

# Testing Performance LOCUST

...

August 1, 2017

# Overview

In software engineering, performance testing is in general, a testing practice performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.



LOCUST

# Automated Load Testing: Locust

# Why need automation testing tools instead manual testing?

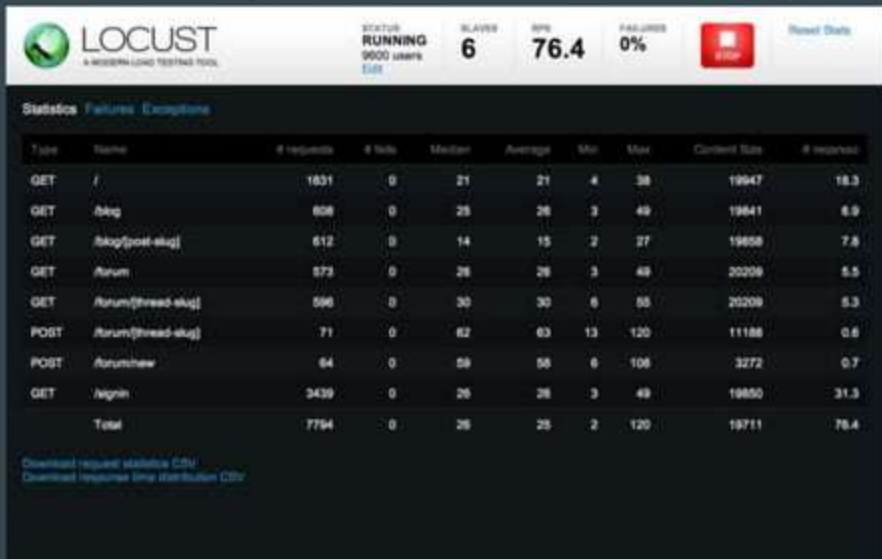
- I'm too lazy to do testing manually (read: Is Boring)
  - I'm too busy to do testing again, again and again (read: Is Repetitive)
  - I'm just ordinary people who have many mistakes (read: Is error prone)
-

# Why need automation testing tools instead manual testing?

BUT,

- I'm smart and I'm a software developer, so I teach the computer how to do the boring, repetitive, tedious stuff
- And then I just watch the computer to do it for me:
  - Again
  - Again
  - Again
  - ○ .....
  - And Again

# LOCUST : An Open Source load testing tool



## Installation

The easiest way to install Locust is from PyPI, using PIP

```
> pip install locustio
```

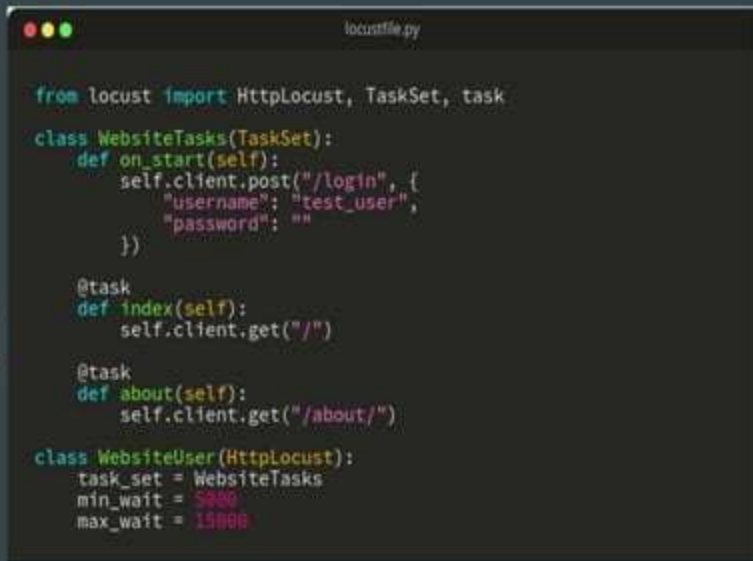


- Read more detailed installations instructions in the [documentation](#)
- Get the source code at [Github](#)

# Example code

A fundamental feature of Locust is that you describe all your test in Python code. No need for clunky UIs or bloated XML, just plain code.

```
$ locust -f locustfile.py
```



```
locustfile.py

from locust import HttpLocust, TaskSet, task

class WebsiteTasks(TaskSet):
    def on_start(self):
        self.client.post("/login", {
            "username": "test_user",
            "password": ""
        })

    @task
    def index(self):
        self.client.get("/")

    @task
    def about(self):
        self.client.get("/about/")

class WebsiteUser(HttpLocust):
    task_set = WebsiteTasks
    min_wait = 5000
    max_wait = 15000
```





LOCUST

**How To used it in Odoo**



# Study Case

- We will testing scenario with some users to create SO and then confirm SO - with xmlrpc
- First testing will use odoo on localhost
- Second testing will use odoo on my own vps server

---

# Locust script

```
from locust import HttpLocust, TaskSet

def index(l):
    l.client.get("/")

class UserBehavior(TaskSet):
    tasks = {index: 2}

    def on_start(self):
        index(self)

class WebsiteUser(HttpLocust):
    task_set = UserBehavior
    min_wait = 5000
    max_wait = 9000
```

## Locust script (Explanation) → need to complete

```
from locust import HttpLocust, TaskSet

def index(l):
    l.client.get("/")

class UserBehavior(TaskSet):
    tasks = {index: 2}

    def on_start(self):
        index(self)

class WebsiteUser(HttpLocust):
    task_set = UserBehavior
    min_wait = 5000
    max_wait = 9000
```

# XML-RPC Code

```
def index(request):
    database = 'blueprint'
    url = 'http://31.220.53.10:8065'

    user = 'admin'
    password = 'wishmeluch'

    common = xmlrpc.client.ServerProxy('{}xmlrpc/2/common'.format(url))
    models = xmlrpc.client.ServerProxy('{}xmlrpc/2/object'.format(url))

    uid = common.authenticate(database, user, password, {})

    # search products
    prod_ids = models.execute_kw(database, uid, password, 'product.product', 'search',
                                 [[{'name', 'ilike', 'ipad'}]])

    # search partners
    partner = models.execute_kw(database, uid, password, 'res.partner', 'search',
                                 [[{'name', 'ilike', 'fletch'}]])

    # create records
    so_id = models.execute_kw(database, uid, password, 'sale.order', 'create', [{
        'partner_id': partner[0],
        'order_line': [(0, 0, {'product_id': prod_ids[0],
                               'product_uom_qty': 1}),
                       (0, 0, {'product_id': prod_ids[1],
                               'product_uom_qty': 2}),
                       ],
    }])
    models.execute_kw(database, uid, password, 'sale.order', 'action_confirm',
                      [so_id])
    return HttpResponse("Hello, world. You're at the polls index.")
```

## XML-RPC Code (Explanation) → need to complete

For now, I create simple django code to call xmlrpc. And then Locust will call url from django.

WHY???

“Actually there is library / plugin to testing odoo with Locust, but when I try, the locust task can working normally, but didn't giving the statistic result ;(”

# Study Case 1 : Using odoo localhost

## Start new Locust swarm

Number of users to simulate

Hatch rate (users spawned/second)

**Start swarming**

# Study Case 1 : Using odoo localhost



**LOCUST**  
A MODERN LOAD TESTING TOOL

STATUS  
**RUNNING**  
10 users  
[Edit](#)

RPS  
**0.6**

FAILURES  
**0%**



[Reset Stats](#)

[Statistics](#) [Failures](#) [Exceptions](#) [Download Data](#)

Type	Name	# requests	# fails	Median	Average	Min	Max	Content Size	# reqs/sec
GET	/polls/	23	0	11000	10157	2045	13253	40	0.6
Total		23	0	11000	10157	2045	13253	40	0.6



# Study Case 2 : Using odoo vps server



LOCUST  
A MODERN LOAD TESTING TOOL

STATUS  
**RUNNING**  
10 users  
[Edit](#)

RPS  
**1**

FAILURES  
**0%**



[Reset Stats](#)

[Statistics](#) [Failures](#) [Exceptions](#) [Download Data](#)

Type	Name	# requests	# fails	Median	Average	Min	Max	Content Size	# reqs/sec
GET	/polls/	21	0	8800	8376	4410	10432	40	1
Total		21	0	8800	8376	4410	10432	40	1

# Notes:

- Limitations

Locust will gather and display stats as it's creating users, but as soon as it reaches your requested number, all the stats are reset and it starts gathering new ones. If you want to see how many users it takes to cause problems on your site, you need to sit and stare at the web interface as it ramps up or that information will be gone.

- Doesn't Get All Resources

Locust won't request a URL unless you explicitly tell it to. A legitimate user with an actual web browser will most likely end up requesting several URLs when loading a page (for images, style sheets, and scripts). Keep this in mind when building your tests. If you want to simulate an actual "page view", you'll need to explicitly list all the URLs for various resources

# Notes:

- Testing Never Stops

That is, you can't tell it how many hits each user should simulate. If you want a specific number, you'll need to just stare at the web interface and stop the test when the total gets close to what you want.

- <http://www.sixfeetup.com/blog/performance-testing-with-locust>
- <https://www.promptworks.com/blog/load-testing-with-locust>
- <https://andrew-jones.com/blog/load-testing-with-locust/>

## Another Presentation:

Odoo Profiler

[https://docs.google.com/presentation/d/1tWEtyQcN5eP6uQ845TrvHQyf7DCtEiciQ-Dw8utwQyY/edit#slide=id.g2f2a887af3\\_0\\_67](https://docs.google.com/presentation/d/1tWEtyQcN5eP6uQ845TrvHQyf7DCtEiciQ-Dw8utwQyY/edit#slide=id.g2f2a887af3_0_67)

Thank You



Angga F. Ikrom

[angga.faizul05@gmail.com](mailto:angga.faizul05@gmail.com)

**Someone who acts as software developer**