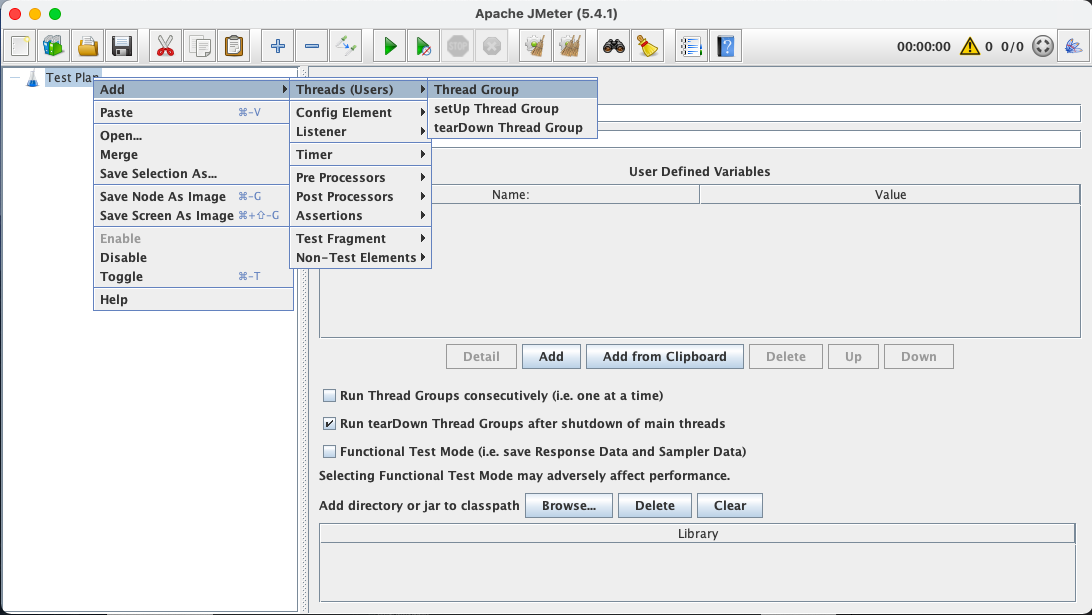
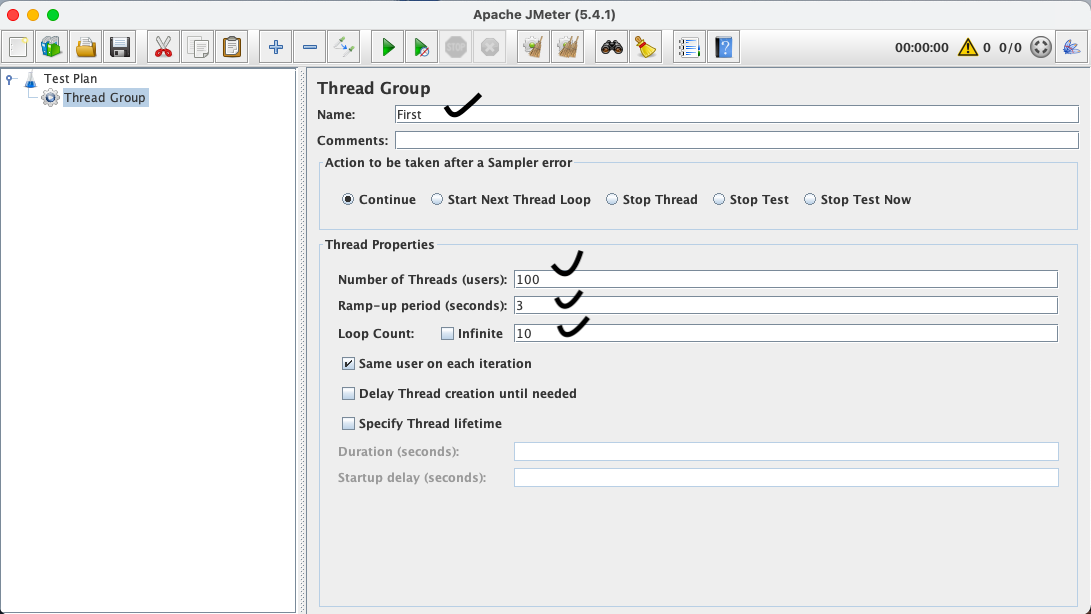
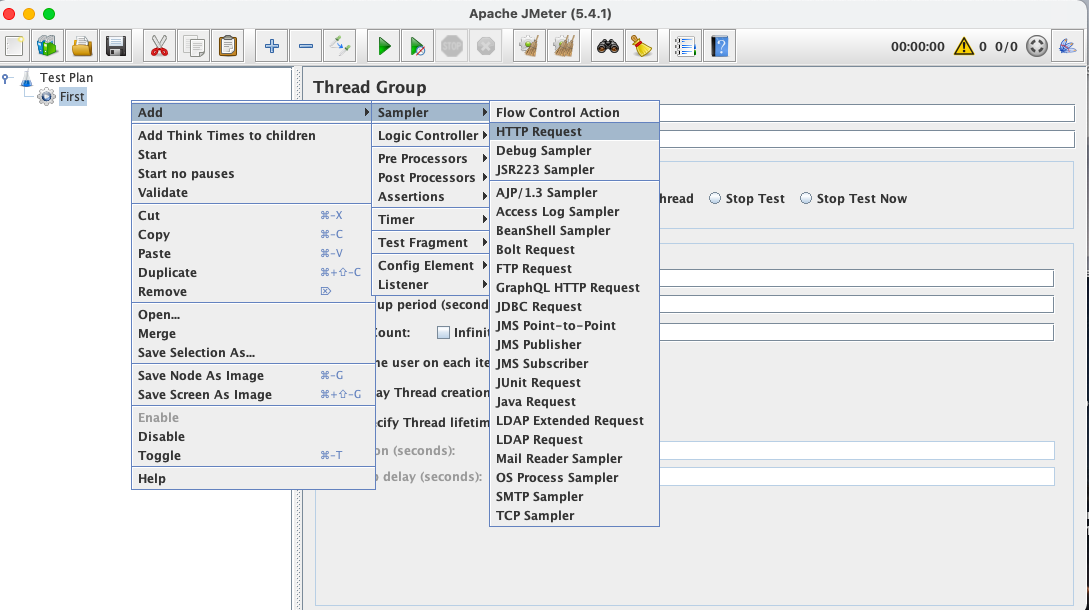
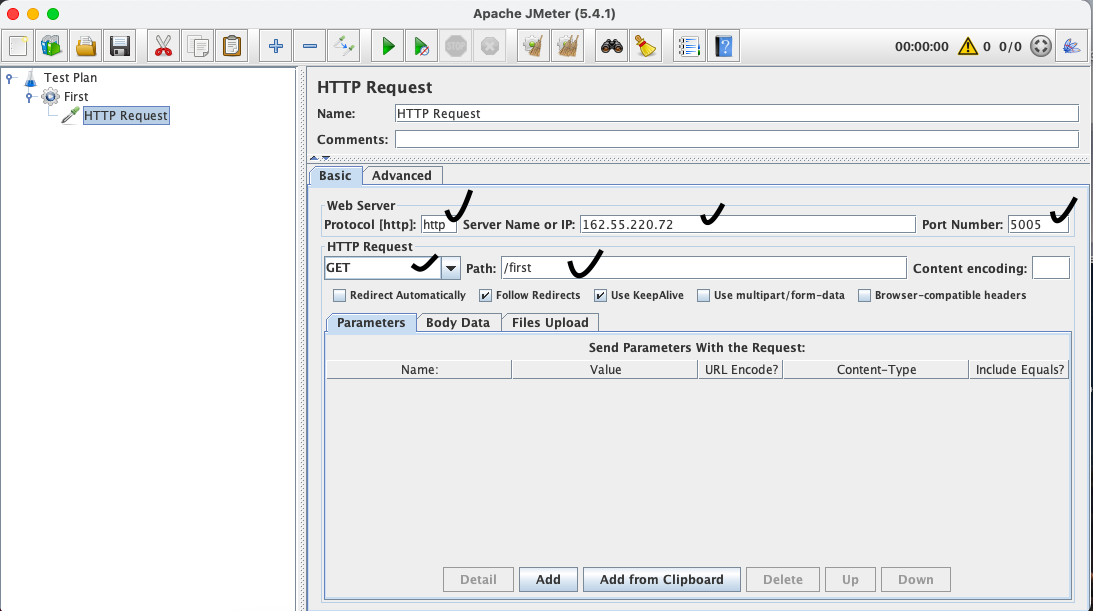
Создадим тестовый сценарий



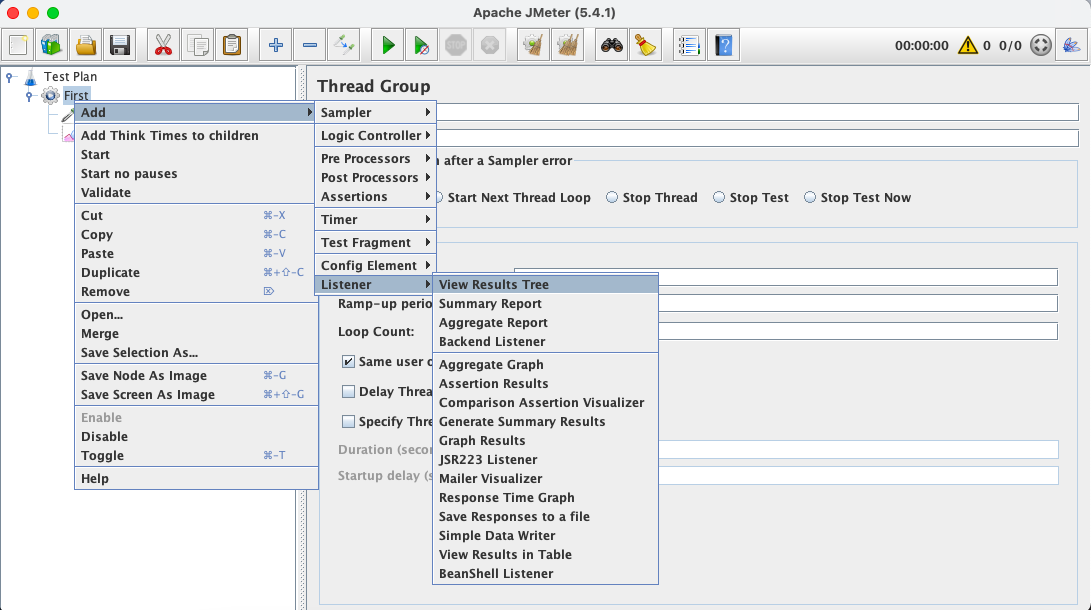
Заполняем

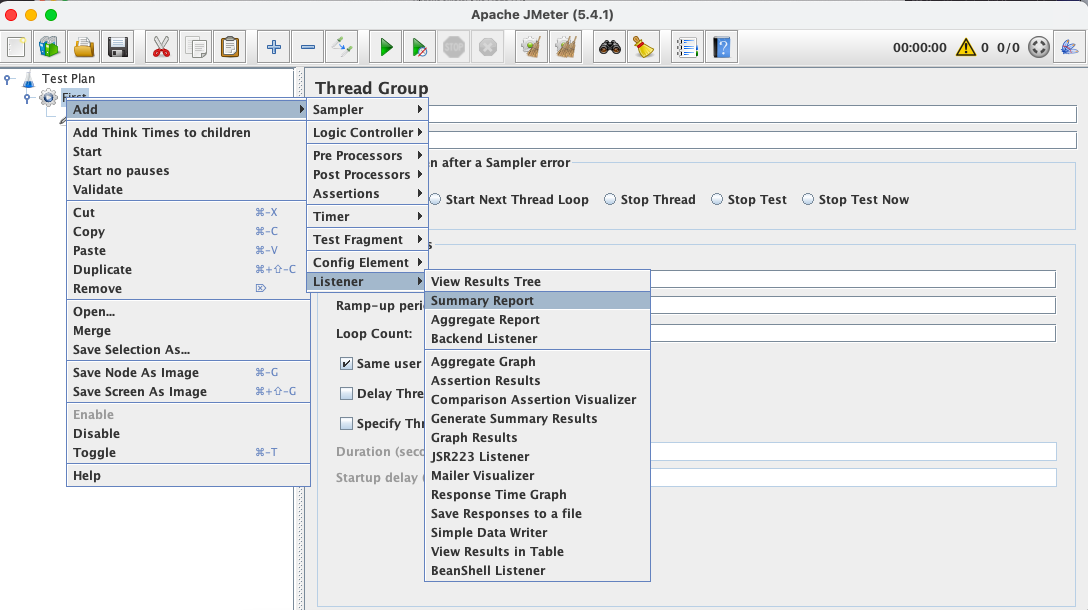
Устанавливаем параметры запроса (http request)



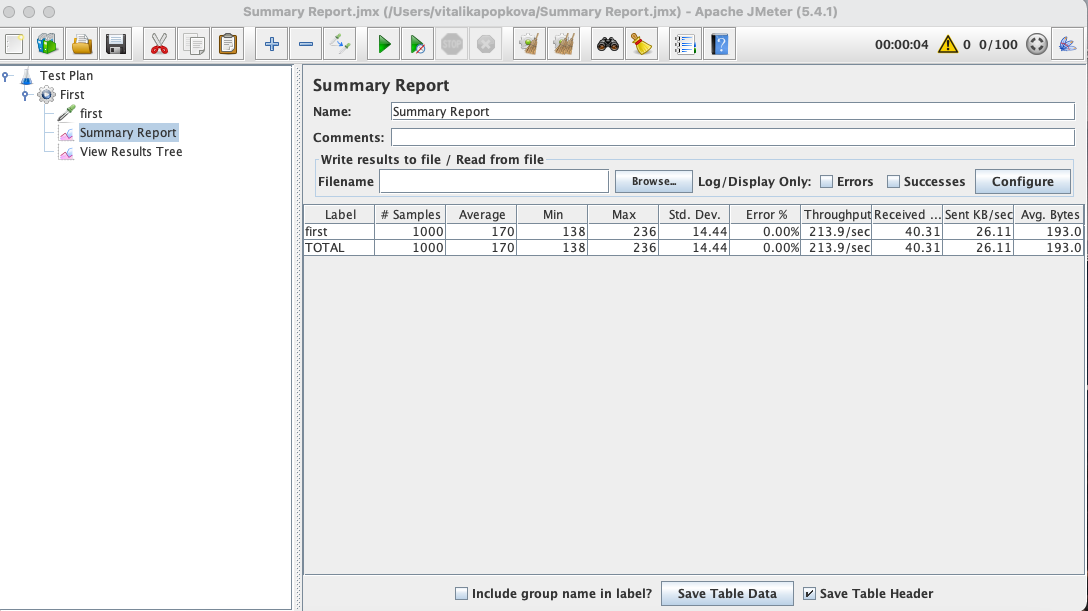
Заполняем

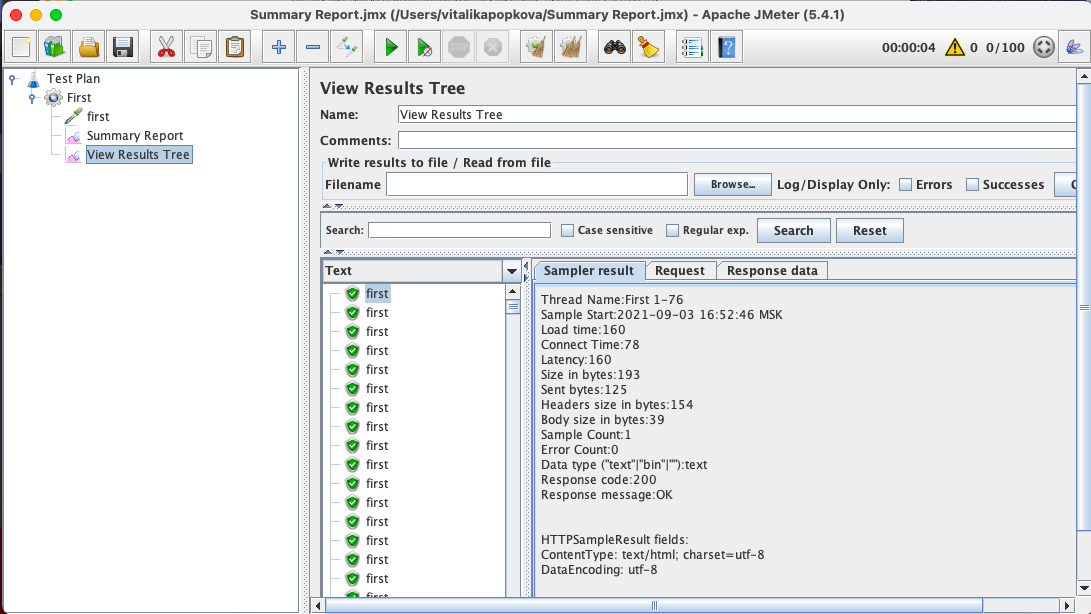
Добавим Listener (слушателей): View Results Tree, Summery Report





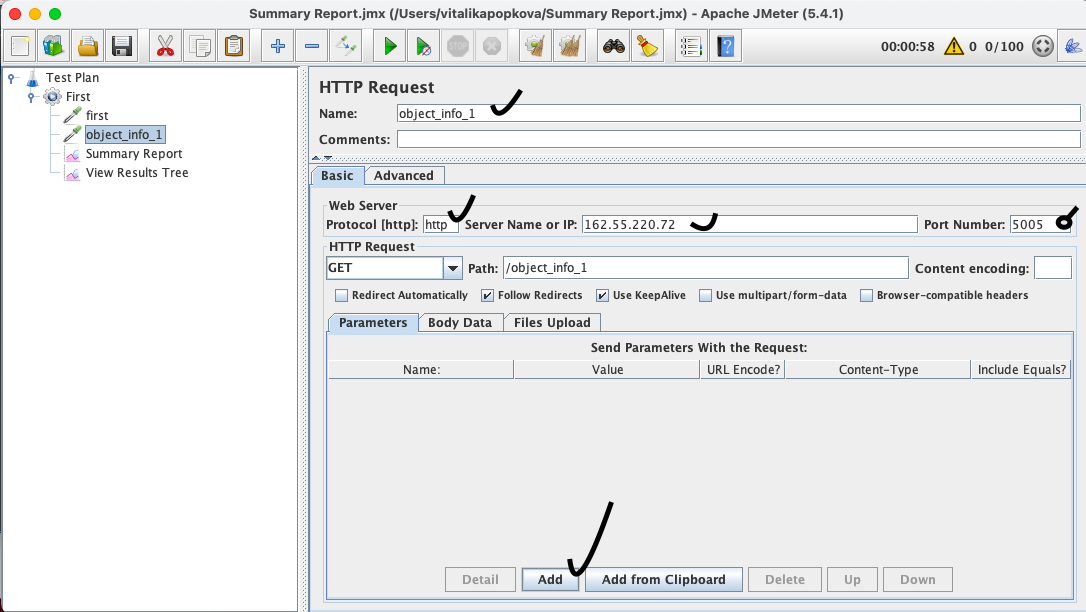
Нажимаем play

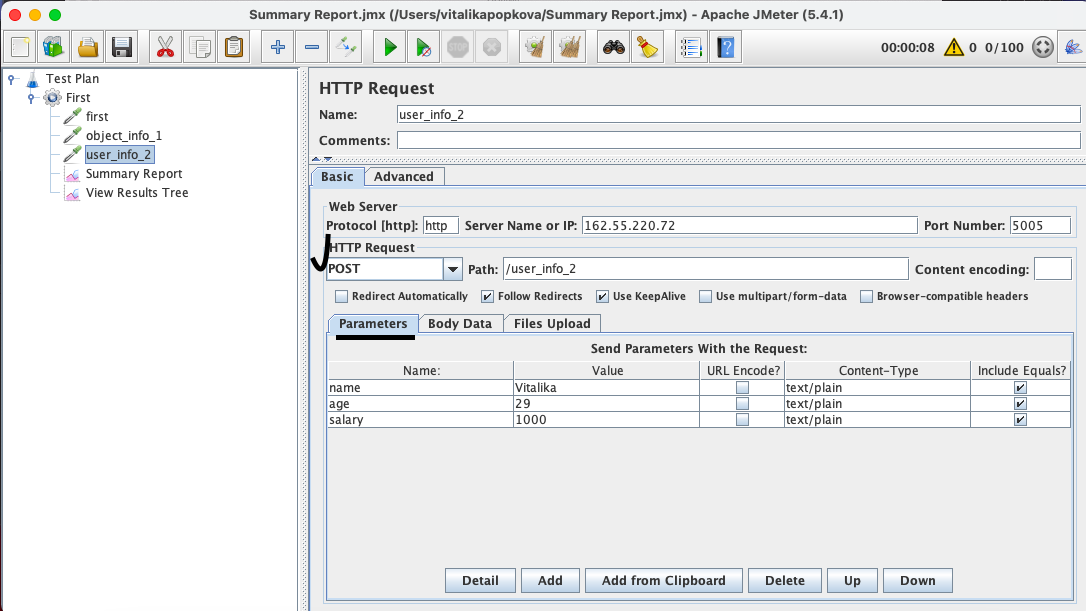


Смотрим результат: количество сделанных запросов, среднее, минимальное, максимальное время ожидания ответа от сервера, % ошибок, пропускная способность (количество запросов в сек, и др)

Добавляем HTTP Request Defaults из данных в Postman (будет заполняться IP и Port автоматически во тест сценарии)

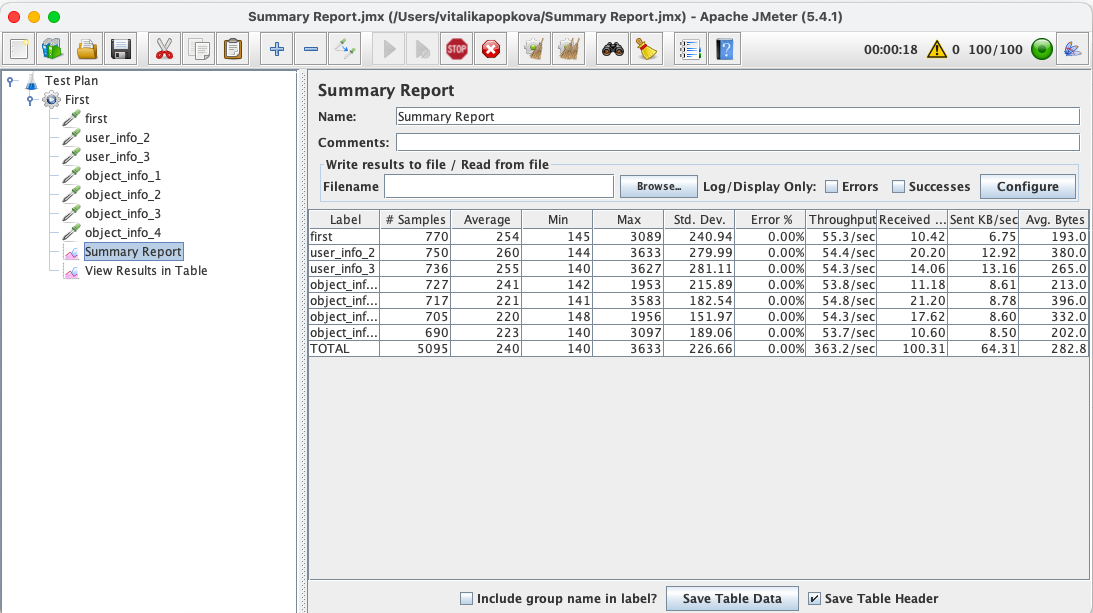
Метод Get



Метод Post

Запросы из Postman

Определение границ производительности с помощью плагина Stepping Thread Group

(Создаем HTTP Request Defaults, заполняем. Затем создаем наш HTTP Request, создаем JSON Extractor со значением $.token и BeanShell Assertion со скриптом, затем выбираем плагин Stepping Thread Group, копируем настройки HTTP Request Defaults, создаем HTTP Request,и создаем скрипт в BeanShell PreProcessor. Настраиваем параметры Stepping Thread Group, выбираем Summary Report  и тд.

Границы производительности определяются как норма и зона стабильной работы, если нет % ошибок, от начала процента ошибок до 3 % критическая зона, где сервер еще работает, но уже пора расширять возможности, а выше 3 процентов можно останавливать тестирование. 