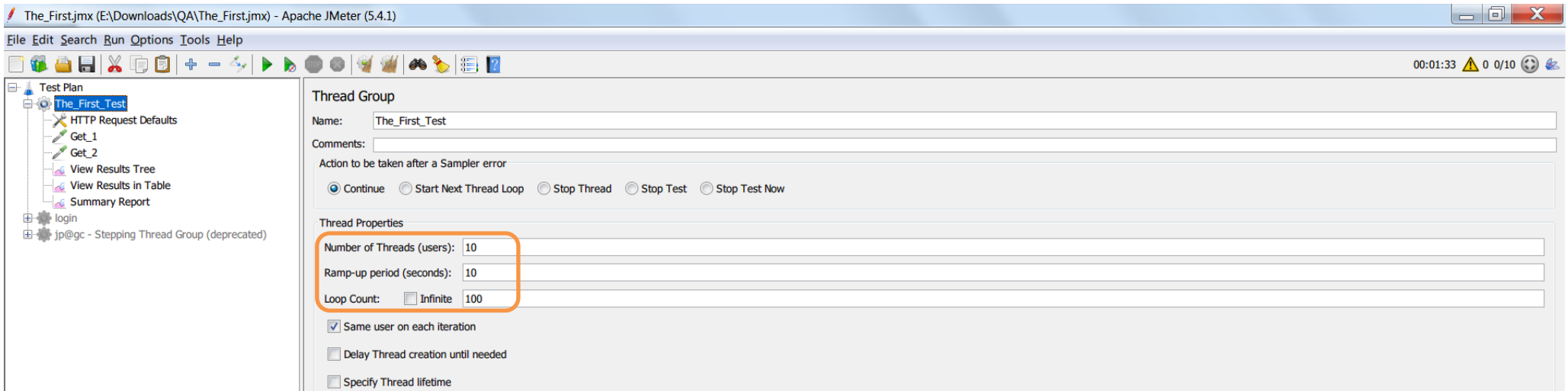


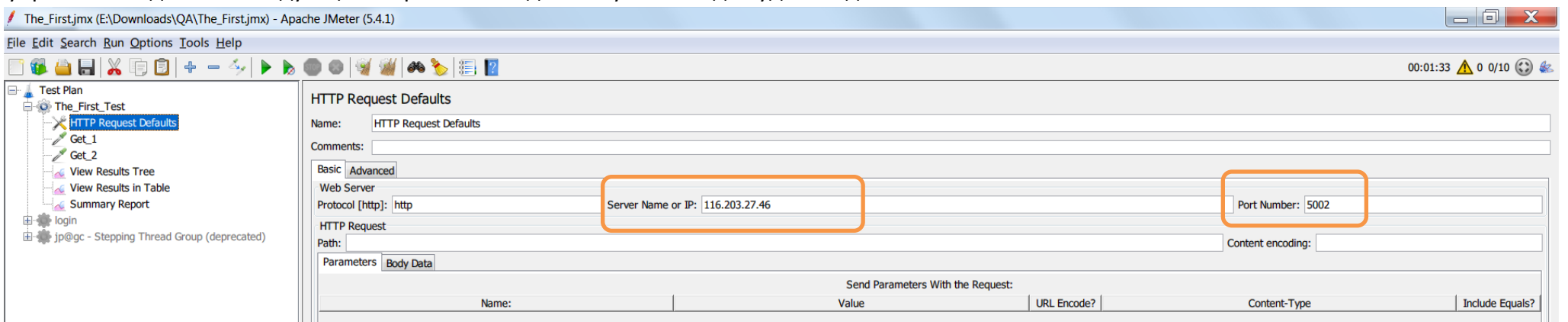
Проведение простого нагрузочного тестирования с использованием JMeter на тестовом сервере.

1. Создаем тестовый план (*add → Threads(Users) → Thread Group*), указываем параметры тестирования:

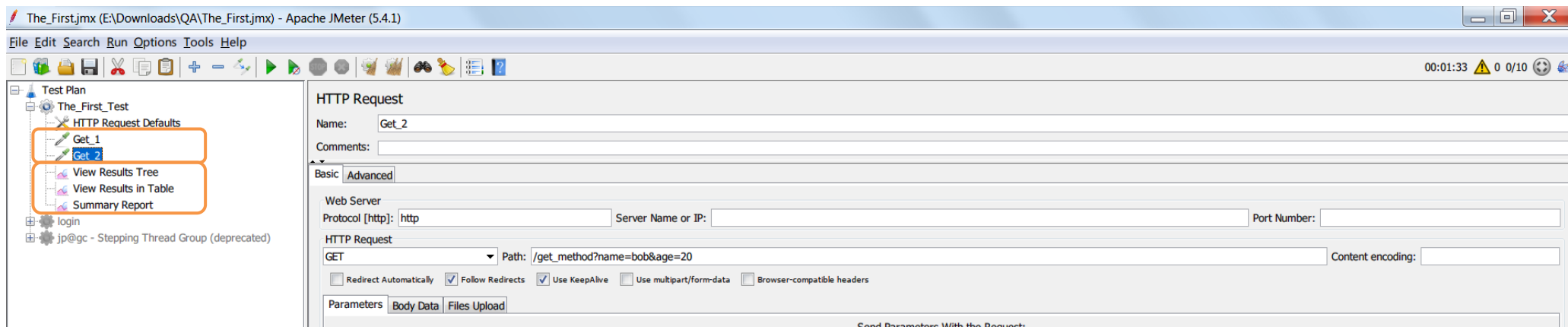
- **Number of Threads** - количество потоков выполняющих тестовые сценарии одновременно,
- **Ramp-Up period** - время, в течение которого будут прибавляться юзеры. Позволяет делать плавный и прогнозируемый старт,
- **Loop Count** - сколько раз выполнит сценарий каждый из потоков.



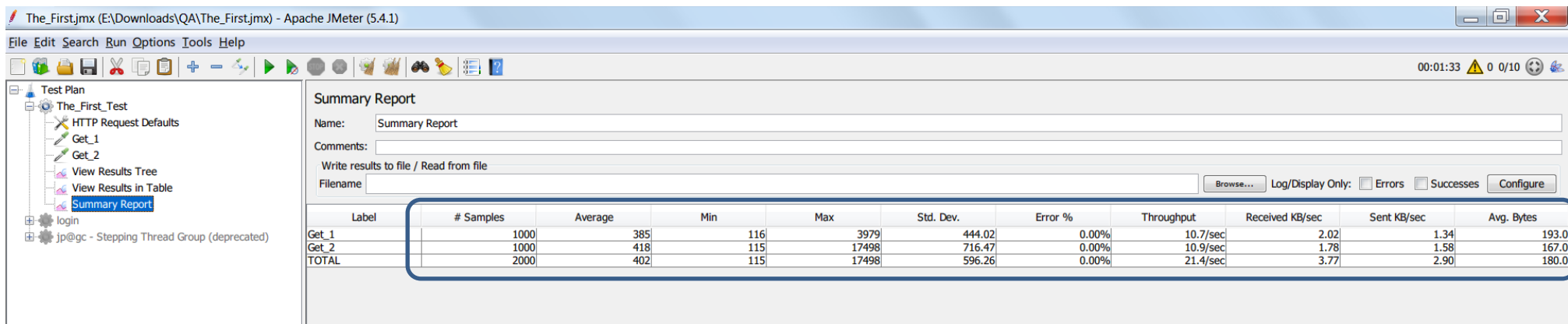
2. Добавляем (*add → Config Element → HTTP Request Defaults*) и настраиваем параметры HTTP запросов, указываем ip сервера и порт. Это позволит упростить создание последующих запросов – эти данные уже не надо будет вводить.



3. Добавляем два простых GET запроса (*add* → *Sampler* → *HTTP Request*) и Listener (слушателей): *View Results Tree*, *View Results in Table*, *Summary Report* (*add* → *Listener* → ...) для наглядного отображения хода и результатов тестирования.



4. Производим запуск теста, по завершению – анализируем результаты.



Samples – количество запросов. **Average** - среднее время ответа; **Min** и **Max** – минимальное и максимальное, соответственно.

Std Dev — показатель стандартного отклонения. Этот показатель позволяет оценить насколько сильно значения из выборки (результата тестового прогона) отличаются от рассчитанного среднего значения.

Error %—количество ошибок в процентах, которые вернул сервер.

Throughput - количество запросов в секунду.

Received и **Sent KB/sec** — количество полученных и отправленных данных.

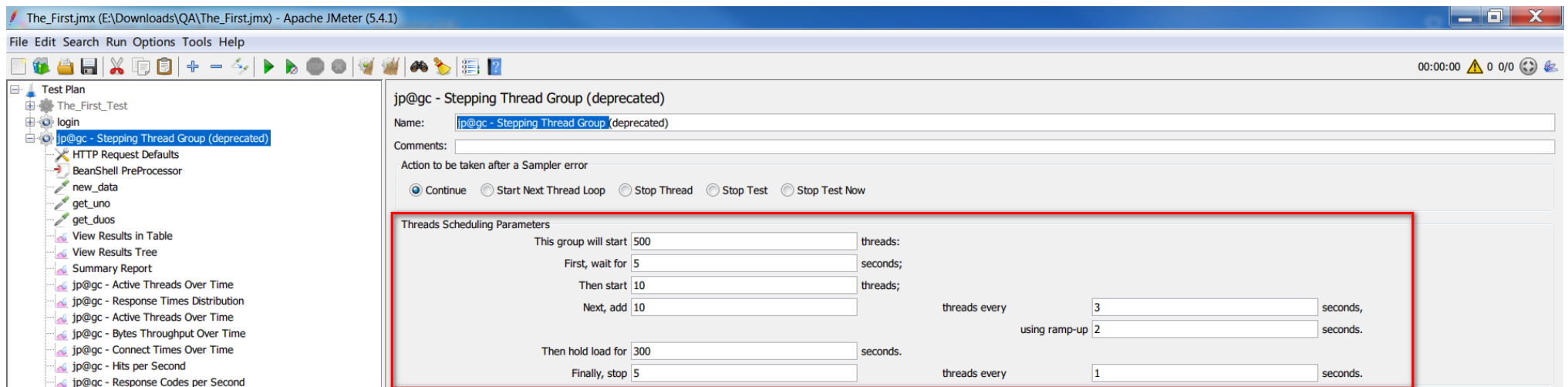
Avg.Bytes — среднее количество полученных данных.

Проведение простого нагрузочного тестирования с использованием плагина jp@gc - Stepping Thread Group.

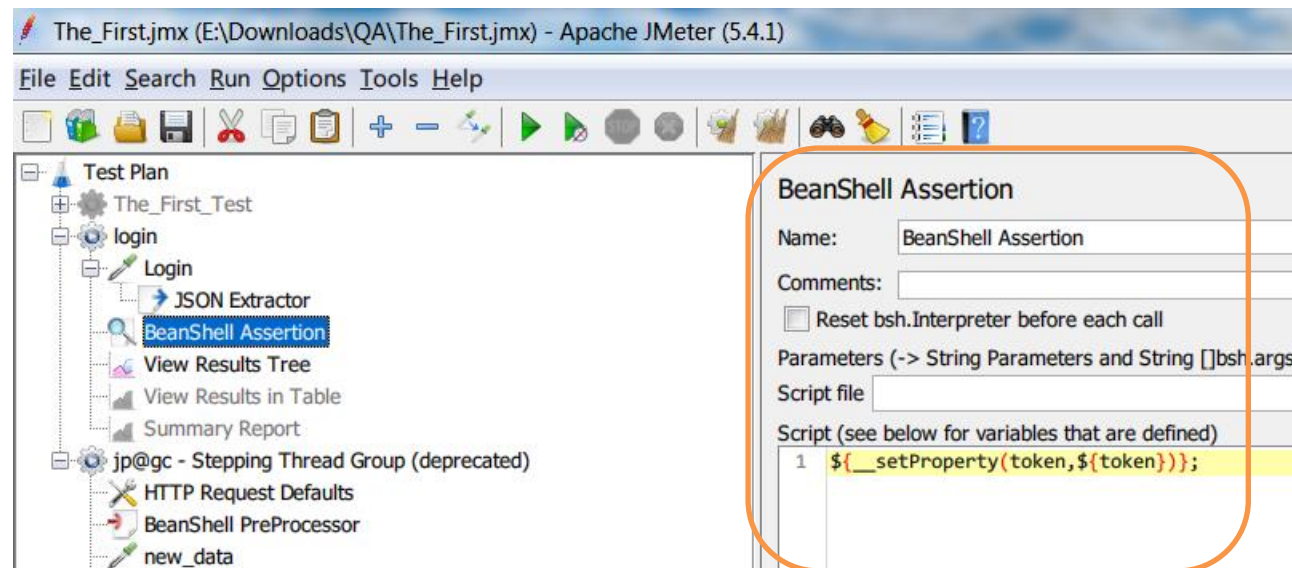
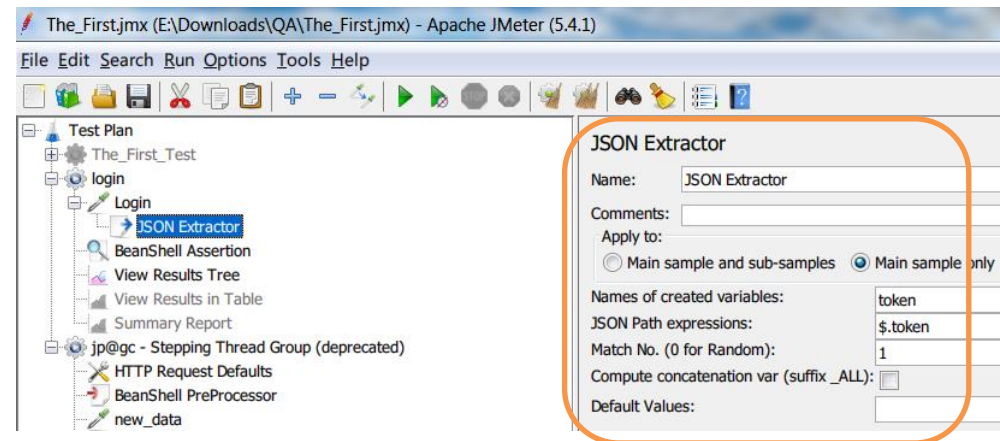
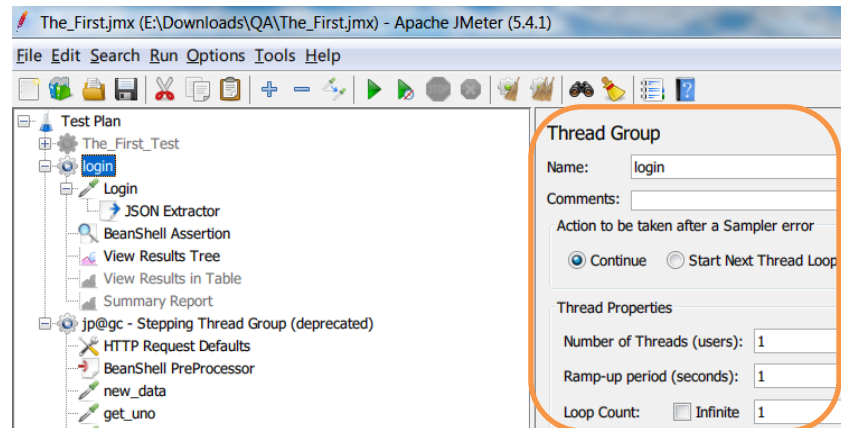
Stepping Thread Group - еще один кастомный Thread Group от JMeter-Plugins. Он не имеет бесконечного количества планировщиков, как Ultimate Thread Group, но имеет более гибкий алгоритм планирования в отличие от стандартного Thread Group. Для Stepping Thread Group мы можем указать следующее:

- количество пользователей;
- задержку перед стартом;
- активирование определенного количества в пользователей с заданным интервалом;
- время работы после активирования всех пользователей;
- завершение работы определенного количества в пользователей с заданным интервалом.

Указываем эти параметры:

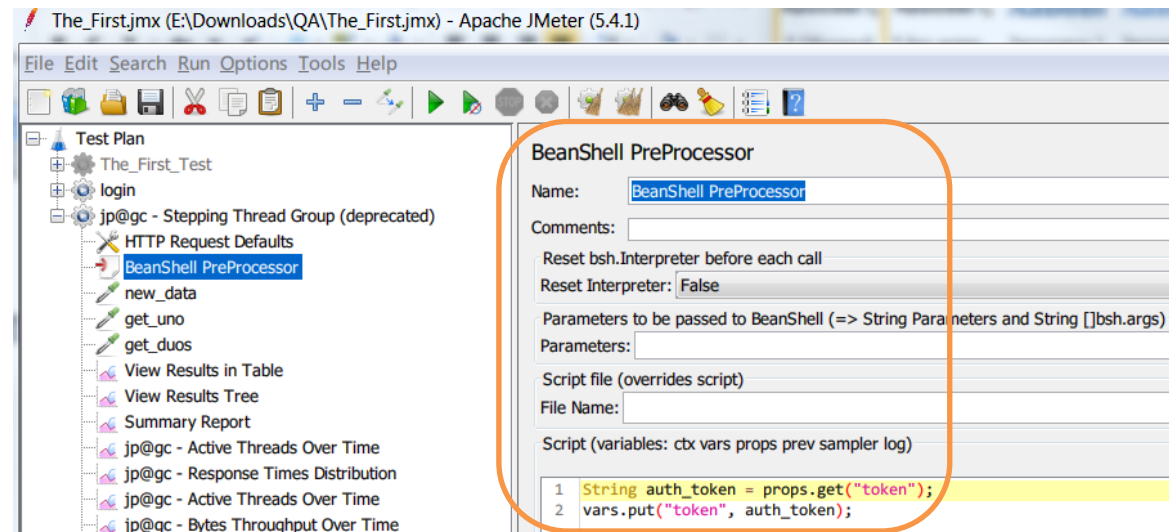


Прохождение тестирования немного усложним. Сначала разово выполним запрос на тестовый сервер для получения токена для аутентификации. При помощи пост-процессора JSON Extractor извлечем из ответа сервера токен и используя BeanShell Assertion сохраним значение токена в переменную для использования в последующих запросах.

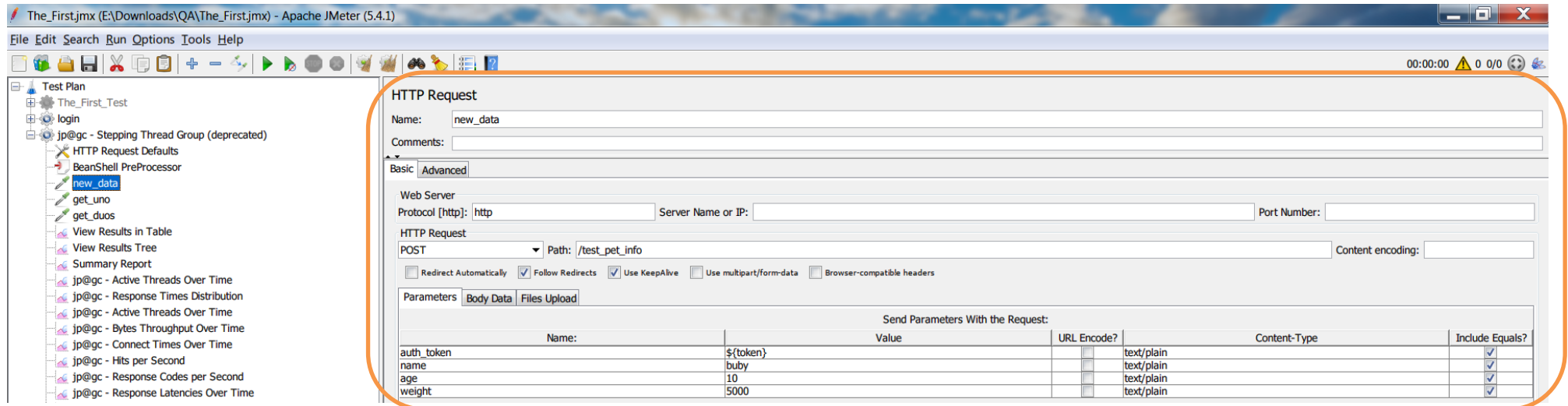


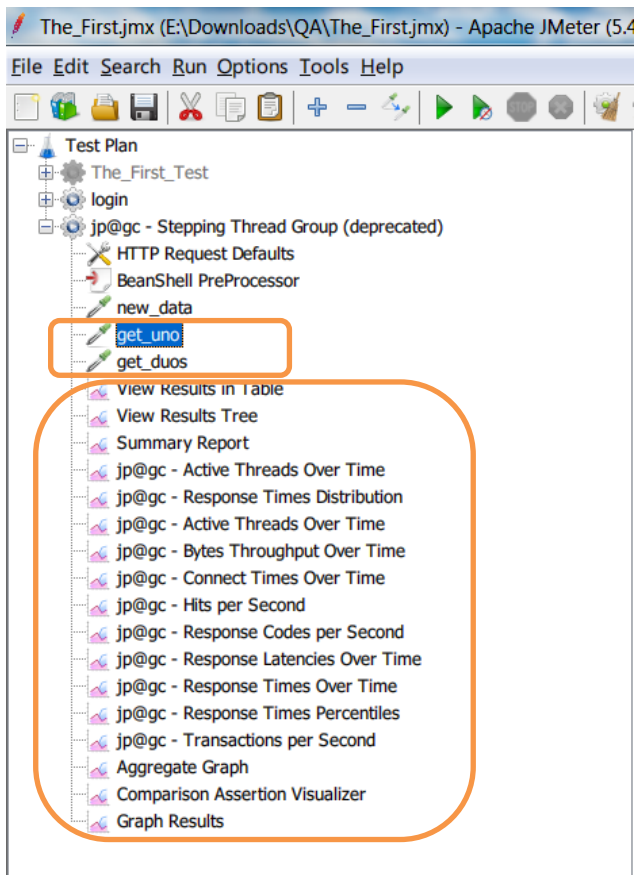
В плагине jp@gc - Stepping Thread Group создаем необходимые нам настройки и запросы:

- HTTP Request Defaults – для ввода данных тестового сервера;
- в BeanShell PreProcessor прописываем скрипт для использования значения полученного ранее токена в последующих запросах;



- создаем POST запрос на сервер, прописываем необходимые параметры;





- создаем еще два простых GET запроса, добавляем группу Listeners по вкусу.

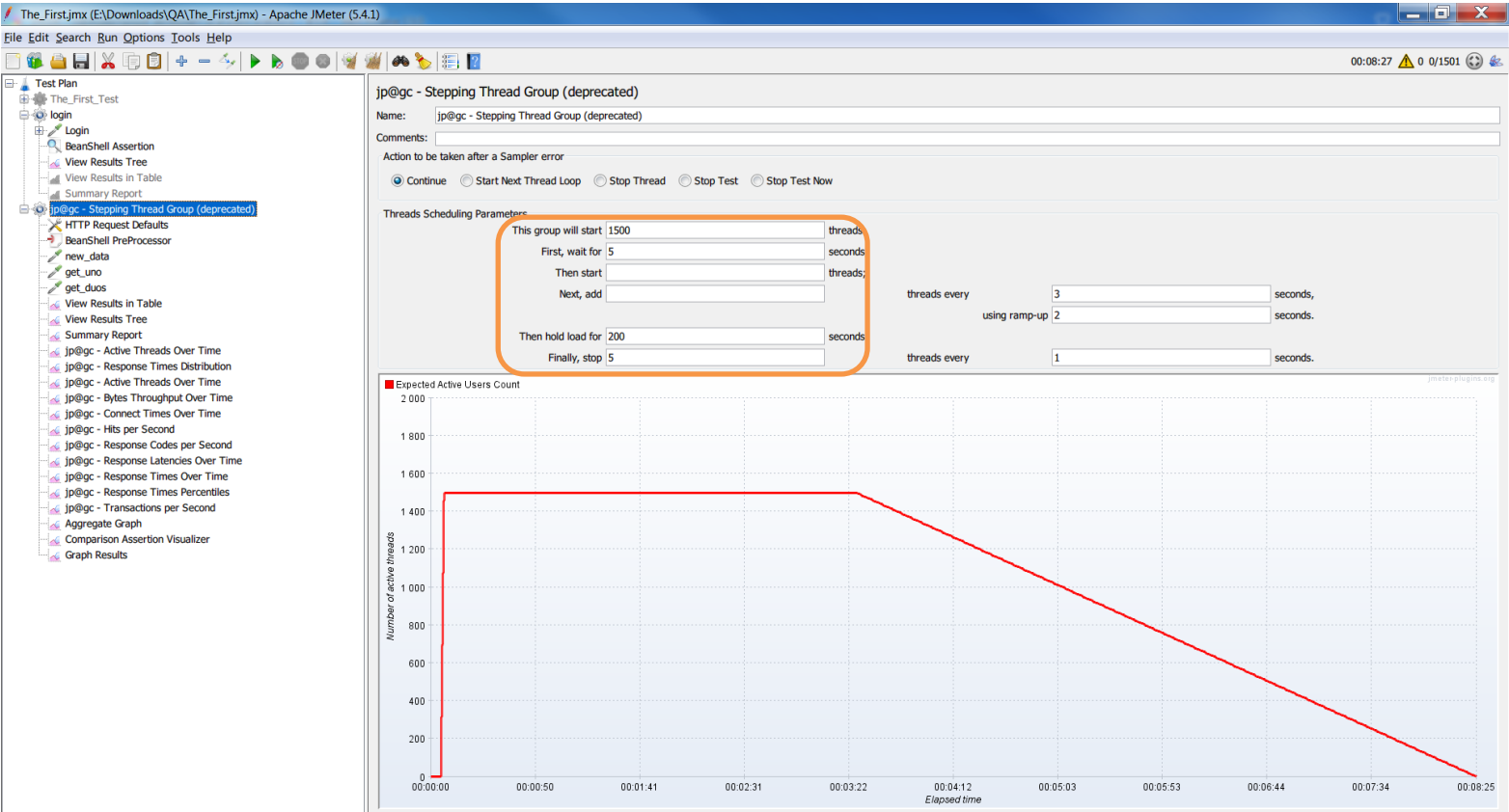
После завершения тестирования анализируем результаты. Ошибок на этом уровне нагрузки практически нет:

The screenshot shows the Apache JMeter 5.4.1 interface with the Summary Report displayed. The report includes a table with the following data:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
new_data	50677	1558	116	21038	979.64	0.02%	78.4/sec	17.73	24.43	231.4
get_uno	50534	1565	114	21010	982.91	0.02%	78.2/sec	12.79	11.38	167.4
get_duos	50363	1562	116	21040	970.01	0.01%	78.0/sec	12.89	11.50	169.2
TOTAL	151574	1562	114	21040	977.55	0.01%	234.6/sec	43.40	47.30	189.4

The 'Error %' column is highlighted with an orange box, showing values of 0.02% for 'new_data', 0.02% for 'get_uno', 0.01% for 'get_duos', and 0.01% for 'TOTAL'.

Проведем **Spike testing** – быстрое экстремальное увеличение нагрузки.



При данной нагрузке сервер справляется, количество ошибок невелико.

The screenshot shows the 'Summary Report' in Apache JMeter 5.4.1. The table displays performance metrics for various test elements. The 'Error %' column is highlighted with an orange box, showing values for 'new_data', 'get_uno', 'get_duos', and 'TOTAL'.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
new_data	34168	5291	117	108030	4493.61	0.96%	68.2/sec	16.96	21.23	254.7
get_uno	33630	5201	121	96980	4269.75	0.94%	67.1/sec	12.52	9.68	190.9
get_duos	33141	5179	122	53272	4220.09	0.86%	66.6/sec	12.40	9.73	190.8
TOTAL	100939	5224	117	108030	4331.14	0.92%	201.4/sec	41.78	40.56	212.5

При увеличении нагрузки:

The_Firstjmx (E:\Downloads\QA\The_Firstjmx) - Apache JMeter (5.4.1)

File Edit Search Run Options Tools Help

00:08:27 0 0/1501

Test Plan

- The_First_Test
 - login
 - Login
 - BeanShell Assertion
 - View Results Tree
 - View Results in Table
 - Summary Report
 - jp@gc - Stepping Thread Group (deprecated)
 - HTTP Request Defaults
 - BeanShell PreProcessor
 - new_data
 - get_uno
 - get_duos
 - View Results in Table
 - View Results Tree
 - Summary Report
 - jp@gc - Active Threads Over Time
 - jp@gc - Response Times Distribution

jp@gc - Stepping Thread Group (deprecated)

Name: jp@gc - Stepping Thread Group (deprecated)

Comments:

Action to be taken after a Sampler error

☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

Threads Scheduling Parameters

This group will start 1900 threads:

First, wait for 5 seconds;

Then start threads;

Next, add threads every 3 seconds,

using ramp-up 2 seconds.

Then hold load for 200 seconds.

Finally, stop threads every 1 seconds.

Сервер справился, но количество ошибок уже приближается к 3 %:

The_Firstjmx (E:\Downloads\QA\The_Firstjmx) - Apache JMeter (5.4.1)

File Edit Search Run Options Tools Help

00:09:51 0 0/1901

Test Plan

- The_First_Test
 - login
 - Login
 - BeanShell Assertion
 - View Results Tree
 - View Results in Table
 - Summary Report
 - jp@gc - Stepping Thread Group (deprecated)
 - HTTP Request Defaults
 - BeanShell PreProcessor
 - new_data

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
new_data	65508	6584	117	156228	5592.51	2.51%	53.0/sec	15.16	16.24	293.1
get_uno	64348	6474	118	156168	5384.24	2.16%	52.1/sec	11.27	7.41	221.8
get_duos	63210	6431	121	156966	5401.55	2.03%	51.2/sec	11.02	7.39	220.5
TOTAL	193066	6497	117	156966	5461.78	2.23%	155.9/sec	37.38	30.98	245.6

Еще одно повышение нагрузки:

The_First.jmx (E:\Downloads\QA\The_First.jmx) - Apache JMeter (5.4.1)

File Edit Search Run Options Tools Help

00:03:17 0 0/3001

Test Plan

- The_First_Test
 - login
 - Login
 - BeanShell Assertion
 - View Results Tree
 - View Results in Table
 - Summary Report
 - jp@gc - Stepping Thread Group (deprecated)
 - HTTP Request Defaults
 - BeanShell PreProcessor
 - new_data
 - get_uno
 - get_duos
 - View Results in Table
 - View Results Tree
 - Summary Report
 - jp@gc - Active Threads Over Time
 - jp@gc - Response Times Distribution

jp@gc - Stepping Thread Group (deprecated)

Name: jp@gc - Stepping Thread Group (deprecated)

Comments:

Action to be taken after a Sampler error

☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

Threads Scheduling Parameters

This group will start 3000 threads;

First, wait for 5 seconds;

Then start threads;

Next, add threads every 3 seconds, using ramp-up 2 seconds.

Then hold load for 200 seconds.

Finally, stop threads every 1 seconds.

Эту нагрузку сервер уже не держит:

The_First.jmx (E:\Downloads\QA\The_First.jmx) - Apache JMeter (5.4.1)

File Edit Search Run Options Tools Help

00:03:17 0 0/3001

Test Plan

- The_First_Test
 - login
 - Login
 - BeanShell Assertion
 - View Results Tree
 - View Results in Table
 - Summary Report
 - jp@gc - Stepping Thread Group (deprecated)
 - HTTP Request Defaults
 - BeanShell PreProcessor
 - new_data

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

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

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
new_data	15654	13218	21	177758	15505.60	15.54%	81.4/sec	47.89	21.63	602.1
get_uno	14660	12543	185	182307	13513.93	15.95%	76.6/sec	41.74	9.37	557.7
get_duos	13623	12584	6	180424	13526.86	15.74%	71.7/sec	38.78	8.91	553.9
TOTAL	43937	12796	6	182307	14262.74	15.74%	228.6/sec	127.77	39.77	572.3