DP2 – A+

GATLING EXTRA

Yoana Dimitrova Penkova
yoadimpen@alum.us.es

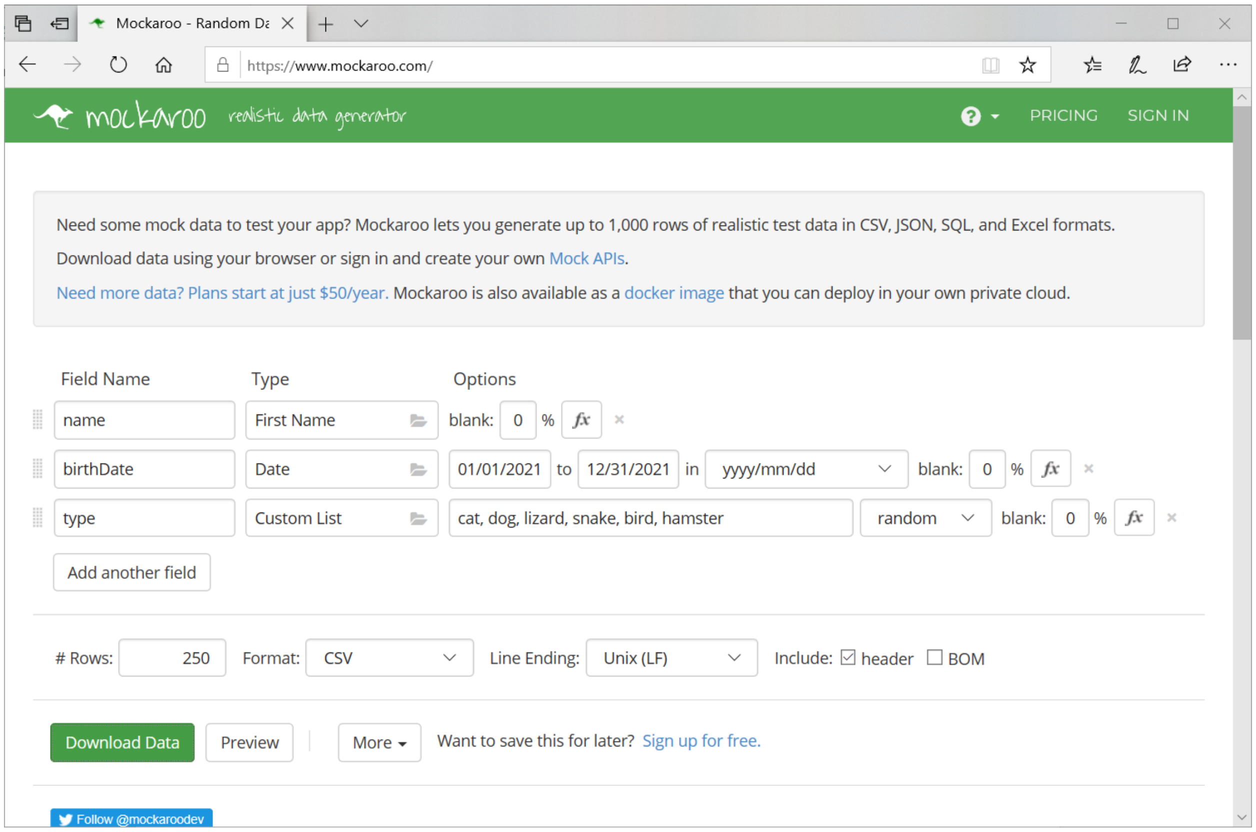
This report is dedicated to the extra task of this deliverable, the A+. The theme is Gatling feeders. Before doing the task itself I wanted to make a little research. I used the following links for that:

- https://gatling.io/docs/current/session/feeder
- https://gatling.io/docs/current/advanced_tutorial

I also used an additional resource for making the CSV files. The link for it is the following:

- https://www.mockaroo.com/

So, the first step was actually making the CSV files. You can check the configuration I used for that in Mockaroo in the next screenshot:



Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, SQL, and Excel formats.
Download data using your browser or sign in and create your own Mock APIs.
Need more data? Plans start at just $50/year. Mockaroo is also available as a docker image that you can deploy in your own private cloud.

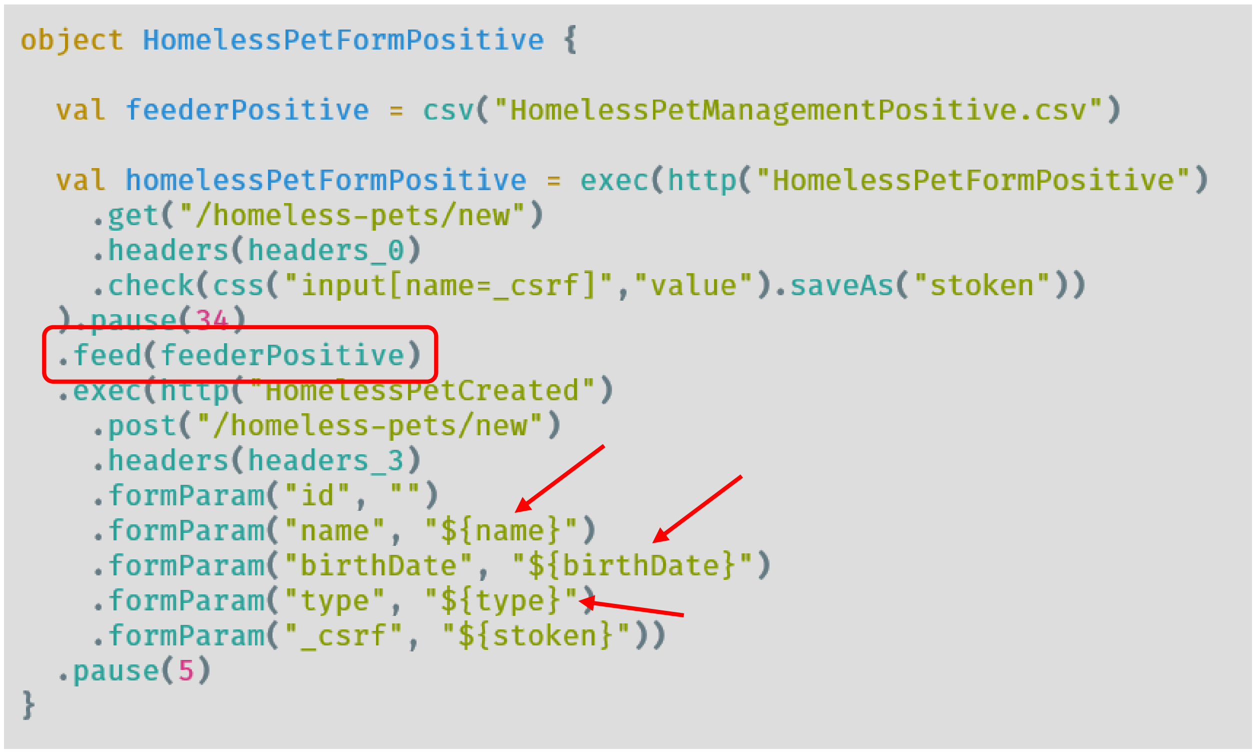
Field Name	Type	Options
name	First Name	blank: 0 %
birthDate	Date	01/01/2021 to 12/31/2021 in yyyy/mm/dd blank: 0 %
type	Custom List	cat, dog, lizard, snake, bird, hamster random blank: 0 %

Add another field

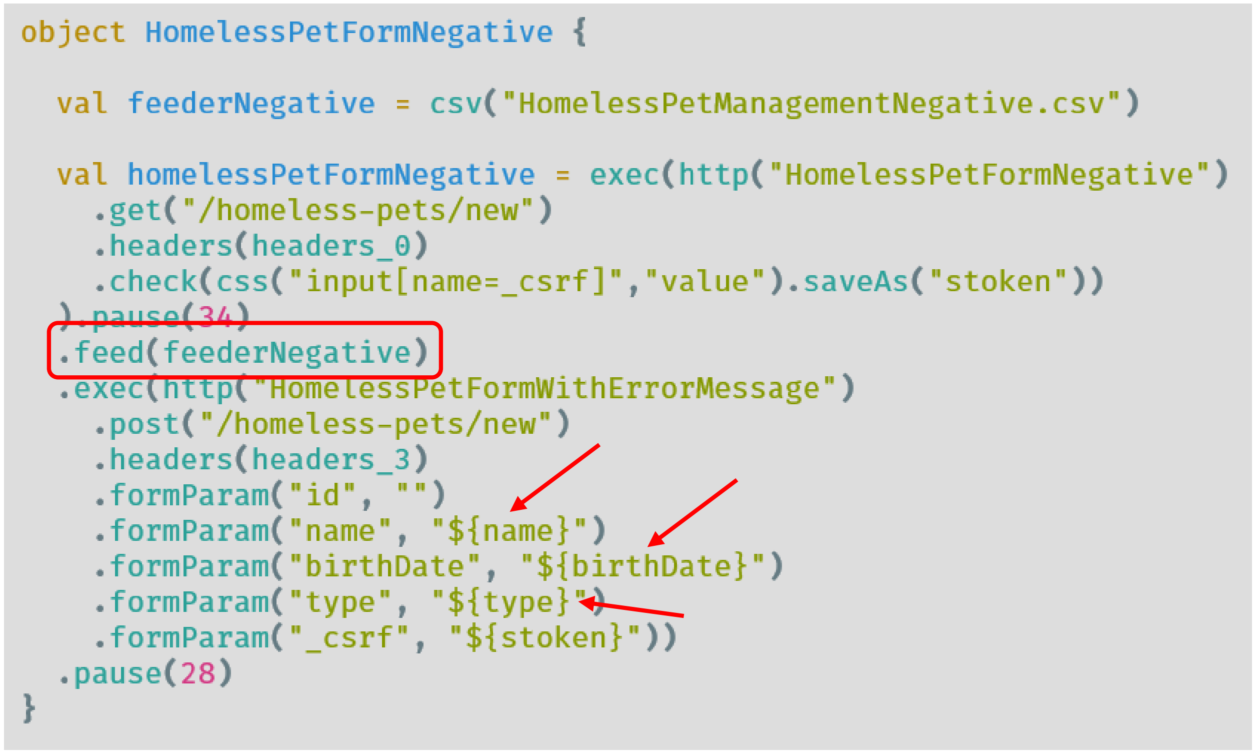
Rows: 250 Format: CSV Line Ending: Unix (LF) Include: ☑ header ☐ BOM

The user story for which I made this is US – 011 Homeless Pet Management in which I had the positive and negative scenario of creating a new pet. For that I needed a different name each time I inserted a new pet. And also, I used random values for the other fields to make it more interesting. The configuration in the previous screenshot is actually only for the negative scenario because the birth dates are in future (impossible to insert a pet with that characteristics). The procedure for the positive scenario is similar anyways.

The second step was actually modifying the Scala file of the scenarios. I needed to add the things in red in the next screenshots:



```
object HomelessPetFormPositive {

  val feederPositive = csv("HomelessPetManagementPositive.csv")

  val homelessPetFormPositive = exec(http("HomelessPetFormPositive")
    .get("/homeless-pets/new")
    .headers(headers_0)
    .check(css("input[name=_csrf]","value").saveAs("stoken"))
  ).pause(34)
  .feed(feederPositive)
  .exec(http("HomelessPetCreated")
    .post("/homeless-pets/new")
    .headers(headers_3)
    .formParam("id", "")
    .formParam("name", "${name}")
    .formParam("birthDate", "${birthDate}")
    .formParam("type", "${type}")
    .formParam("_csrf", "${stoken}"))
  .pause(5)
}
```

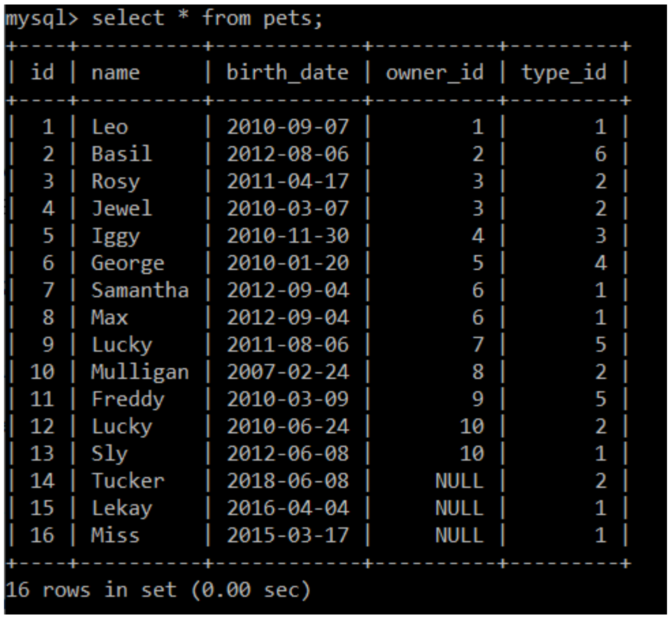


```
object HomelessPetFormNegative {

  val feederNegative = csv("HomelessPetManagementNegative.csv")

  val homelessPetFormNegative = exec(http("HomelessPetFormNegative")
    .get("/homeless-pets/new")
    .headers(headers_0)
    .check(css("input[name=_csrf]","value").saveAs("stoken"))
  ).pause(34)
  .feed(feederNegative)
  .exec(http("HomelessPetFormWithErrorMessage")
    .post("/homeless-pets/new")
    .headers(headers_3)
    .formParam("id", "")
    .formParam("name", "${name}")
    .formParam("birthDate", "${birthDate}")
    .formParam("type", "${type}")
    .formParam("_csrf", "${stoken}"))
  .pause(28)
}
```

To edit the Scala file, I used this resource: https://scastie.scala-lang.org/

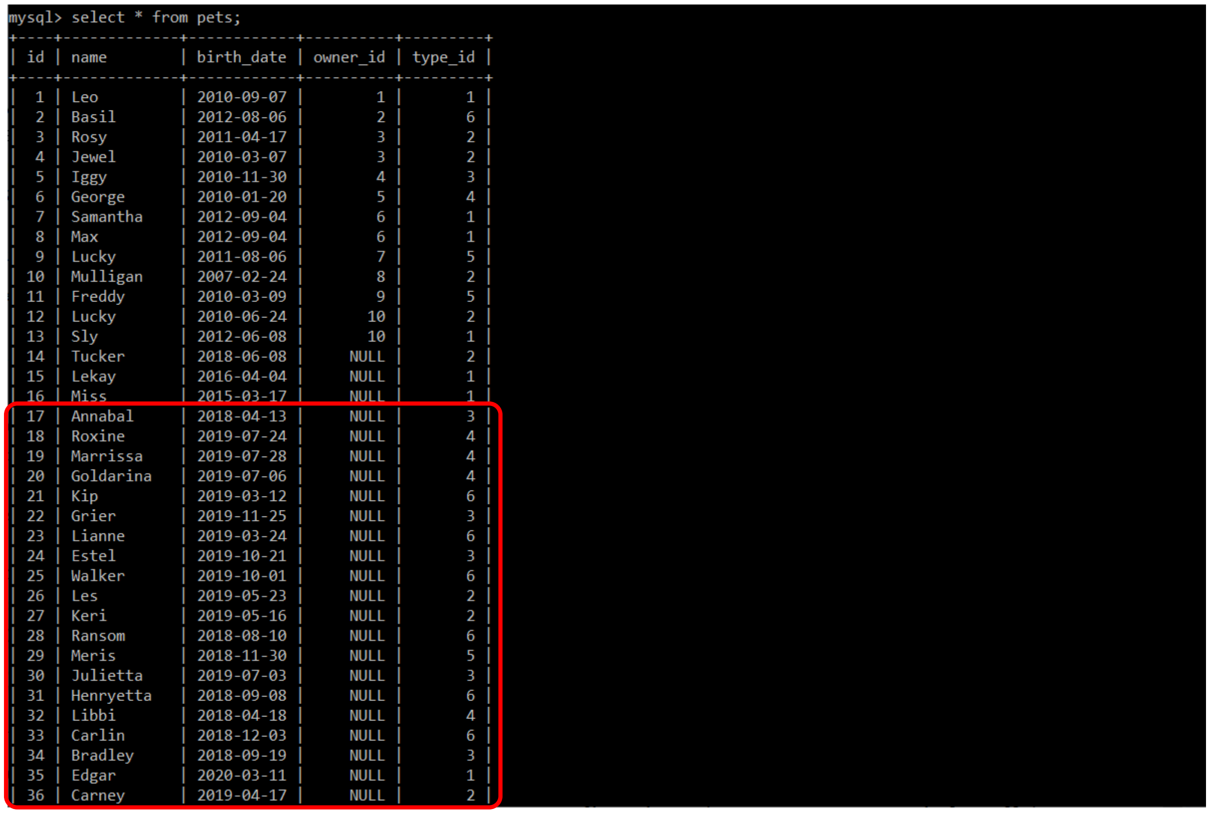
The third step was to actually start the database and launch the application and after that I executed the script saved before.

These are the pets before executing the script:



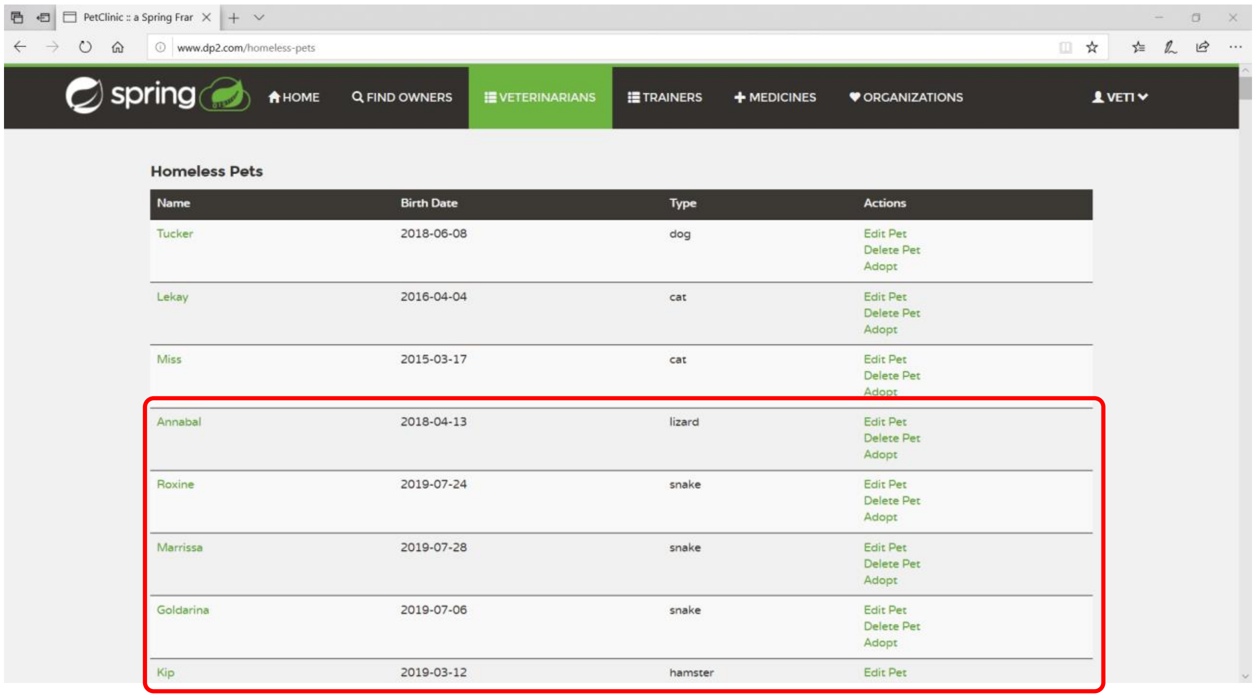
```
mysql> select * from pets;
+----+----------+------------+----------+---------+
| id | name     | birth_date | owner_id | type_id |
+----+----------+------------+----------+---------+
|  1 | Leo      | 2010-09-07 |        1 |       1 |
|  2 | Basil    | 2012-08-06 |        2 |       6 |
|  3 | Rosy     | 2011-04-17 |        3 |       2 |
|  4 | Jewel    | 2010-03-07 |        3 |       2 |
|  5 | Iggy     | 2010-11-30 |        4 |       3 |
|  6 | George   | 2010-01-20 |        5 |       4 |
|  7 | Samantha | 2012-09-04 |        6 |       1 |
|  8 | Max      | 2012-09-04 |        6 |       1 |
|  9 | Lucky    | 2011-08-06 |        7 |       5 |
| 10 | Mulligan | 2007-02-24 |        8 |       2 |
| 11 | Freddy   | 2010-03-09 |        9 |       5 |
| 12 | Lucky    | 2010-06-24 |       10 |       2 |
| 13 | Sly      | 2012-06-08 |       10 |       1 |
| 14 | Tucker   | 2018-06-08 |     NULL |       2 |
| 15 | Lekay    | 2016-04-04 |     NULL |       1 |
| 16 | Miss     | 2015-03-17 |     NULL |       1 |
+----+----------+------------+----------+---------+
16 rows in set (0.00 sec)
```

After executing the script, the following screenshot shows part of the query result I got from the database:



```
mysql> select * from pets;
+----+------------+------------+----------+---------+
| id | name       | birth_date | owner_id | type_id |
+----+------------+------------+----------+---------+
|  1 | Leo        | 2010-09-07 |        1 |       1 |
|  2 | Basil      | 2012-08-06 |        2 |       6 |
|  3 | Rosy       | 2011-04-17 |        3 |       2 |
|  4 | Jewel      | 2010-03-07 |        3 |       2 |
|  5 | Iggy       | 2010-11-30 |        4 |       3 |
|  6 | George     | 2010-01-20 |        5 |       4 |
|  7 | Samantha   | 2012-09-04 |        6 |       1 |
|  8 | Max        | 2012-09-04 |        6 |       1 |
|  9 | Lucky      | 2011-08-06 |        7 |       5 |
| 10 | Mulligan   | 2007-02-24 |        8 |       2 |
| 11 | Freddy     | 2010-03-09 |        9 |       5 |
| 12 | Lucky      | 2010-06-24 |       10 |       2 |
| 13 | Sly        | 2012-06-08 |       10 |       1 |
| 14 | Tucker     | 2018-06-08 |     NULL |       2 |
| 15 | Lekay      | 2016-04-04 |     NULL |       1 |
| 16 | Miss       | 2015-03-17 |     NULL |       1 |
| 17 | Annabal    | 2018-04-13 |     NULL |       3 |
| 18 | Roxine     | 2019-07-24 |     NULL |       4 |
| 19 | Marrissa   | 2019-07-28 |     NULL |       4 |
| 20 | Goldarina  | 2019-07-06 |     NULL |       4 |
| 21 | Kip        | 2019-03-12 |     NULL |       6 |
| 22 | Grier      | 2019-11-25 |     NULL |       3 |
| 23 | Lianne     | 2019-03-24 |     NULL |       6 |
| 24 | Estel      | 2019-10-21 |     NULL |       3 |
| 25 | Walker     | 2019-10-01 |     NULL |       6 |
| 26 | Les        | 2019-05-23 |     NULL |       2 |
| 27 | Keri       | 2019-05-16 |     NULL |       2 |
| 28 | Ransom     | 2018-08-10 |     NULL |       6 |
| 29 | Meris      | 2018-11-30 |     NULL |       5 |
| 30 | Julietta   | 2019-07-03 |     NULL |       3 |
| 31 | Henryetta  | 2018-09-08 |     NULL |       6 |
| 32 | Libbi      | 2018-04-18 |     NULL |       4 |
| 33 | Carlin     | 2018-12-03 |     NULL |       6 |
| 34 | Bradley    | 2018-09-19 |     NULL |       3 |
| 35 | Edgar      | 2020-03-11 |     NULL |       1 |
| 36 | Carney     | 2019-04-17 |     NULL |       2 |
```

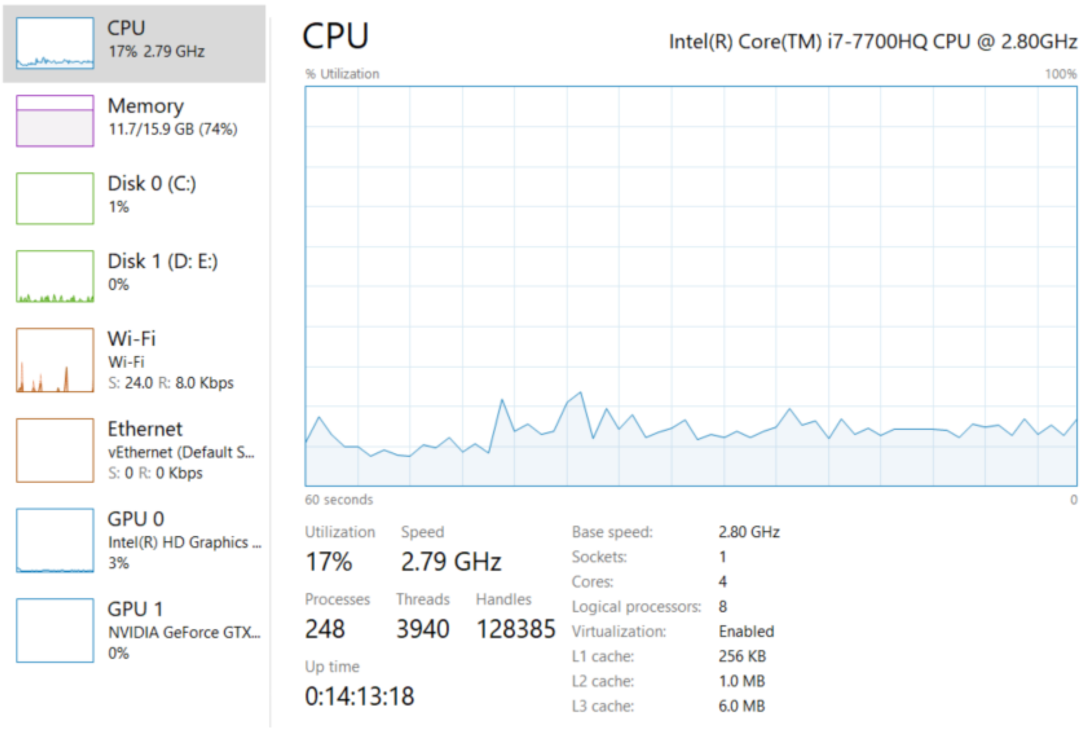
And we can also see them in the actual page:



Homeless Pets

Name	Birth Date	Type	Actions
Tucker	2018-06-08	dog	Edit Pet Delete Pet Adopt
Lekay	2016-04-04	cat	Edit Pet Delete Pet Adopt
Miss	2015-03-17	cat	Edit Pet Delete Pet Adopt
Annabal	2018-04-13	lizard	Edit Pet Delete Pet Adopt
Roxine	2019-07-24	snake	Edit Pet Delete Pet Adopt
Marrissa	2019-07-28	snake	Edit Pet Delete Pet Adopt
Goldarina	2019-07-06	snake	Edit Pet Delete Pet Adopt
Kip	2019-03-12	hamster	Edit Pet

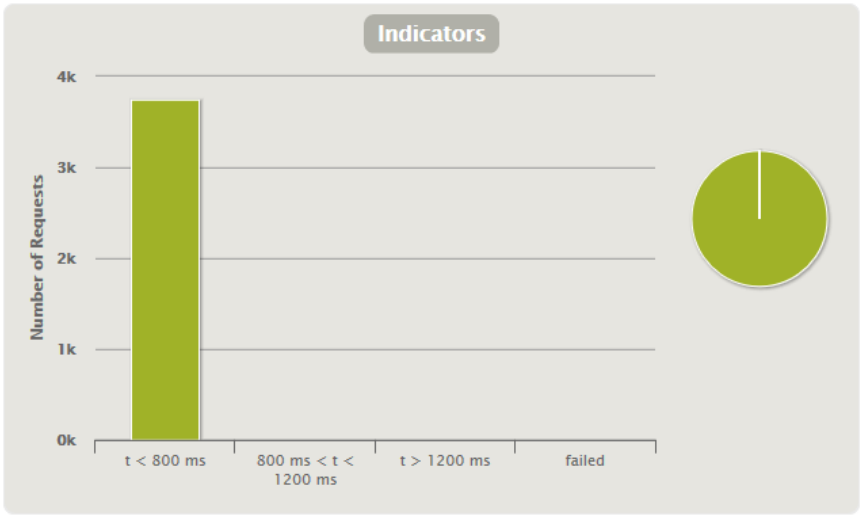
We can also check the performance results while using feeders. You can see the CPU overall performance during the execution of the script and some data from the Gatling reports on the next page.

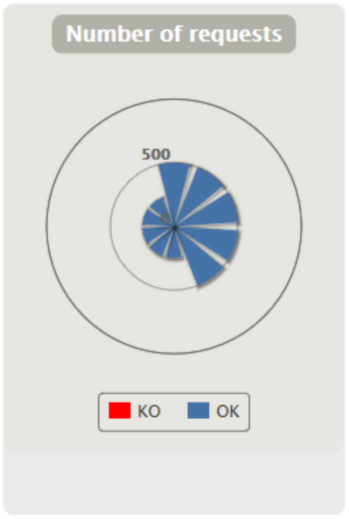


Utilization	Speed	
17%	2.79 GHz	
Processes	Threads	Handles
248	3940	128385
Up time		
0:14:13:18		

Base speed:	2.80 GHz
Sockets:	1
Cores:	4
Logical processors:	8
Virtualization:	Enabled
L1 cache:	256 KB
L2 cache:	1.0 MB
L3 cache:	6.0 MB

> Global Information





ASSERTIONS

Assertion	Status
Global: max of response time is less than 5000.0	OK
Global: mean of response time is less than 1000.0	OK
Global: percentage of successful events is greater than 95.0	OK

STATISTICS

Expand all groups | Collapse all groups

Requests	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
Global Information	3749	3749	0	0%	15.686	0	12	23	47	135	1314	20	54
Home	500	500	0	0%	2.092	2	5	7	16	670	1314	19	114
Login	500	500	0	0%	2.092	0	2	2	4	20	28	2	3
Logged	500	500	0	0%	2.092	4	8	11	18	48	287	11	23
Logged Redirect 1	500	500	0	0%	2.092	1	3	4	8	22	25	4	3
ListHomelessPets	500	500	0	0%	2.092	11	22	33	62	452	464	35	61
Homeless...Positive	250	250	0	0%	1.046	9	16	22	32	387	409	26	54
Homeless...Negative	250	250	0	0%	1.046	9	15	22	34	393	401	26	55
Homeless...rMessage	250	250	0	0%	1.046	16	25	36	48	53	74	29	10
HomelessPetCreated	250	250	0	0%	1.046	20	30	44	57	132	135	37	18
Homeless...direct 1	249	249	0	0%	1.042	12	33	44	58	70	82	34	13