

Лабораторная работа «Тестирование с помощью JMeter»

Цель работы: получить практические навыки тестирования сервисов с помощью инструмента JMeter.

Теоретическая часть

Приложение Apache JMeter™ — это программное обеспечение с открытым исходным кодом, предназначенное для загрузки тестов функционального поведения и измерения производительности.

На рисунке 1 представлена стартовая страница JMeter.

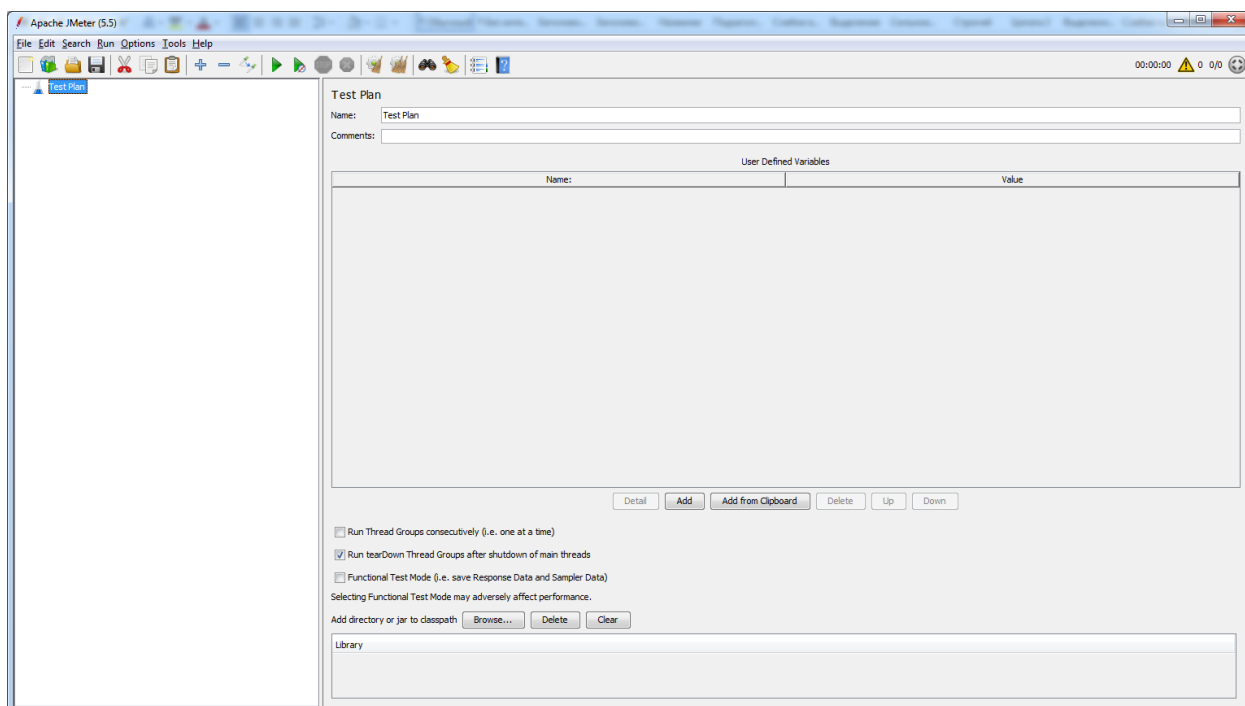


Рисунок 1 – Стартовая страница JMeter

JMeter предоставляет возможность загрузки и тестирования производительности множества различных типов приложений/серверов/протоколов:

- Интернет — HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET, ...);
- Веб-сервисы SOAP/REST;
- FTP;
- База данных через JDBC;
- LDAP;
- Промежуточное программное обеспечение, ориентированное на сообщения (MOM) через JMS;

- Почта — SMTP(S), POP3(S) и IMAP(S);
- Собственные команды или сценарии оболочки;
- TCP;
- Java-объекты.

Преимущества JMeter:

- open source, бесплатный;
- простой, понятный интерфейс;
- наличие шаблонов;
- большое количество плагинов;
- удобная работа с многопоточностью;
- анализ и визуализация данных.

1. Пример тестирования веб-страницы

Для тестирования веб-сайте будет разработан базовый план тестирования. В качестве объекта тестирования используются 2 страницы веб-сайта JMeter. Параметрами тестирования являются:

- количество пользователей – 10;
- количество повторений запросов – 5.

Итак, общее количество запросов (10 пользователей) x (2 запроса) x (повторить 5 раза) = 100 HTTP-запросов.

Первым шагом при построении плана тестирования является создание группы потоков (ThreadGroup). Группа потоков сообщает JMeter количество пользователей, которые будут имитироваться при выполнении теста, как часто пользователи должны отправлять запросы и сколько запросов они должны отправлять.

Для добавления группы потоков необходимо произвести действия: щелкнуть правой кнопкой мыши для открытия меню Add, а затем выбрать Add → ThreadGroup (рисунок 2).

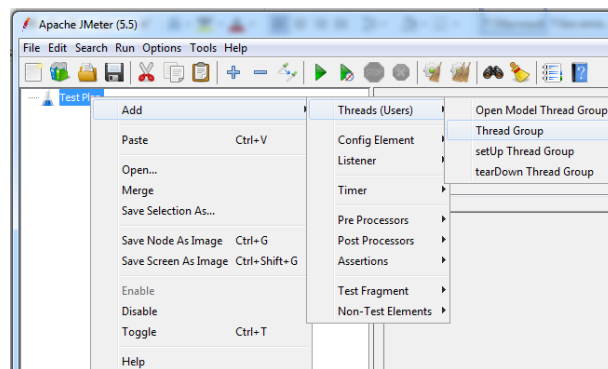


Рисунок 2 – Добавление ThreadGroup

На рисунке 3 представлена настройка пользователей: количество пользователей «Number Of Thread», период нарастания «Ramp-up period» - количество секунд, необходимых для увеличения потоков до установленного, количество повторов запросов «Loop Count».

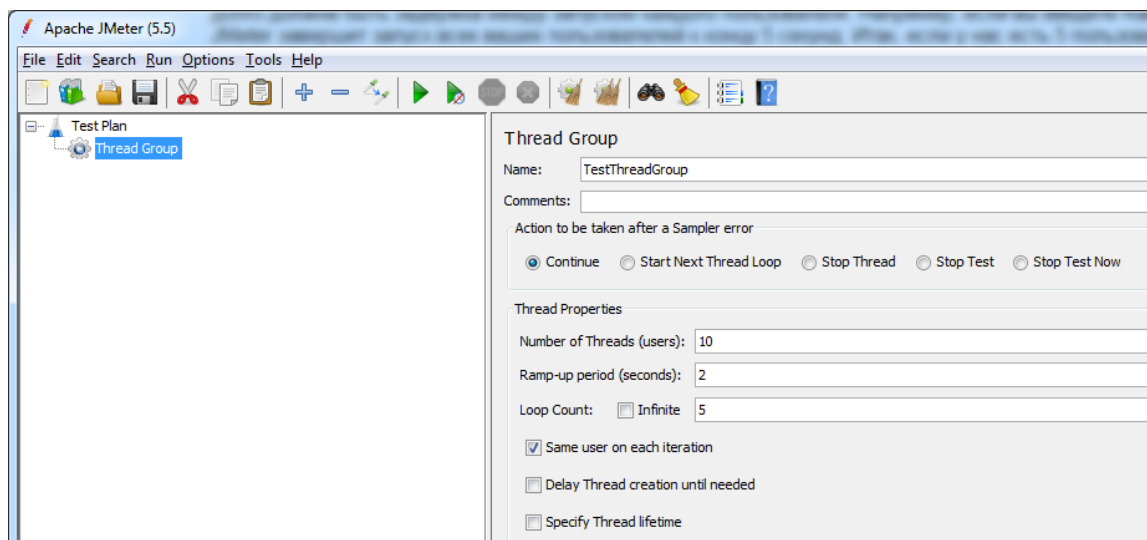


Рисунок 3 – Настройка параметров ThreadGroup

Вторым шагом необходимо определить какие задачи будут выполнять описанные пользователи.

Сначала установим параметры запроса по умолчанию. Для добавления объекта с параметрами необходимо открыть контекстное меню ThreadGroup, выбрать «Add» → «Config Elements» → «HTTP Request Defaults».

На рисунке 4 представлена настройка параметров запроса по умолчанию, где указан в качестве тестируемого сервера «jmeter.apache.org».

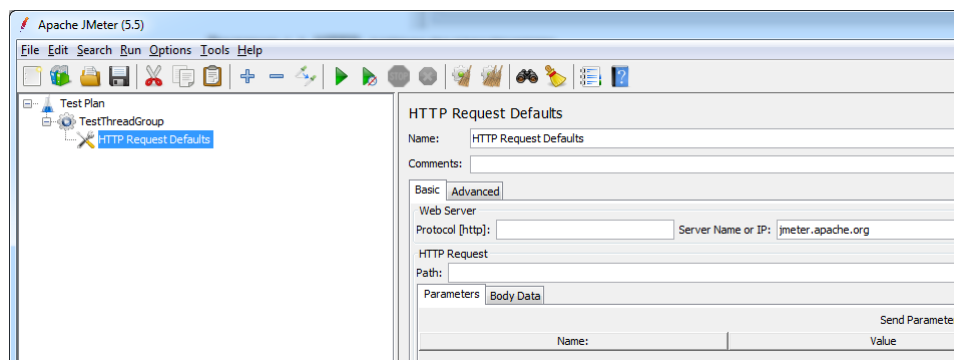


Рисунок 4 – Настройка параметров запроса по умолчанию

Почти все веб-тестирования должны использовать поддержку файлов cookie, если только ваше приложение специально не использует файлы cookie. Для добавления поддержки файлов cookie необходимо добавить объект HTTP Cookie Manager в каждую группу потоков в плане тестирования. Это гарантирует, что каждый поток получит свои собственные файлы cookie, но общие для всех объектов HTTP-запроса.

Следующим этапом производится добавление HTTP-запроса в группу потоков (Add → Sampler → HTTP-Request). Первым тестовым запросом является запрос к стартовой странице JMeter. На рисунке 5 представлена настройка параметров запроса:

— поле Path – «/», так как имя сервера уже указано в значениях по умолчанию.

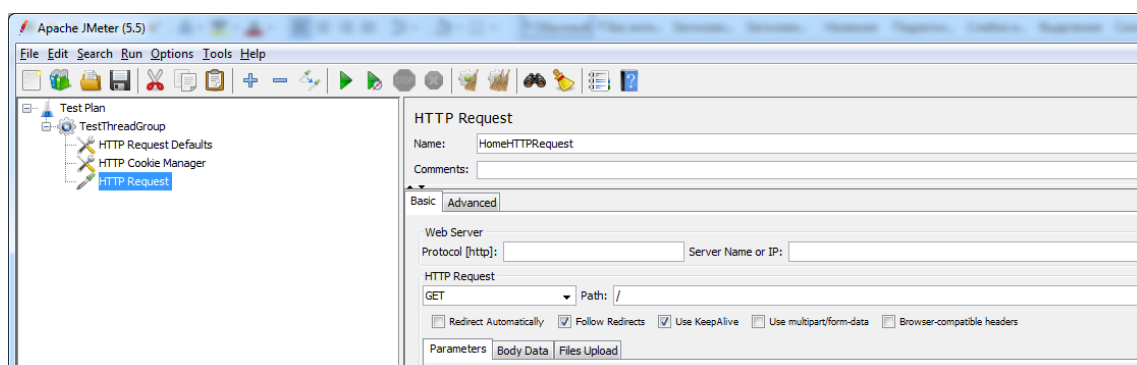


Рисунок 5 – Настройка параметров первого HTTP-запроса

Затем необходимо добавить второй HTTP-запрос и отредактировать следующие параметры (рисунок 6):

— в поле «Path» указать «/changes.html».

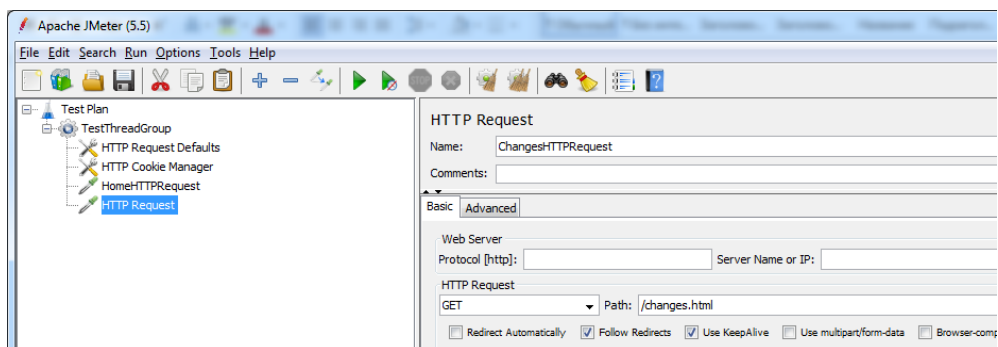


Рисунок 6 – Настройка параметров второго HTTP-запроса

Третьим шагом является добавление в план тестирования объекта «Listener», необходимый для сохранения всех результатов HTTP-запросов в файле и представление визуальной модели данных.

Для добавление объекта Listener необходимо в контекстном меню группы потоков выбрать Add → Listener → Graph Results.

Результат выполнения построенного плана тестирования приведено на рисунке 7. Элемент Listener предоставляет информацию о следующих параметрах:

- No of Samples – номер выборки;
- Latest Sample (черный цвет) – текущая выборка. указанная в миллисекундах;
- Average (синий цвет) – текущее среднее значение по выборкам, указанная в миллисекундах;
- Deviation (красный цвет) – стандартное отклонение времени отклика, указанная в миллисекунды;
- Throughput (зеленый цвет) – текущая пропускная способность, указанная в Кб/сек.

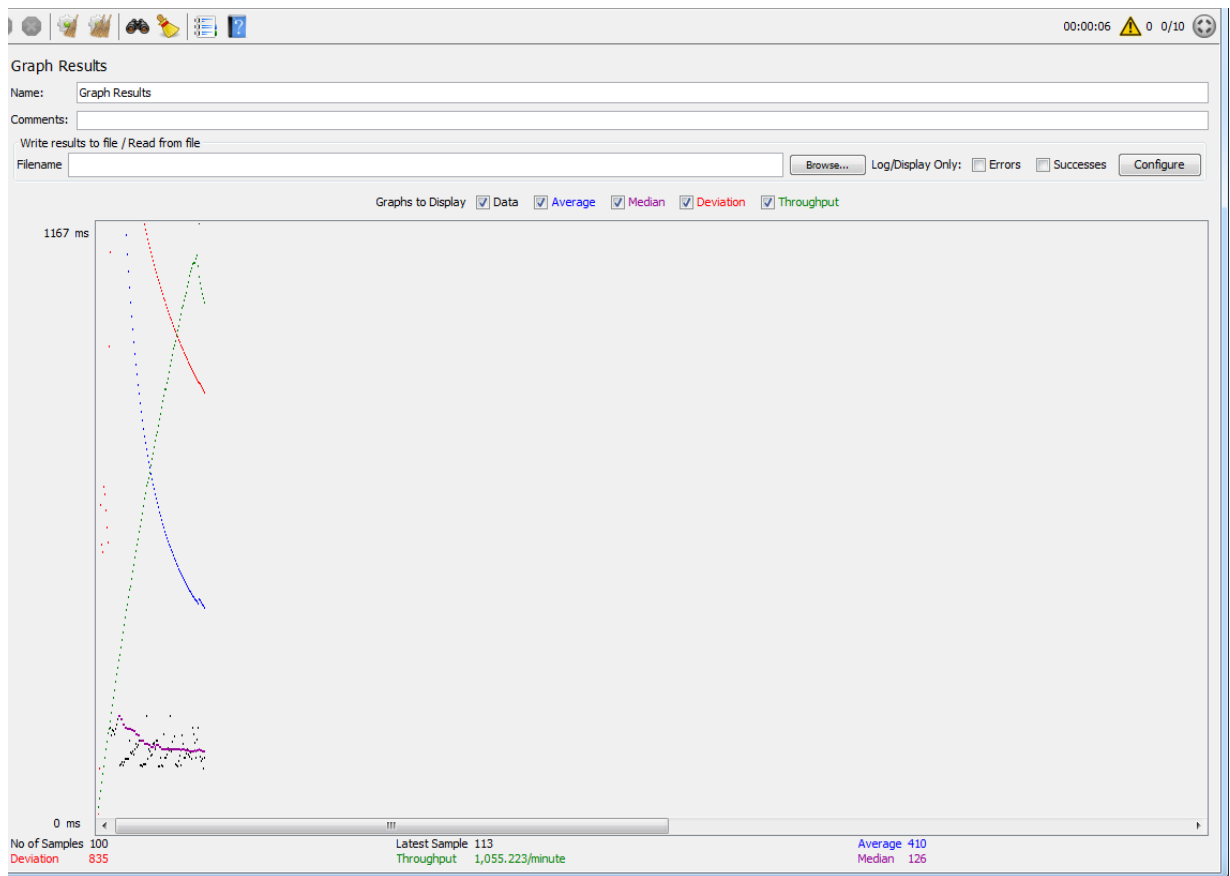


Рисунок 7 – Результат выполнения плана тестирования страниц JMeter

2. Пример тестирования через API

С помощью JMeter возможно выполнить нагрузочное тестирование через API, то есть оценить поведение не API, а сервера: насколько быстро производится обработка запросы, сколько запросов может обработать в единицу времени и так далее.

Пример тестирования будет производиться на бесплатном сервисе, предоставляющем возможности регистрации и аутентификации пользователя:

<https://www.appsloveworld.com/sample-rest-api-url-for-testing-with-authentication>

Первым шагом необходимо создать переменную тест плана «base_url», содержащую основной URL: <http://restapi.adequateshop.com> (рисунок 8).

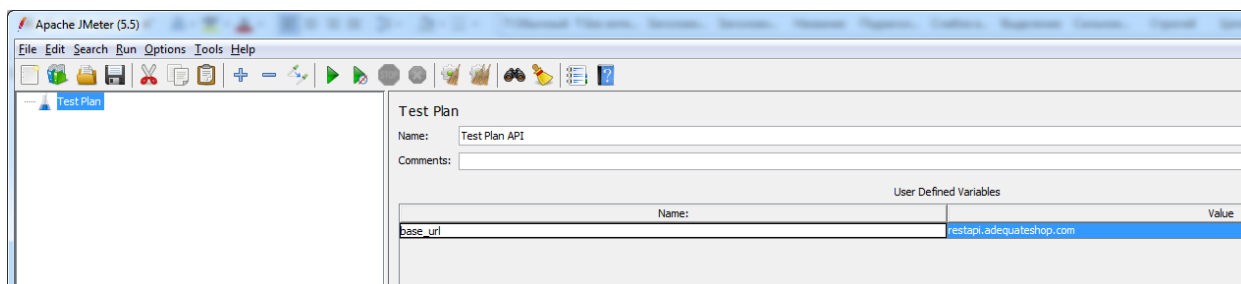


Рисунок 8 – Настройка плана с указанием глобальной переменной

Первым методом, который будет протестирован, является регистрация пользователей «User Registration API».

Для этого необходимо создать элемент Группа потоков и HTTP-запрос, в котором устанавливается метод «POST» и указывается путь «/api/authaccount/registration» в соответствии с описанием API. В качестве названия сервера указывается переменная «\${base_url}». Вместе с запросом будут отправлены данные, которые указываются на вкладке «body data» и в соответствии с описанием API представлены в формате JSON. Настройка HTTP-запроса представлена на рисунке 9.

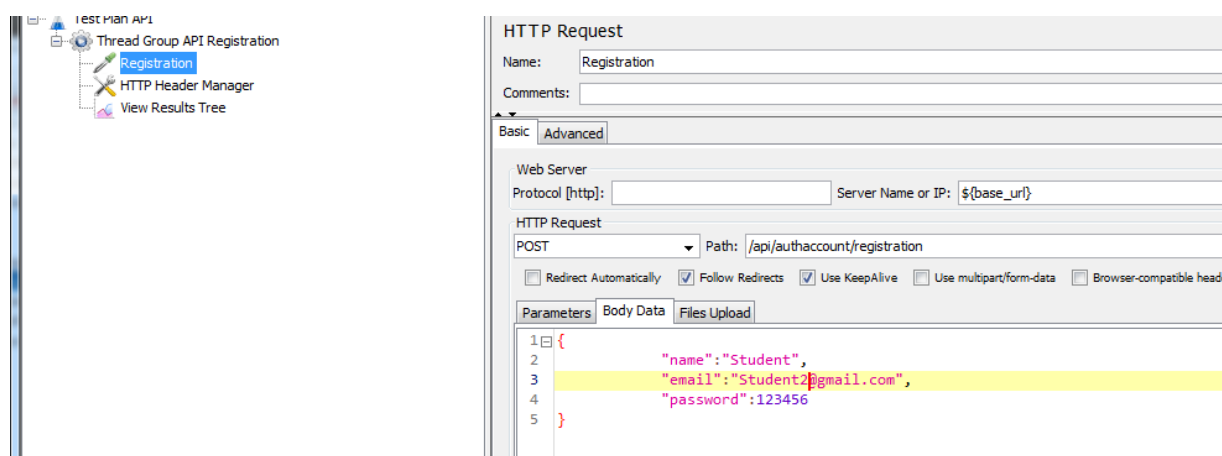


Рисунок 9 – Настройка HTTP-запроса для регистрации

Использование формата данных JSON указывается в заголовке (элемент HTTP Header Manager): Content-Type со значением «application/json» (рисунок 10).

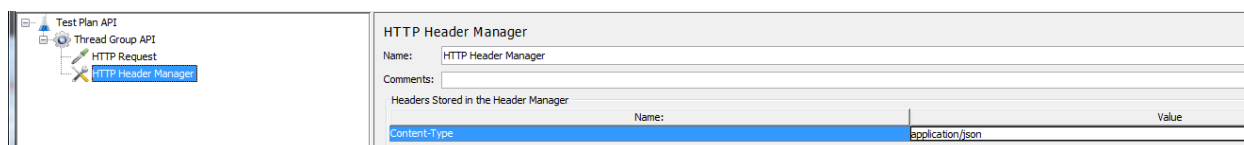


Рисунок 10 – Установка параметров формата содержимого файла JSON

Для просмотра результатов выполнения запроса необходимо добавить элемент Listeners → View Results Tree.

Результат выполнения запроса на регистрацию представлен на рисунке 11. В ответ на запрос были возвращены некоторые данные также в формате JSON.

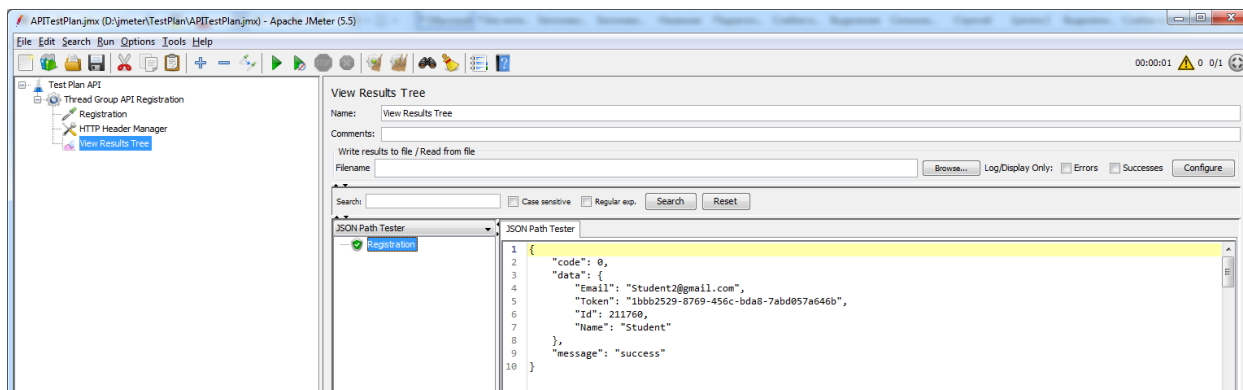


Рисунок 11 – Результат выполнения запроса регистрации

Вторым методом, который будет протестирован, является аутентификация пользователей «User Login API». За основу построения теста будет взята копия Группы потоков регистрации.

В HTTP-запросе необходимо изменить путь и отправляемые данные (рисунок 12).

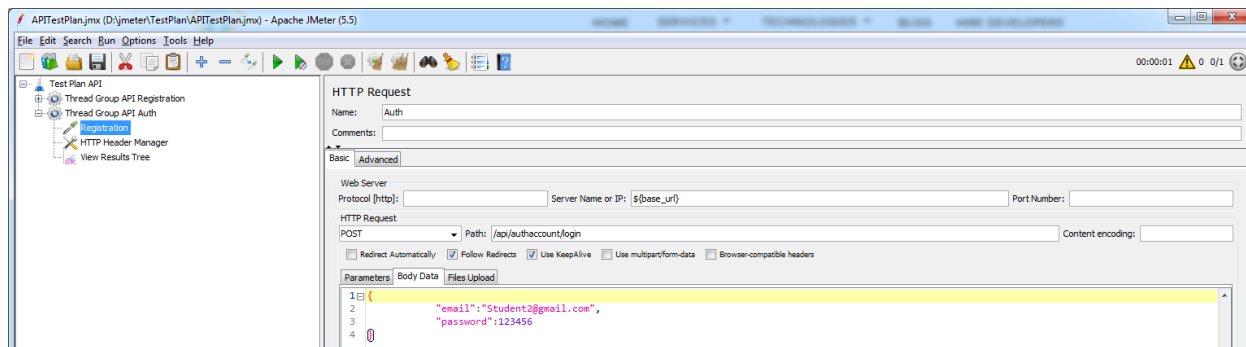


Рисунок 12 – Настройка HTTP-запроса для аутентификации

Для выделения необходимых данных из ответа сервера используется JSON Extractor (Add → Post Processors → JSON Extractor). В добавленном элементе указывается имя переменной и путь, по которому будет производиться извлечение данных.

Для выполнения запроса аутентификации необходимо выделить объект Token, который находится в объекте Data, записываемый в корне. Поэтому для выделения Токена будет прописан путь: \$.data.Token (рисунок 13).

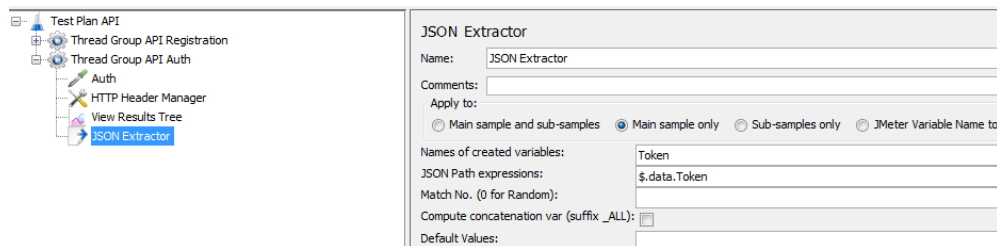


Рисунок 13 – Добавление ссылки на Токен

Настройка доступа к переменной Токен для всего тест плана осуществляется с помощью элемента BeanShell Assertion (Add → Assertions → BeanShell Assertion), в котором прописывается скрипт для объявления глобальной переменной из локальной (рисунок 14):

```
${__setProperty(GlobalToken, ${Token})};
```

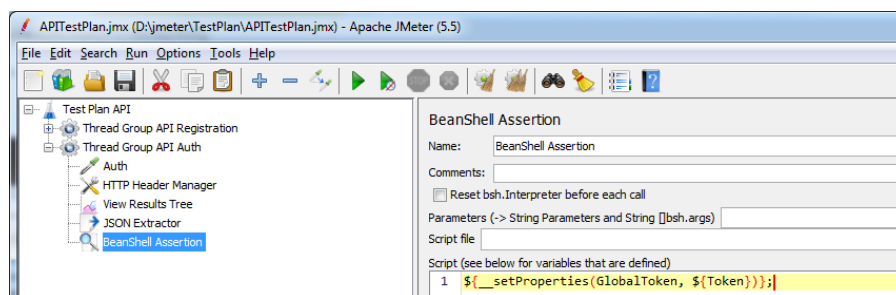


Рисунок 14 – Создание глобальной переменной с помощью BeanShell Assertion

Проверка успешности запроса авторизации может производиться с помощью элемента Response Assertion, то есть проверка ответа. В нем можно указать, что ответ должен содержать определенное значение поля, например: "code":0 (рисунок 15).

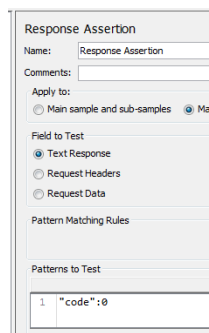


Рисунок 15 – Response Assertion

С целью проверки реакции сервера на неверные данные необходимо изменить данные запроса, например, пароль на некорректное значение и запустить выполнение группы. Результат выполнения теста представлен на рисунке 16, где представлен ответ сервера, рисунок 17 демонстрирует результат выполнения элемента Response Assertion с сообщением об ожидании кода 0 в ответе сервера.

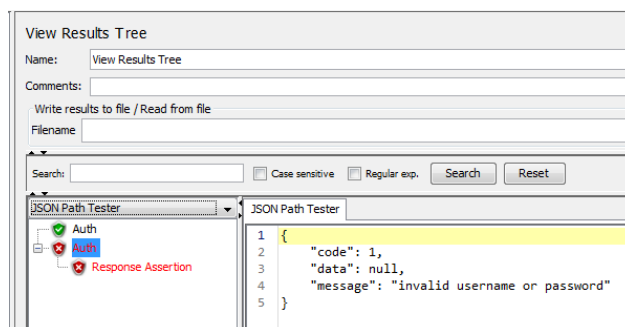


Рисунок 16 – Ответ сервера на запрос аутентификации

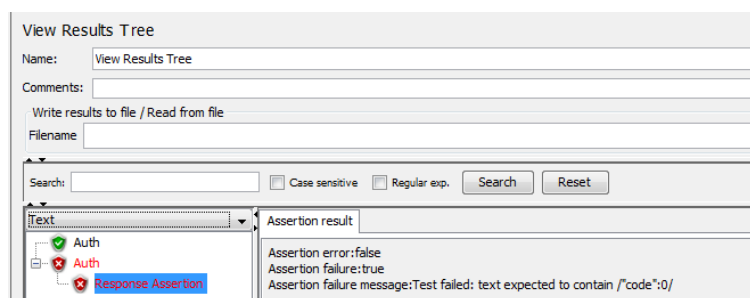


Рисунок 17 – Результат выполнения Response Assertion

Третьим методом, который будет протестирован, является получение информации о пользователях «Get All Users».

В новой группе потоков добавляем HTTP-запрос с методом GET, именем сервера «base_url» и путем: /api/users?page=1. Параметр page в пути запроса можно указать в специальной вкладке в настройках HTTP-запроса (рисунок 18).

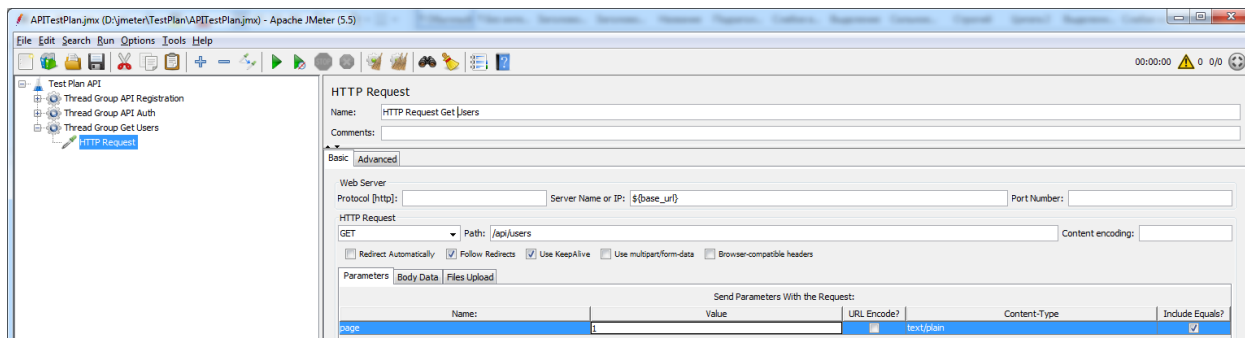


Рисунок 18 – Запрос получения всех пользователей

Для получения всех пользователей необходимо так же указать Токен, для этого добавляется элемент «HTTP Header Manager», в котором устанавливается параметр:

Authorization – Bearer \${__property(GlobalToken)}

Bearer – тип токена.

Теперь необходимо добавить элемент View Results Tree и запустить тест на выполнение (рисунок 19).

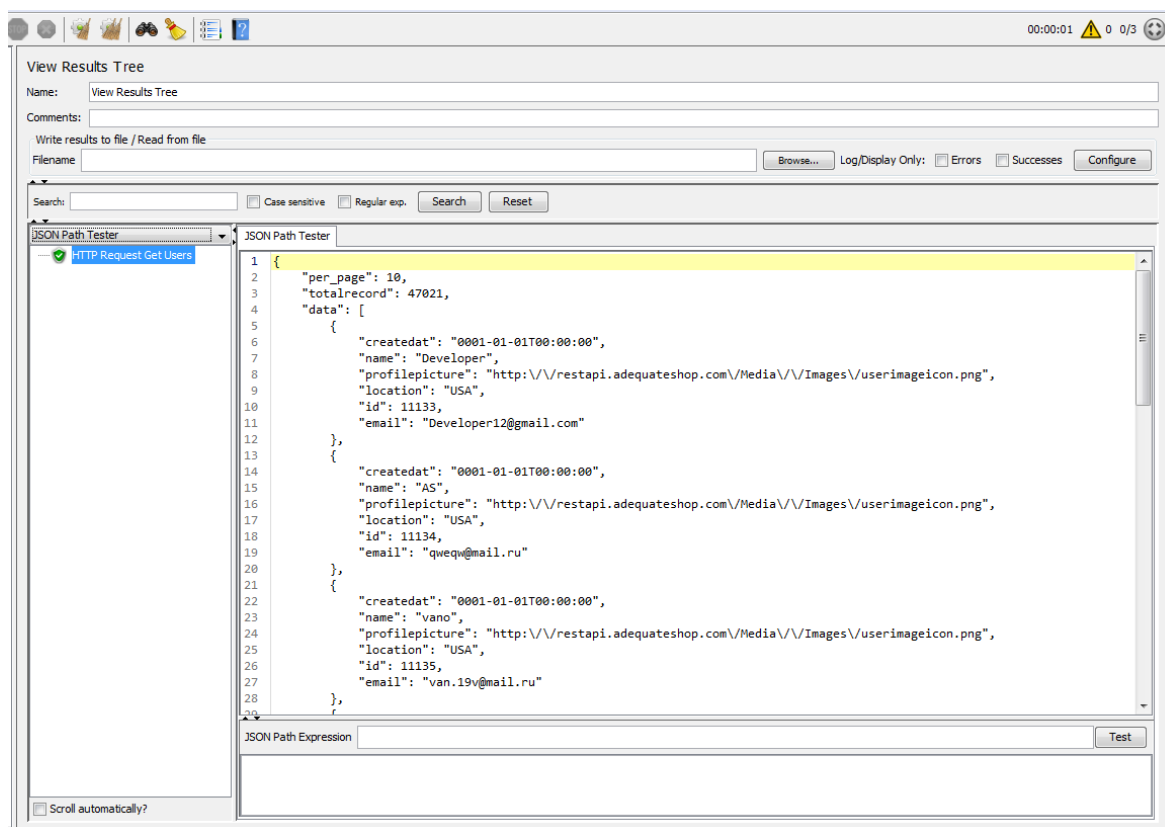
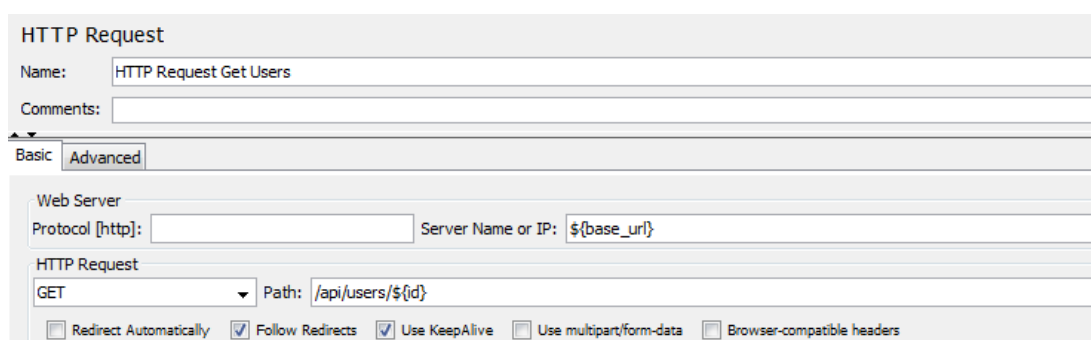


Рисунок 19 – Результат выполнения запроса информации о пользователях

Четвертым методом, который будет протестирован, является получение информации о пользователе «Get user by Id». За основу построения теста будет взята копия Группы потоков Get Users.

Путь данного сервиса содержит параметр «id», который установлен параметром или с использованием функции «CSV Data Reader» (Add → Config Element → CSV Data Set Config).

При использовании файла со значениями параметров в настройках запроса поля «Path» указывается наименование переменной, в которую будут подставлены соответствующие значения при выполнении теста (рисунок 20).



HTTP Request

Name: HTTP Request Get Users

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: \${base_url}

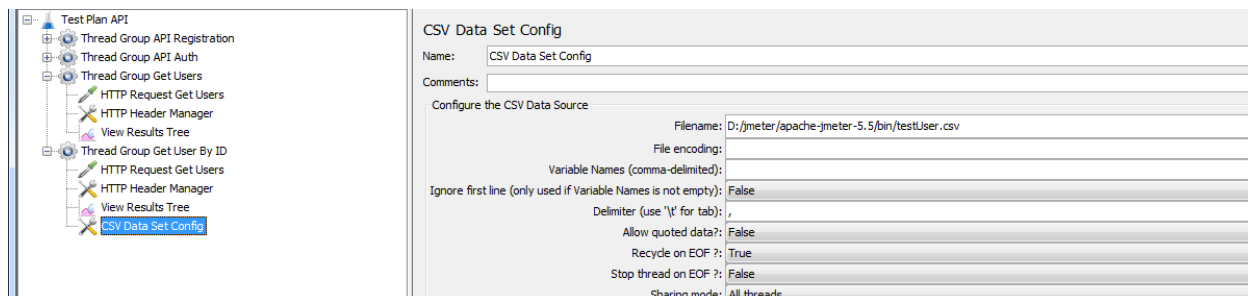
HTTP Request

GET Path: /api/users/\${id}

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Рисунок 20 – Настройка HTTP-Request «Get user by Id»

В элементе «CSV Data Set Config» указывается путь к файлу с расширением «.csv», в котором хранятся данные для тестирования (например, id для тестирования «Get user by Id»). В файле csv в первой строке обязательно указывается наименование переменной, значения которой будут использованы, далее в столбце заполняются сами значения. На рисунке 21 представлен элемент с указанием пути к файлу.



Test Plan API

- Thread Group API Registration
- Thread Group API Auth
- Thread Group Get Users
 - HTTP Request Get Users
 - HTTP Header Manager
 - View Results Tree
- Thread Group Get User By ID
 - HTTP Request Get Users
 - HTTP Header Manager
 - View Results Tree
- CSV Data Set Config

CSV Data Set Config

Name: CSV Data Set Config

Comments:

Configure the CSV Data Source

Filename: D:/jmeter/apache-jmeter-5.5/bin/testUser.csv

File encoding:

Variable Names (comma-delimited):

Ignore first line (only used if Variable Names is not empty): ☒

Delimiter (use '\t' for tab): ,

Allow quoted data?: ☒

Recycle on EOF?: ☒

Stop thread on EOF?: ☒

Sharing mode: All threads

Рисунок 21 – Настройка элемента «CSV Data Set Config»

Выполнение теста для каждой записи в csv файле зависит от количества повторений, установленных в настройках Группы потоков. Значение пользователей должно быть равно количеству записей в файле. Результат

выполнения тестирования представлен на рисунке 22. Запрос был произведен для 4-х id пользователей.

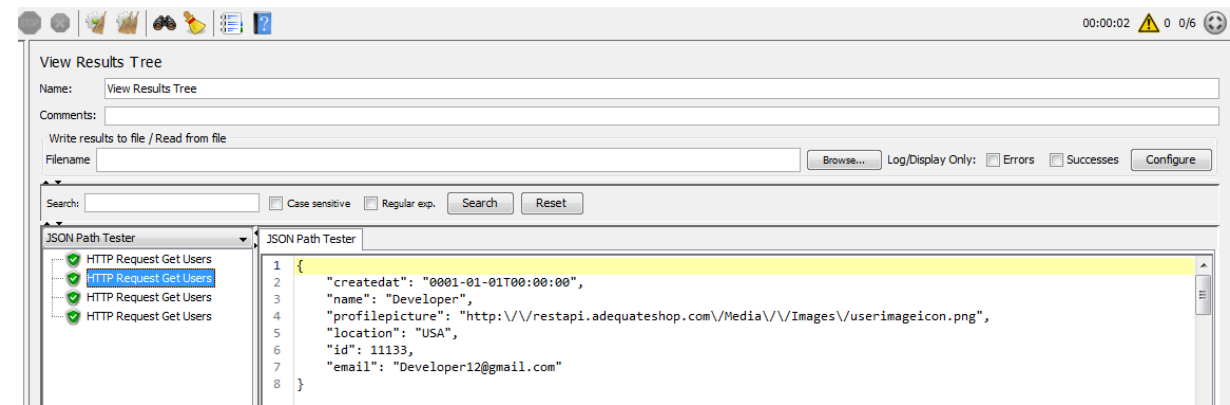


Рисунок 22 – Результат запроса пользователя по id

Общий отчет по тест плану можно отобразить с помощью элемента «Summary Report» (Add → Listener → Summary Report), добавленный в корневой элемент «Тест план». В нем будут отображены все запросы, выполняемые в ходе выполнения всего тест плана. Элемент «Summary Report» позволяет проанализировать такие характеристики запроса, как: время выполнения запроса (среднее, минимальное, максимальное), стандартное отклонение, процент ошибок, пропускная способность, количество Кбайт принятых и отправленных.

Практическая часть

1. Реализуйте тестирование доступа к следующим сервисам, указанным в таблице 1.

Проведите тестирование с разными характеристиками количества пользователей и повторений.

Таблица 1 – Варианты заданий для пункта 1

№ варианта	Ссылка на сервис
1	https://yandex.ru/maps
2	https://www.google.com/calendar
3	https://www.google.com/maps
4	https://translate.google.ru/
5	https://yandex.ru/pogoda

6	https://afisha.yandex.ru/ryazan
7	https://www.bing.com/
8	https://www.bing.com/maps
9	https://drive.google.com/drive
10	https://ru.wikipedia.org/wiki

2. Проведите анализ полученных результатов.

3. Проведите тестирование API.

Тестирование сервиса, рассмотренного в примерах:
<https://www.appsloveworld.com/sample-rest-api-url-for-testing-with-authentication>

3.1 Провести тестирование сервисов регистрации и аутентификации.

3.2 Провести тестирование сервиса «Get All Users» со страницами, указанными в таблице 2.

Таблица 2 – Варианты заданий для пункта 3.3

№ варианта	Номер страницы
1	3
2	4
3	5
4	6
5	7
6	8
7	9
8	10
9	11
10	12

3.3 Провести тестирование сервиса «Get user by Id» для всех пользователей, полученных в результате выполнения пункта 3.2.

4. Добавьте элемент «Summary Report» для тест-плана пункта 3 практического задания и проведите анализ полученных результатов.

Содержание отчета

По результатам выполнения работы оформляется отчет в соответствии с требованиями ГОСТ 7.32-2017 «Отчет о научно исследовательской работе.

Структура и правила оформления», включающий:

- титульный лист;
- цель работы;
- описание хода выполнения работы, содержащий скрипты тест-кейсов и результат их выполнения;
- выводы.

Контрольные вопросы

1. Что такое JMeter?
2. Для каких задач подходит JMeter?
3. Что указывается в группе потоков?
4. За что отвечает параметр Ramp-up period?
5. Какие параметры можно проанализировать с помощью элемента Listeners?
6. Как указываются переменные в JMeter?
7. С помощью каких элементов можно проанализировать выполнение запроса?
8. Как указывается глобальная переменная в JMeter?
9. Где указываются данные при POST запросе?
10. В каких форматах может быть построен запрос?
11. Для чего используется элемент «CSV Data Set Config»?
12. Какой элемент позволяет составить отчет по выполнению запросов?