



Aleksandar Karamfilov

Founder @ Pragmatic

Senior QA Manager @ Kiteworks

Performance & Load Testing with Gatling

Java binding



Следете актуалните обяви за **Software QA**

DEV.BG



Free/Open-source Performance tools

- JMeter (UI configuration, not much of coding)
- Gatling (Scala, Kotlin, Java)
- K6 (JavaScript)

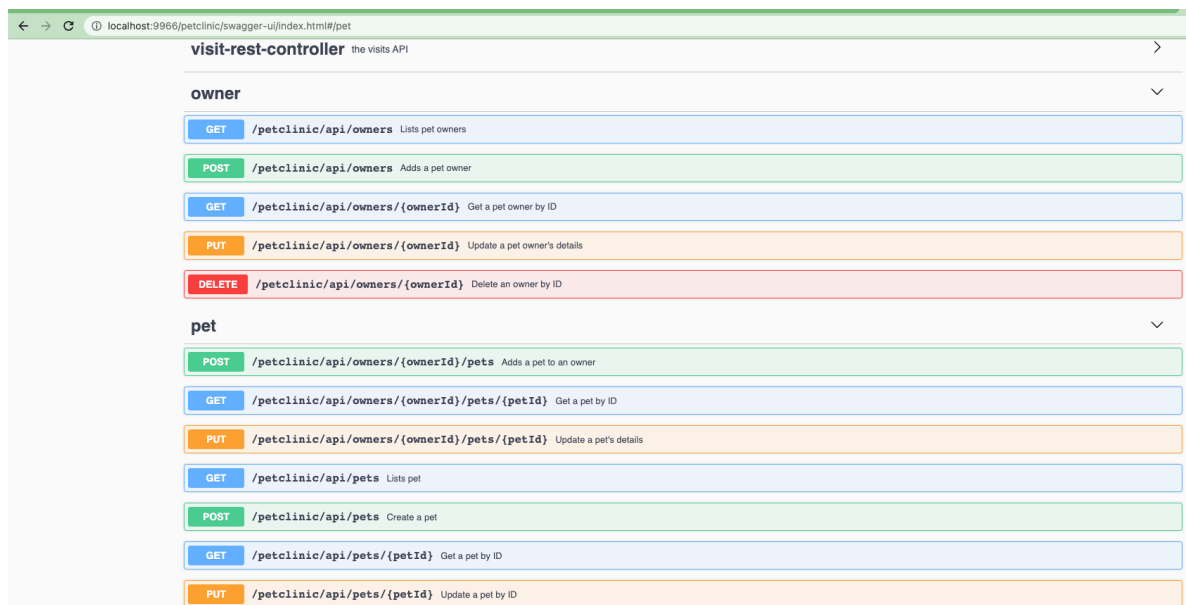




PetClinic REST API

(because going to prison is not an option)

<https://github.com/spring-petclinic/spring-petclinic-rest>



Следете актуалните обяви за **Software QA**

DEV.BG



Good old JMeter

The past or may be not 😊

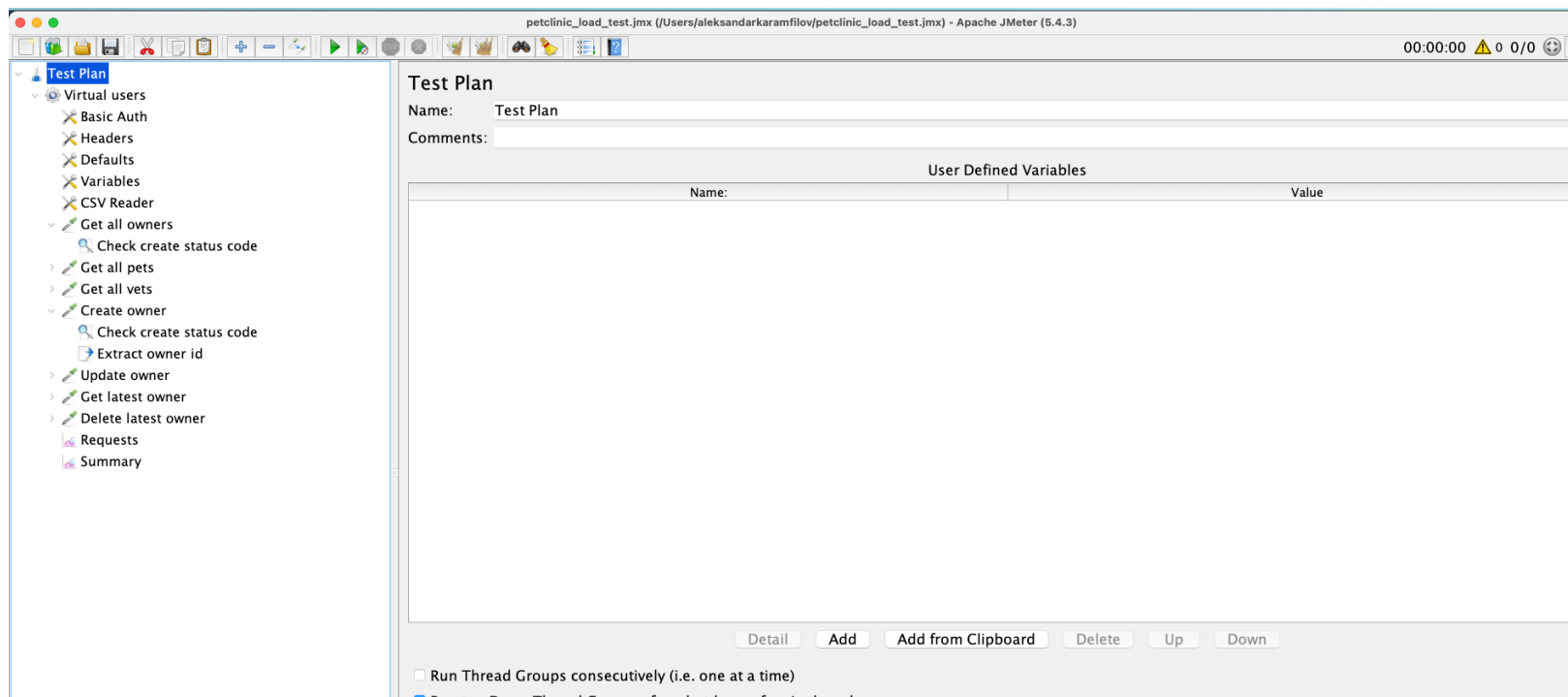


Следете актуалните обяви за **Software QA**

DEV.BG



Test plans

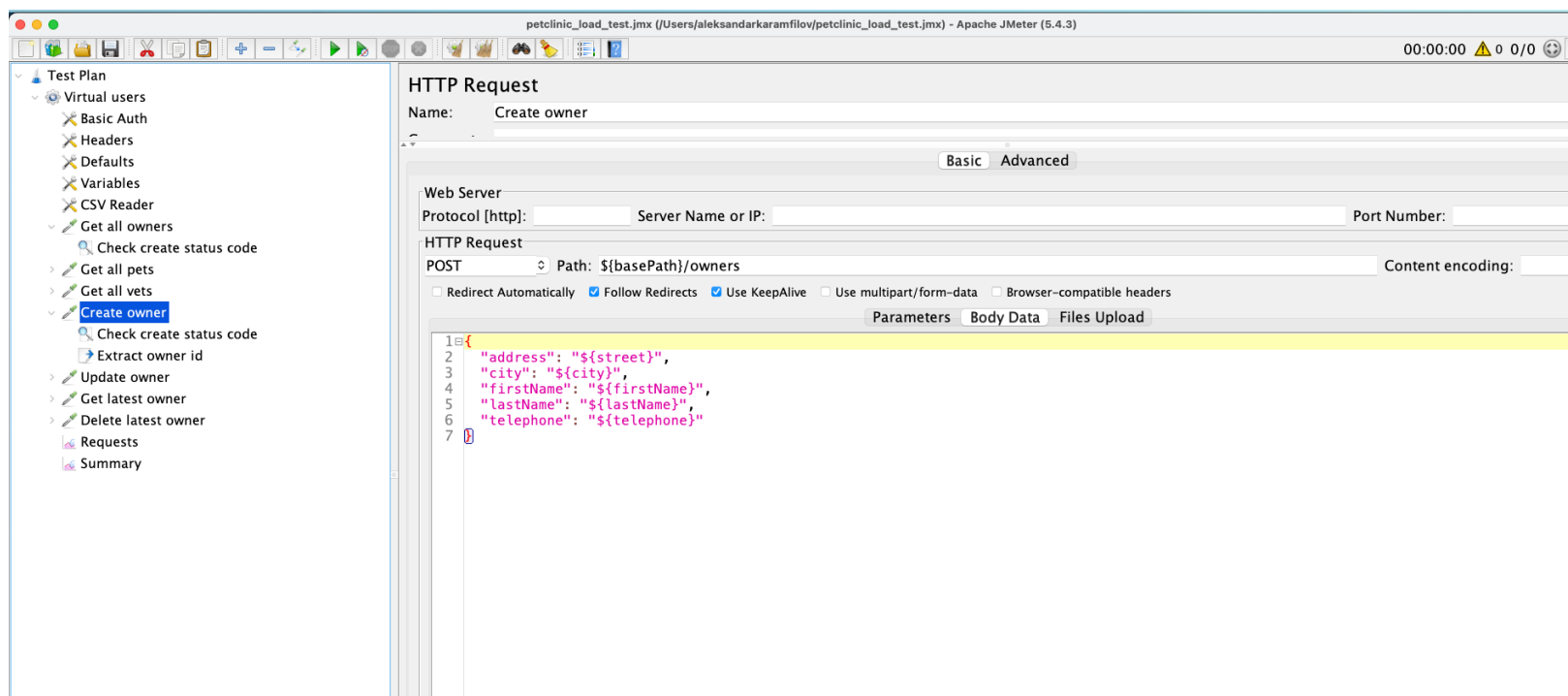


Следете актуалните обяви за **Software QA**

DEV.BG



Samplers

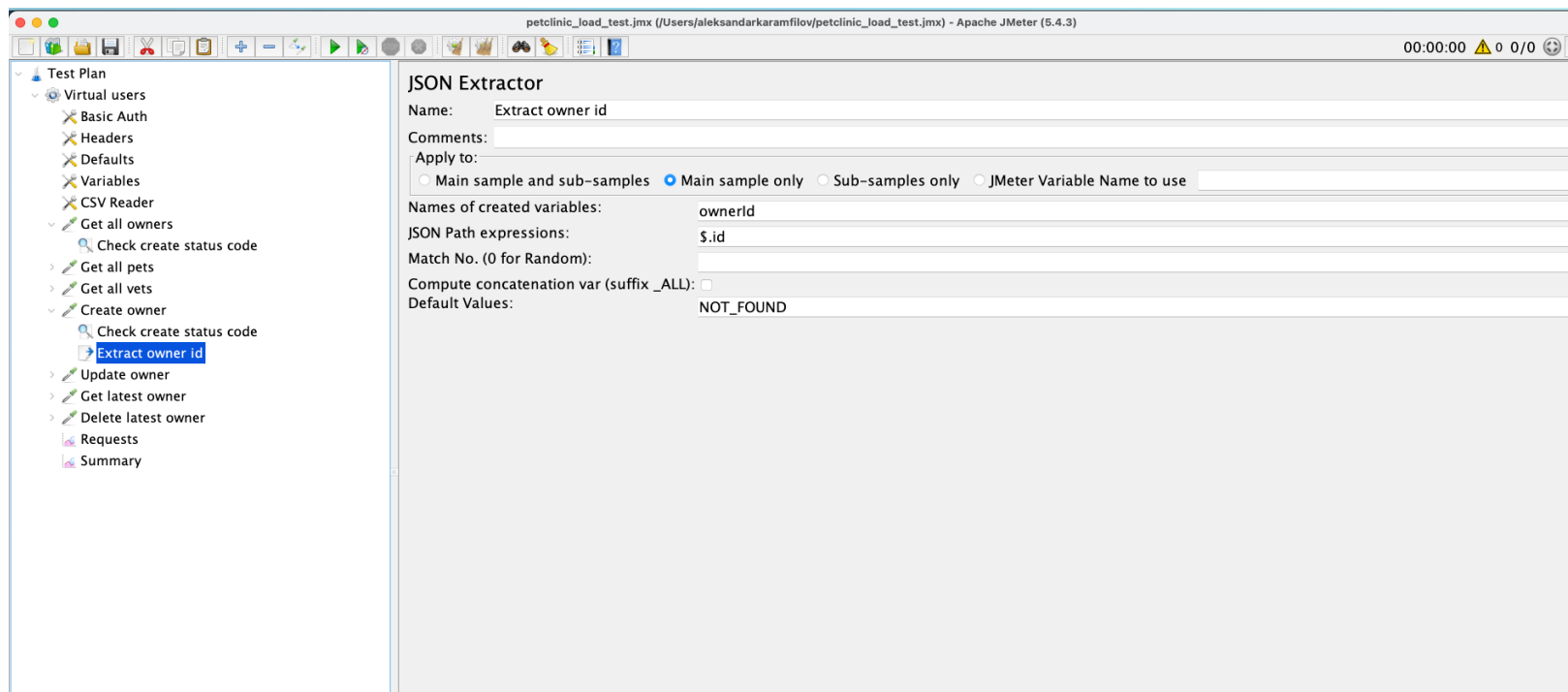


Следете актуалните обяви за **Software QA**

DEV.BG



Extractors



Следете актуалните обяви за **Software QA**

DEV.BG



Authorization managers

The screenshot shows the Apache JMeter 5.4.3 interface. On the left, the 'Test Plan' tree is expanded, showing 'Virtual users' with 'Basic Auth' selected. The main panel displays the 'HTTP Authorization Manager' configuration. The 'Name' field is set to 'Basic Auth'. The 'Options' section has two checkboxes: 'Clear auth on each iteration?' (unchecked) and 'Use Thread Group configuration to control clearing' (checked). Below this, a table titled 'Authorizations Stored in the Authorization Manager' is shown.

Base URL	Username	Password	Domain	Realm	Mechanism
http://localhost:9966/petcl...	\$(username)	*****			BASIC

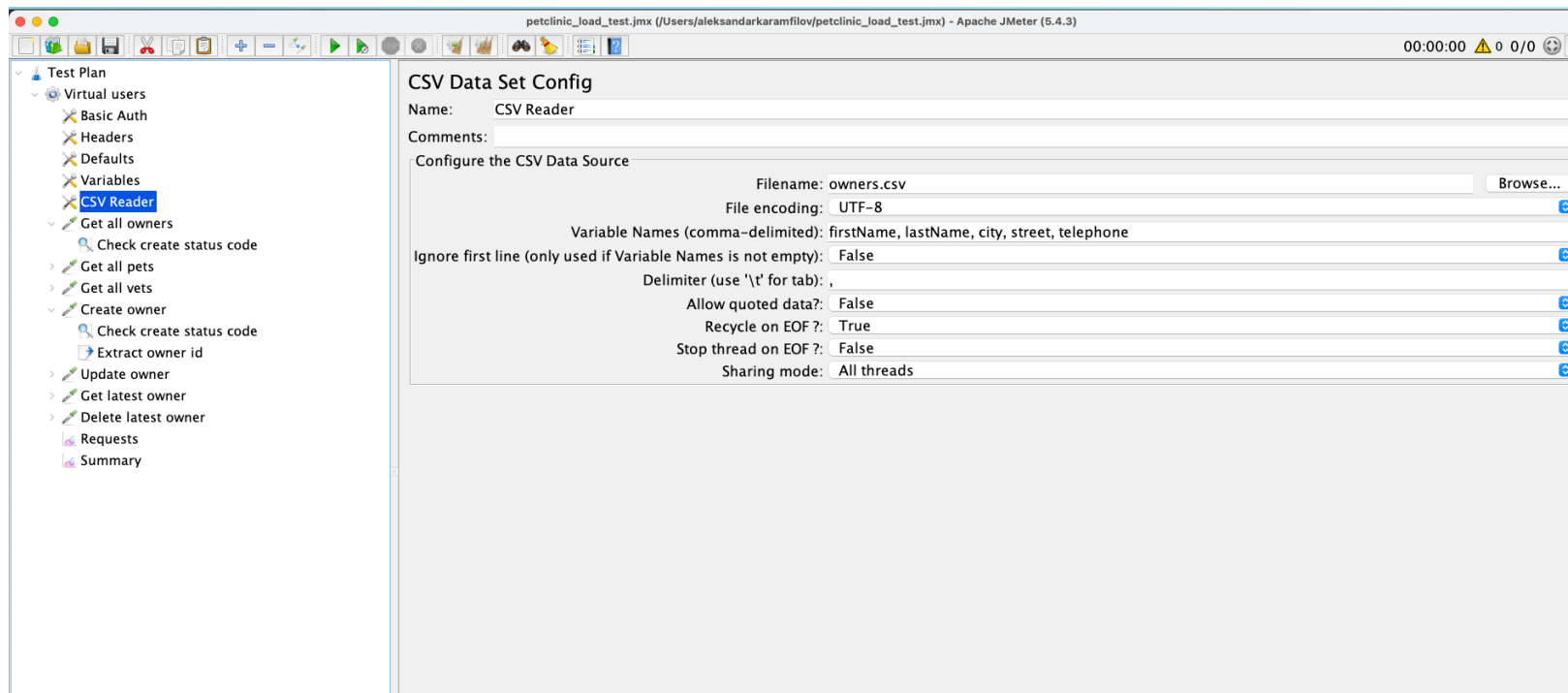


Следете актуалните обяви за **Software QA**

DEV.BG



CSV Readers



Следете актуалните обяви за **Software QA**

DEV.BG



Variables

petclinic_load_test.jmx (/Users/aleksandarkamfilov/petclinic_load_test.jmx) - Apache JMeter (5.4.3) 00:00:00 0 0/0

Test Plan

- Virtual users
 - Basic Auth
 - Headers
 - Defaults
 - Variables**
 - CSV Reader
- Get all owners
 - Check create status code
- Get all pets
- Get all vets
- Create owner
 - Check create status code
 - Extract owner id
- Update owner
- Get latest owner
- Delete latest owner
- Requests
- Summary

User Defined Variables

Name: Variables

Comments:

Name:	Value	Description
baseUri	localhost	
basePath	/petclinic/api	
port	9966	
protocol	http	
username	admin	
password	admin	



Следете актуалните обяви за **Software QA**

DEV.BG



Listeners

petclinic_load_test.jmx (/Users/aleksandarkaramfilov/petclinic_load_test.jmx) - Apache JMeter (5.4.3) 00:00:00 0/1

Test Plan

- Virtual users
 - Basic Auth
 - Headers
 - Defaults
 - Variables
 - CSV Reader
- Get all owners
 - Check create status code
- Get all pets
 - Get all vets
- Create owner
 - Update owner
 - Get latest owner
 - Delete latest owner
- Requests
- Summary

Summary Report

Name: Summary

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Get all owners	3	16	7	30	9.88	0.00%	22.1/sec	92.11	4.46	4276.0
Get all pets	3	10	6	18	5.66	0.00%	26.8/sec	56.06	5.36	2143.0
Get all vets	3	4	4	5	0.47	0.00%	30.6/sec	30.88	6.13	1033.0
Create owner	3	5	3	9	2.62	0.00%	30.9/sec	20.44	10.89	676.7
Update owner	3	4	4	5	0.47	0.00%	38.0/sec	18.13	14.49	489.0
Get latest owner	3	2	2	3	0.47	0.00%	39.5/sec	26.16	8.10	678.7
Delete latest o...	3	3	3	4	0.47	0.00%	40.0/sec	17.85	9.06	457.0
TOTAL	21	6	2	30	6.33	0.00%	131.2/sec	178.59	33.15	1393.3



Следете актуалните обяви за **Software QA**

DEV.BG



JMeter pros

- Supports a lot of protocols (http/sftp/smtp/you name it)
- No need to write code
- Good documentation
- Good amount of examples/trainings/books
- Big community
- Many custom plugins available





JMeter cons

- Limited ability to write code (few exceptions like abstracta library)
- Tricky(ugly) to integrate with CI
- Hard to review what is changed in the JMX file
- Hard to follow and debug
- Not very efficient in load generation





Gatling

Gatling is a powerful open-source load testing solution. Gatling is designed for continuous load testing and integrates with your development pipeline.



Следете актуалните обяви за **Software QA**

DEV.BG



Gatling with Maven

```
<dependencies>
  <dependency>
    <groupId>io.gatling.highcharts</groupId>
    <artifactId>gatling-charts-highcharts</artifactId>
    <version>MANUALLY_REPLACE_WITH_LATEST_VERSION</version>
    <scope>test</scope>
  </dependency>
</dependencies>

<plugin>
  <groupId>io.gatling</groupId>
  <artifactId>gatling-maven-plugin</artifactId>
  <version>MANUALLY_REPLACE_WITH_LATEST_VERSION</version>
</plugin>
```





Gatling main components

- Simulation
- Scenario
- Request
- Session
- Feeder
- Check
- Duration





Simulation (Test)

```
import io.gatling.javaapi.core.Simulation;  
import io.gatling.javaapi.http.HttpProtocolBuilder;  
import static io.gatling.javaapi.http.HttpDsl.http;
```

//Simulation is the parent class your tests must extend so Gatling can launch them.
`public class UploadFileSimulation extends Simulation`





Configure protocol for all requests

```
//Configure http protocol for all requests  
protected HttpProtocolBuilder httpProtocol = http.baseUrl(BASE_URI)  
    .acceptHeader("application/json")  
    .contentTypeHeader("application/json")  
    .header("X-API-VERSION", API_VERSION)  
    .header("Authorization", "Bearer " + api.getToken())  
    .userAgentHeader(USER_AGENT);
```





Use HTTP2

```
//Configure http protocol for all requests (http 2 enabled)  
protected HttpProtocolBuilder httpProtocol = http.baseUrl(BASE_URI)  
    .acceptHeader("application/json")  
    .contentTypeHeader("application/json")  
    .header("X-Accellion-Version", API_VERSION)  
    .header("Authorization", "Bearer " + api.getToken())  
    .userAgentHeader(USER_AGENT)  
    .enableHttp2();
```





HTTP Request (ChainBuilder)

```
ChainBuilder sendMailRequest = exec(http("Send mail request")
    .post(SEND_EMAIL_ENDPOINT)
    .header("returnEntity", "true")
    .body(StringBody(GSON.toJson(EmailReq.builder() //DTO to String
        .to(to)
        .files(files)
        .draft(false)
        .body(DataGenerator.text(100, 2000))
        .subject("Send Email - Load Test")
        .build()))))
    .check(status().is(201)));
```





Scenario (sequence of requests)

```
// Upload file scenario  
ScenarioBuilder uploadFileScn = scenario("Upload file scenario")  
    .forever()  
    .on(exec(createFolderRequest, initiateUploadRequest, uploadFileRequest));
```





Grouping requests (get total time)

```
// Upload and download file scenario with grouping
ScenarioBuilder uploadDownloadFileScn = scenario("Upload/Download file")
    .repeat(REPEAT).on(exec(createFolderRequest,
        group("Upload file group").on(exec(initiateUploadRequest,
            uploadFileRequest).pause(Duration.ofSeconds(THINK_TIME))), downloadFileRequest))
    .pause(Duration.ofSeconds(THINK_TIME));
```





Data Feeder

```
//Creating feeder data
Iterator<Map<String, Object>> feeder =
    Stream.generate((Supplier<Map<String, Object>>) () -> {
        String folderName = DataGenerator.nameWithDate("ArtificialUploadFD_" + FILE_SIZE + "MB_");
        return Collections.singletonMap("folderName", folderName);
    }).iterator();

//Using feeder in request by passing the key name with # {key}
ChainBuilder createFolderRequest = feed(feeder).exec(http("Create folder request")
    .post(FOLDERS_ENDPOINT + "/0/folders")
    .body(StringBody(GSON.toJson(FolderReq.builder().name("#{folderName}").build()))))
    .queryParams("returnEntity", true)
    .check(status().is(201))
```





Checks

```
.check(status().is(201))  
  //Put the folder id in session  
.check(jsonPath("$.id").saveAs("folderId")));
```





User load pattern and ramp up(setUp)

```
//Setup user load and ramp up  
setUp(uploadFileScn.injectOpen(rampUsers(WORKERS)  
    .during(Duration.ofSeconds(RAMP_UP))))  
    .protocols(httpProtocol);
```





Running simulations

```
mvn gatling:test -Dgatling.simulationClass=simulations.MyFirstSimulation
```



Следете актуалните обяви за **Software QA**

DEV.BG



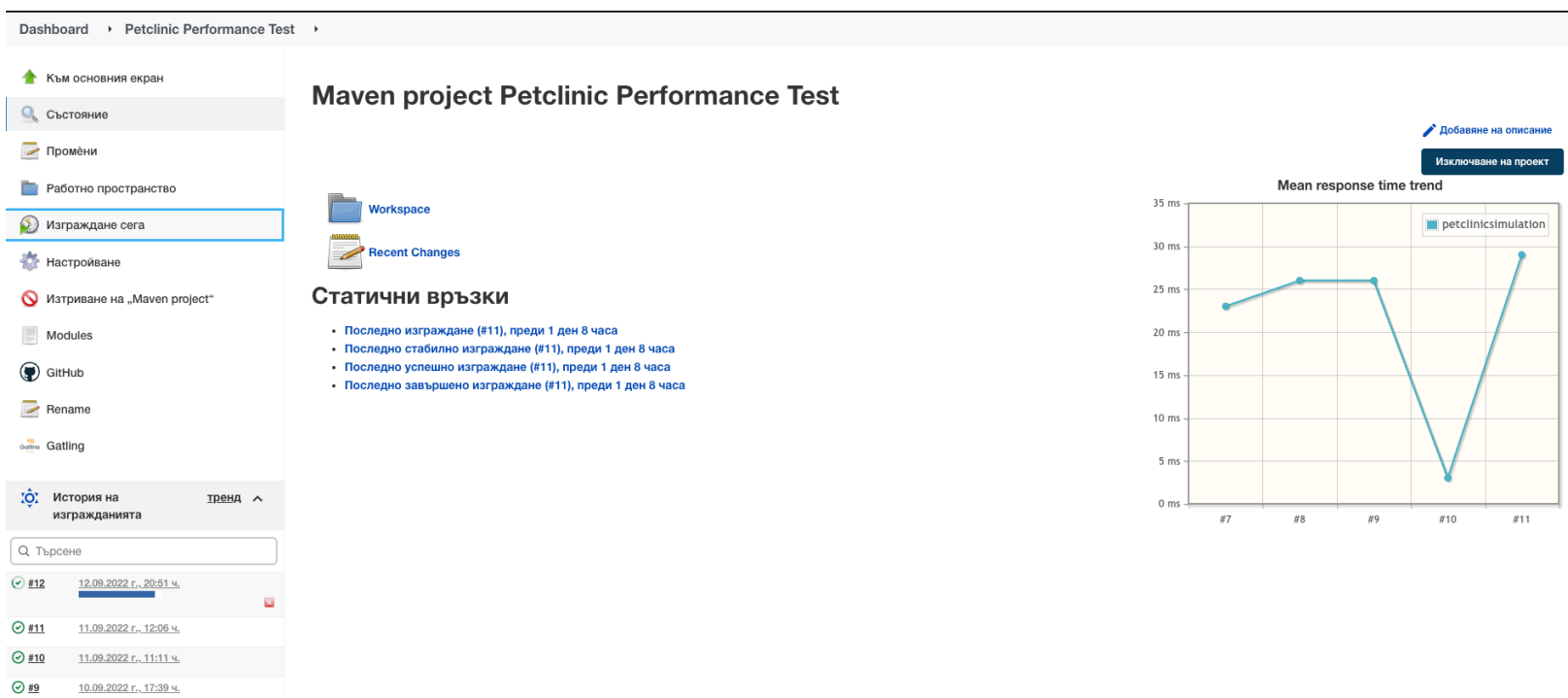
Configurations

```
52  ssl {
53      useOpenSsl = true           # If OpenSSL should be used instead of JSSE (only the latter can be debugged with -Djava.net.debug=ssl)
54      useOpenSslFinalizers = false # If OpenSSL contexts should be freed with Finalizer or if using RefCounted is fine
55      handshakeTimeout = 10000    # TLS handshake timeout in millis
56      useInsecureTrustManager = true # Use an insecure TrustManager that trusts all server certificates
57      enabledProtocols = []       # Array of enabled protocols for HTTPS, if empty use Netty's defaults
58      enabledCipherSuites = []   # Array of enabled cipher suites for HTTPS, if empty enable all available ciphers
59      sessionCacheSize = 0       # SSLSession cache size, set to 0 to use JDK's default
60      sessionTimeout = 0         # SSLSession timeout in seconds, set to 0 to use JDK's default (24h)
61      enableSni = true           # When set to true, enable Server Name indication (SNI)
62      keyStore {
63          type = ""              # Type of SSLContext's KeyManagers store
64          file = ""              # Location of SSLContext's KeyManagers store
65          password = ""          # Password for SSLContext's KeyManagers store
66          algorithm = ""         # Algorithm used SSLContext's KeyManagers store
67      }
68      trustStore {
69          type = ""              # Type of SSLContext's TrustManagers store
70          file = ""              # Location of SSLContext's TrustManagers store
71          password = ""          # Password for SSLContext's TrustManagers store
72          algorithm = ""         # Algorithm used by SSLContext's TrustManagers store
73      }
74  }
75  charting {
76      noReports = false          # When set to true, don't generate HTML reports
77      maxPlotPerSeries = 1000    # Number of points per graph in Gatling reports
78      useGroupDurationMetric = false # Switch group timings from cumulated response time to group duration.
79      indicators {
80          lowerBound = 000       # Lower bound for the requests' response time to track in the reports and the console summary
81          higherBound = 1200     # Higher bound for the requests' response time to track in the reports and the console summary
82          percentile1 = 50       # Value for the 1st percentile to track in the reports, the console summary and Graphite
83          percentile2 = 75       # Value for the 2nd percentile to track in the reports, the console summary and Graphite
84          percentile3 = 95       # Value for the 3rd percentile to track in the reports, the console summary and Graphite
85          percentile4 = 99       # Value for the 4th percentile to track in the reports, the console summary and Graphite
86      }
87  }
88  http {
89      fetchedCssCacheMaxCapacity = 200 # Cache size for CSS parsed content, set to 0 to disable
90      fetchedHtmlCacheMaxCapacity = 200 # Cache size for HTML parsed content, set to 0 to disable
91      perUserCacheMaxCapacity = 200    # Per virtual user cache size, set to 0 to disable
92      warmUpUrl = "https://gatling.io" # The URL to use to warm-up the HTTP stack (blank means disabled)
93      pooledConnectionIdleTimeout = 60000 # Timeout in millis for a connection to stay idle in the pool
```





Continuous performance testing



Следете актуалните обяви за **Software QA**

DEV.BG



Gatling pros

- Everything is code (easy to review and manage)
- Effective in load generation
- Good documentation and logical DSL
- Good reporting
- Easy to plug in CI/CD





Gatling cons

- Immutability can be quite confusing
- Does not support all protocols you might need (mostly http)
- Java DSL is the same as Scala DSL (bit weird for Java developers)
- Configuration is bit abstract
- Hard to find advanced usage examples





K6

Grafana k6 is an open-source load testing tool that makes performance testing easy and productive for engineering teams. k6 is free, developer-centric, and extensible.



Следете актуалните обяви за **Software QA**

DEV.BG



K6 Tests

```
export let options = {
  thresholds: {
    'http_req_duration{type:slow}': ['p(95)<300'], // threshold on API requests only
    'http_req_duration{type:api}': ['p(95)<200'], // threshold on search api only
  },
};

const basicAuthToken = "Basic YWRtaW46YWRtaW4="
const baseUrl = "http://localhost:9966/petclinic/api"
const params = { headers: { 'Content-Type': 'application/json', "Authorization": basicAuthToken }, tags: {type: 'api'} };

export default function () {
  group('get all pets', function () {
    //Get all pets
    let res = http.get(baseUrl + "/pets", params)
    check(res, {
      'Get pet status code is 200': (r) => r.status == 200,
      'Body includes animal names': (r) => r.body.includes('Leo')
    });
  });

  group('create new owner and get it', function () {
    //Create owner
    let data = JSON.stringify({
      address: "Ivan Stranski",
      city: "Sofia",
      firstName: "Alex" + Math.random(),
      lastName: "Karamfilov",
    });
  });
}
```





K6 Tests

```
});  
let createResponse = http.post(`${baseUrl}/vets`, data, params)  
check(createResponse, {  
  'Create vet status code is 201': (r) => r.status == 201,  
  'Create vet body includes name of the vet': (r) => r.body.includes('Alex')  
});  
  
//Delete vet  
let vetId = JSON.parse(createResponse.body).id;  
let deleteResponse = http.del(`${baseUrl}/vets/${vetId}`, params)  
check(deleteResponse, {  
  'Delete vet status code is 204': (r) => r.status == 204,  
});  
  
})  
  
group('Get owner by name', function(){  
  //Get owner by name  
  let lastName = 'Davis';  
  params.tags = {type: 'slow'}  
  let getOwner = http.get(`${baseUrl}/owners/*/lastname/${lastName}`, params)  
  check(getOwner, {  
    'Get owner by last name status code is 200': (r) => r.status == 200,  
    'Get owners by last name body includes name of the owner': (r) => r.body.includes(lastName)  
  });  
})
```





K6 pros

- Everything is code (easy to review and manage)
- Effective in load generation
- Good documentation and logical DSL
- Good reporting
- Easy to plug in CI/CD





K6 cons

- Need to write code in JS
- Does not support all protocols you might need (mostly http)
- Hard to find advanced usage examples because it is still new





Talk is cheap 😊



Следете актуалните обяви за **Software QA**

DEV.BG

Thank you!

Contacts:

 <https://www.linkedin.com/in/akaramfilov>

 <https://github.com/karamfilovs>

СЛЕДВАЩО СЪБИТИЕ



Лектор

Дата

Език



Следете актуалните обяви за **Software QA**

DEV.BG