POST'])
28 def showSummary():
29     club = [club for club in clubs if club['email'] == request.form['email']][0]
30     return render_template('welcome.html', club=club, competitions=competitions)
31
```

```
32
33 @app.route('/book/<competition>/<club>')
34 def book(competition, club):
35     foundClub = [c for c in clubs if c['name'] == club][0]
36     foundCompetition = [c for c in competitions if c['name'] == competition][0]
37     if foundClub and foundCompetition:
38         return render_template('booking.html', club=foundClub, competition=foundCompetition)
39     else:
40         flash("Something went wrong-please try again")
41         return render_template('welcome.html', club=club, competitions=competitions)
42
43
44 @app.route('/purchasePlaces', methods=['POST'])
45 def purchasePlaces():
46     competition = [c for c in competitions if c['name'] == request.form['competition']][0]
47     club = [c for c in clubs if c['name'] == request.form['club']][0]
48     placesRequired = int(request.form['places'])
49     competition['numberOfPlaces'] = int(competition['numberOfPlaces']) - placesRequired
50     flash('Great-booking complete!')
51     return render_template('welcome.html', club=club, competitions=competitions)
52
53 # TODO: Add route for points display
54 @app.route('/logout')
55 def logout():
56     return redirect(url_for('index'))

```

PROBLÈMES

CONSOLE DE DÉBOGAGE

TERMINAL

```
(Unit_test_Webapp) → Webapp_testing-master python3 server.py
(Unit_test_Webapp) → Webapp_testing-master
```



## Route Index

- Afin de lancer la Webapp il est nécessaire d'ajouter au code server.py la condition:

```
60
61  if __name__ == "__main__":
62      app.run(debug=True)
```

- Test pour vérifier que le serveur se lance correctement:

```
12  # pytest --disable-warnings -v ./tests/test_server.py::test_root_index
    Run | Debug | Show in Test Explorer
13  def test_root_index(client):
14      response = client.get('/')
15      assert response.status_code == 200
16      assert b'<form' in response.data
17      assert b'<input' in response.data
```

### Welcome to the GUDLFT Registration Portal!

Please enter your secretary email to continue:

Email:

## Route showSummary

- Test pour s'assurer que *summary* s'affiche correctement avec un *secretary* email

```
24 # pytest --disable-warnings -v ./tests/test_server.py::test_InSummary
Run | Debug | Show in Test Explorer
25 def test_InSummary(client):
26     response = client.post("/showSummary", data = dict(email = "john@simplylift.co"), follow_redirects=True)
27     assert response.status_code == 200
28     data = response.data.decode()
29     print(data)
30     assert "Welcome, john@simplylift.co" in data
```

**Welcome, john@simplylift.co**

[Logout](#) Points available: 13

### Competitions:

- Spring Festival  
Date: 2020-03-27 10:00:00  
Number of Places: 25 [Book Places](#)
- Fall Classic  
Date: 2020-10-22 13:30:00  
Number of Places: 13 [Book Places](#)

## Welcome to the GUDLFT Registration Portal!

Please enter your secretary email to continue:

Email:

# Route showSummary

- Test pour s'assurer que *summary* ne s'affiche pas si le mail n'est pas référencé

```
33 #pytest --disable-warnings -v ./tests/test_server.py::test_noInSummary
Run | Debug | Show in Test Explorer
34 def test_noInSummary(client):
35     response = client.post("/showSummary", data = dict(email = "john@exemple.com"), follow_redirects=True)
36     assert response.status_code == 403
37     data = response.data.decode()
38     print(data)
39     assert "Désolé, cet email n'a pas été trouvé" in data
```

## Welcome to the GUDLFT Registration Portal!

Please enter your secretary email to continue:

Email:

→ Modification du script de la Webapp et du fichier index.html

### IndexError

IndexError: list index out of range

Traceback (most recent call last)

```
File "opt\anaconda3\nvs\Unit_Test\Webapp\lib\python3.8\site-packages\flask\app.py", line 2092, in wsgi_app
    return self.wsgi_app(environ, start_response)
File "opt\anaconda3\nvs\Unit_Test\Webapp\lib\python3.8\site-packages\flask\app.py", line 2076, in wsgi_app
    response = self.handle_exception(e)
File "opt\anaconda3\nvs\Unit_Test\Webapp\lib\python3.8\site-packages\flask\app.py", line 2073, in wsgi_app
    response = self.full_dispatch_request()
File "opt\anaconda3\nvs\Unit_Test\Webapp\lib\python3.8\site-packages\flask\app.py", line 1518, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "opt\anaconda3\nvs\Unit_Test\Webapp\lib\python3.8\site-packages\flask\app.py", line 1516, in full_dispatch_request
    rv = self.dispatch_request()
File "opt\anaconda3\nvs\Unit_Test\Webapp\lib\python3.8\site-packages\flask\app.py", line 1502, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**req.view_args)
File "C:\Users\zhilipezov\Documents\Ynor_M1\02_Test\UnitTest\Project_Final\Webapp_Avant_debug\server.py", line 28, in showSummary
    club = [club for club in clubs if club["email"] == request.form["email"]]
```

IndexError: list index out of range

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it by clicking on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

- `dump()` shows all variables in the frame
- `dump(obj)` dumps all that's known about the object

```
@app.route('/showSummary', methods=['POST'])
def showSummary():
    club = [club for club in clubs if club['email'] == request.form['email']]
    if club:
        club_email = club[0]
        return render_template('welcome.html', club=club_email, competitions=competitions)
    else:
        return render_template('index.html', error = "Désolé, cet email n'a pas été trouvé"), 403
```

```
15 <button type="submit"><a href="{{url_for('
16 {%if error%}
17 <p class=error>{{error}}</p>
18 {%endif%}
```

## Welcome to the GUDLFT Registration Portal!

Please enter your secretary email to continue:

Email:

Désolé, cet email n'a pas été trouvé





## Route booking / compétition / club

- Test pour s'assurer de l'affichage d'une page **booking.html** pour un club et une compétition valide

```
42 #pytest --disable-warnings -v ./tests/test_server.py::test_Booking
Run | Debug | Show in Test Explorer
43 def test_Booking(client):
44     response = client.post("/showSummary", data = dict(email = "john@simplylift.co"), follow_redirects=True)
45     assert response.status_code == 200
46     data = response.data.decode()
47     print(data)
48     assert "/book/Spring%20Festival/Simply%20Lift" in data
```

### Spring Festival

Places available: 25

How many places?



Book

# Route showSummary

- Test pour s'assurer que *l'affichage utilisateur empêche la réservation d'une compétition ou d'un club invalide* :

```
51 #pytest --disable-warnings -v ./tests/test_server.py::test_invalideBooking
Run | Debug | Show in Test Explorer
52 def test_invalideBooking(client):
53     response = client.get("/book/Winter%20Festival/Simply%20Lift", follow_redirects = True)
54     assert response.status_code == 200
55     data = response.data.decode()
56     print(data)
57     assert "Welcome to the GUDLFT Registration Portal!" in data
```

→ Modification du script de la Webapp

## IndexError

IndexError: list index out of range

Traceback (most recent call last)

```
File "/opt/anaconda3/envs/Unit_test_Webapp/lib/python3.8/site-packages/flask/app.py", line 2092, in __call__
    return self.wsgi_app(environ, start_response)
File "/opt/anaconda3/envs/Unit_test_Webapp/lib/python3.8/site-packages/flask/app.py", line 2076, in wsgi_app
    response = self.handle_exception(e)
File "/opt/anaconda3/envs/Unit_test_Webapp/lib/python3.8/site-packages/flask/app.py", line 2073, in wsgi_app
    response = self.full_dispatch_request()
File "/opt/anaconda3/envs/Unit_test_Webapp/lib/python3.8/site-packages/flask/app.py", line 1518, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/opt/anaconda3/envs/Unit_test_Webapp/lib/python3.8/site-packages/flask/app.py", line 1516, in full_dispatch_request
    rv = self.dispatch_request()
File "/opt/anaconda3/envs/Unit_test_Webapp/lib/python3.8/site-packages/flask/app.py", line 1502, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**req.view_args)
File "/Users/philippeozor/Documents/Trav_M1/02_Test Unitaire/Projet_Final/Webapp_Avant debug/server.py", line 36, in book
    foundCompetition = [c for c in competitions if c['name'] == competition][0]
```

IndexError: list index out of range

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

- `dump()` shows all variables in the frame
- `dump(obj)` dumps all that's known about the object

```
@app.route('/book/<competition>/<club>')
def book(competition,club):
    try:
        foundClub = [c for c in clubs if c['name'] == club][0]
        foundCompetition = [c for c in competitions if c['name'] == competition][0]
        if foundClub and foundCompetition:
            return render_template('booking.html',club=foundClub,competition=foundCompetition)
    except:
        #flash("Something went wrong-please try again")
        return redirect(url_for('index'))
```

## Welcome to the GUDLFT Registration Portal!

Please enter your secretary email to continue:

Email:



Enter



## Route booking / compétition / club

- Test pour s'assurer de l'affichage d'un message d'erreur lorsque la **date est dépassée**

```
68 #pytest --disable-warnings -v ./tests/test_server.py::test_invalid_purchase_point_limite_date
Run | Debug | Show in Test Explorer
69 def test_invalid_purchase_point_limite_date(client):
70     rv_test=client.get('/book/Fall%20Classic/Iron%20Temple')
71     assert rv_test.status_code==403
72     data =rv_test.data.decode()
73     print(data)
74     assert 'Date limite d&#39;achat dépassé, cliquez ici pour revenir en arrière :' in data
```

### Fall Classic

Places available: 13

How many places?

Book

Il est possible de réserver une place pour un événement dépassé → Modification du script de la Webapp

```
@app.route('/book/<competition>/<club>')
def book(competition,club):
    try:
        foundClub = [c for c in clubs if c['name'] == club][0]
        foundCompetition = [c for c in competitions if c['name'] == competition][0]

        if foundClub and foundCompetition:
            if foundCompetition['date'] > todayData:
                return render_template('booking.html',club=foundClub,competition=foundCompetition)
            else:
                return render_template('booking.html',club=foundClub,competition=foundCompetition, error="Date limite d'achat dépassé, cliquez ici pour revenir en arrière :"), 403
    except :
        return redirect(url_for('index'))
```

Date limite d'achat dépassé, cliquez ici pour revenir en arrière : [Retour](#)



## Route PurchasePlaces (1/3)

- Test pour s'assurer que *lorsque l'utilisateur achète, les places sont correctement déduites*

```
59 #pytest --disable-warnings -v ./tests/test_server.py::test_valid_purchase_point
Run | Debug | Show in Test Explorer
60 def test_valid_purchase_point(client):
61     rv_test=client.post('/purchasePlaces', data=dict(club = 'Iron Temple', competition='Spring Festival', places = 1), follow_redirects = True)
62     data =rv_test.data.decode()
63     print(data)
64     assert 'Number of Places: 24' in data
65     assert 'Great-booking complete!' in data
66     assert 'Points available: 3' in data
```

Les “points available” ne sont pas correctement déduits → Modification du script de la Webapp

**Welcome, admin@irontemple.com**

[Logout](#)

- Great-booking complete!

Points available: 4

### Competitions:

- Spring Festival  
Date: 2020-03-27 10:00:00  
Number of Places: 24 [Book Places](#)
- Fall Classic  
Date: 2020-10-22 13:30:00  
Number of Places: 13 [Book Places](#)



## Route PurchasePlaces (2/3)

- Test pour s'assurer de l'affichage d'un **message d'erreur lorsque plus de 12 places** sont achetés

```
76 #pytest --disable-warnings -v ./tests/test_server.py::test_invalid_purchase_point_limite_date
Run | Debug | Show in Test Explorer
77 def test_invalid_purchase_point_more_than_12(client):
78     rv_test=client.post('/purchasePlaces', data=dict(club='Iron Temple', competition='Spring Festival', places=13), follow_redirects=True)
79     data=rv_test.data.decode()
80     print(data)
81     assert 'Impossible to buy more than 12 places and less than 1' in data
82
```

### Spring Festival

Places available: 25

How many places? 13



Book

Il est possible de commander plus de 12 places → Modification du script de la Webapp

### Welcome, john@simplylift.co

[Logout](#)

- Great-booking complete!

Points available: 13

#### Competitions:

- Spring Festival  
Date: 2020-03-27 10:00:00  
Number of Places: 12 [Book Places](#)
- Fall Classic  
Date: 2020-10-22 13:30:00  
Number of Places: 13 [Book Places](#)



## Route PurchasePlaces (3/3)

Modification de la Webapp et du welcome.html

```
@app.route('/purchasePlaces',methods=['POST'])
def purchasePlaces():
    competition = [c for c in competitions if c['name'] == request.form['competition']][0]
    club = [c for c in clubs if c['name'] == request.form['club']][0]
    placesRequired = int(request.form['places'])
    pts = int(club["points"])

    if placesRequired <= 12 and placesRequired>0:
        if placesRequired <= int(competition['numberOfPlaces']) and pts*3 >= placesRequired and pts-placesRequired*3>=0 :
            competition['numberOfPlaces'] = int(competition['numberOfPlaces'])-placesRequired
            club["points"] = pts - placesRequired*3
            pts = pts - placesRequired*3
            flash('Great-booking complete!')
            return render_template('welcome.html', club=club, competitions=competitions)
        else:
            return render_template('welcome.html', club=club, competitions=competitions, errorPurchase = "Not enoough point for buy"), 403
    else:
        return render_template('welcome.html', club=club, competitions=competitions, errorPurchase = "Impossible to buy more than 12 places and less than 1"), 403
```

Welcome, john@simplylift.co

[Logout](#) Points available: 13

Impossible to buy more than 12 places and less than 1

### Competitions:

- Impossible to buy more than 12 places and less than 1

Spring Festival  
Date: 2022-03-27 10:00:00  
Number of Places: 25 [Book Places](#)

- Impossible to buy more than 12 places and less than 1

Fall Classic  
Date: 2020-10-22 13:30:00  
Number of Places: 13 [Book Places](#)

Welcome, admin@irontemple.com

[Logout](#)

- Great-booking complete!

Points available: 1

### Competitions:

- Spring Festival  
Date: 2022-03-27 10:00:00  
Number of Places: 24 [Book Places](#)
- Fall Classic  
Date: 2020-10-22 13:30:00  
Number of Places: 13 [Book Places](#)

Les 2 fonctionnalités  
fonctionnent désormais !



## Test Logout

```
84
85  #Test logout
    Run | Debug | Show in Test Explorer
86  def test_logout_user(client):
87      rv = client.get("/logout", follow_redirects=True)
88      assert rv.status_code == 200
89      data = rv.data.decode()
90      print(data)
91      assert "Welcome to the GUDLFT Registration Portal!" in data
92
```

**Welcome to the GUDLFT Registration Portal!**

Please enter your secretary email to continue:

Email:

[Scoreboard](#)





# Test d'intégration

```
93 # Test integration
94 def test_integration(client):
95     clubs = loadClubs()
96     competitions = loadCompetitions()
97     response = client.get('/')
98     assert response.status_code == 200
99     rv_showSumm = client.post('/showSummary', data={"email": "john@simplylift.co"}, follow_redirects=True)
100     assert rv_showSumm.status_code == 200
101     data = rv_showSumm.data.decode()
102     print(data)
103     assert competitions[0]['name'] in data
104     assert clubs[0]['email'] in data
105
106     rv_book = client.get('/book/Spring%20Festival/She%20Lifts', follow_redirects=True)
107     assert rv_book.status_code == 200
108
109     rv_purchasePlaces = client.post('/purchasePlaces', data=dict(club = 'Simply Lift', competition='Spring Festival', places = 2), follow_redirects = True)
110     assert rv_purchasePlaces.status_code == 200
111     data = rv_purchasePlaces.data.decode()
112     assert 'Number of Places: 22' in data
113     assert 'Points available: 7' in data
114     assert 'Great-booking complete!' in data
115
116     rv_logout = client.get("/logout", follow_redirects=True)
117     assert rv_logout.status_code == 200
118     data = rv_logout.data.decode()
119     print(data)
120     assert "Welcome to the GUDLFT Registration Portal!" in data
```





## Tableau des scores

- Afficher un tableau des scores sans avoir à se connecter

### Welcome to the GUDLFT Registration Portal!

Please enter your secretary email to continue:

Email:

[Scoreboard](#)

### Welcome to the Score and Points Portal!

#### Listes des clubs:

- Name : Simply Lift  
email : john@simplylift.co  
points : 13
- Name : Iron Temple  
email : admin@irontemple.com  
points : 1
- Name : She Lifts  
email : kate@shelifts.co.uk  
points : 12

[logout](#)



## Coverage

```
coverage run -m pytest  
coverage html
```

### Coverage report: 99%

<i>Module</i>	<i>statements</i>	<i>missing</i>	<i>excluded</i>	<i>coverage</i>
server.py	63	1	0	98%
tests\__init__.py	0	0	0	100%
tests\test_server.py	97	0	0	100%
<b>Total</b>	<b>160</b>	<b>1</b>	<b>0</b>	<b>99%</b>

*coverage.py v6.1.2, created at 2021-12-09 10:07 +0100*





# Locust

```
locust -f ./tests/performance_tests/locust.py --host http://127.0.0.1:5000 --users 500 --spawn-rate 500
```

```
Regarder l'onglet Charts de http://localhost:8089
```

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	254	254	2100	3100	2470	2022	3144	0	26.57	26.57
GET	/book/Spring%20Festival/Simply%20Lift	255	255	2100	3100	2515	2023	3130	0	30.29	30.29
GET	/logout	275	275	2100	3100	2505	2025	3138	0	30.43	30.43
POST	/showSummary	283	283	2100	3100	2471	2028	3118	0	31	31
Aggregated		1067	1067	2100	3100	2490	2022	3144	0	118.29	118.29