

РАЗРАБОТКА НАГРУЗОЧНЫХ ТЕСТОВ

Сорокина Татьяна Александровна

студент, Московский государственный технологический университет «СТАНКИН», РФ, г. Москва

Процесс разработки нагрузочных тестов включает в себя следующие этапы: создание структуры теста и разработка заглушек, программирование функций, входящих в структуру теста, отладка тестов, настройка «CI/CD» для автоматического запуска тестов, а также фиксация тестов для их дальнейшего использования.

Структура теста (рис.1) включает в себя группу потоков для каждого автоматизируемого процесса. Каждая группа состоит из «HTTP» запросов к серверу и «Java» функций, необходимых для автоматизации ручных действий специалиста.

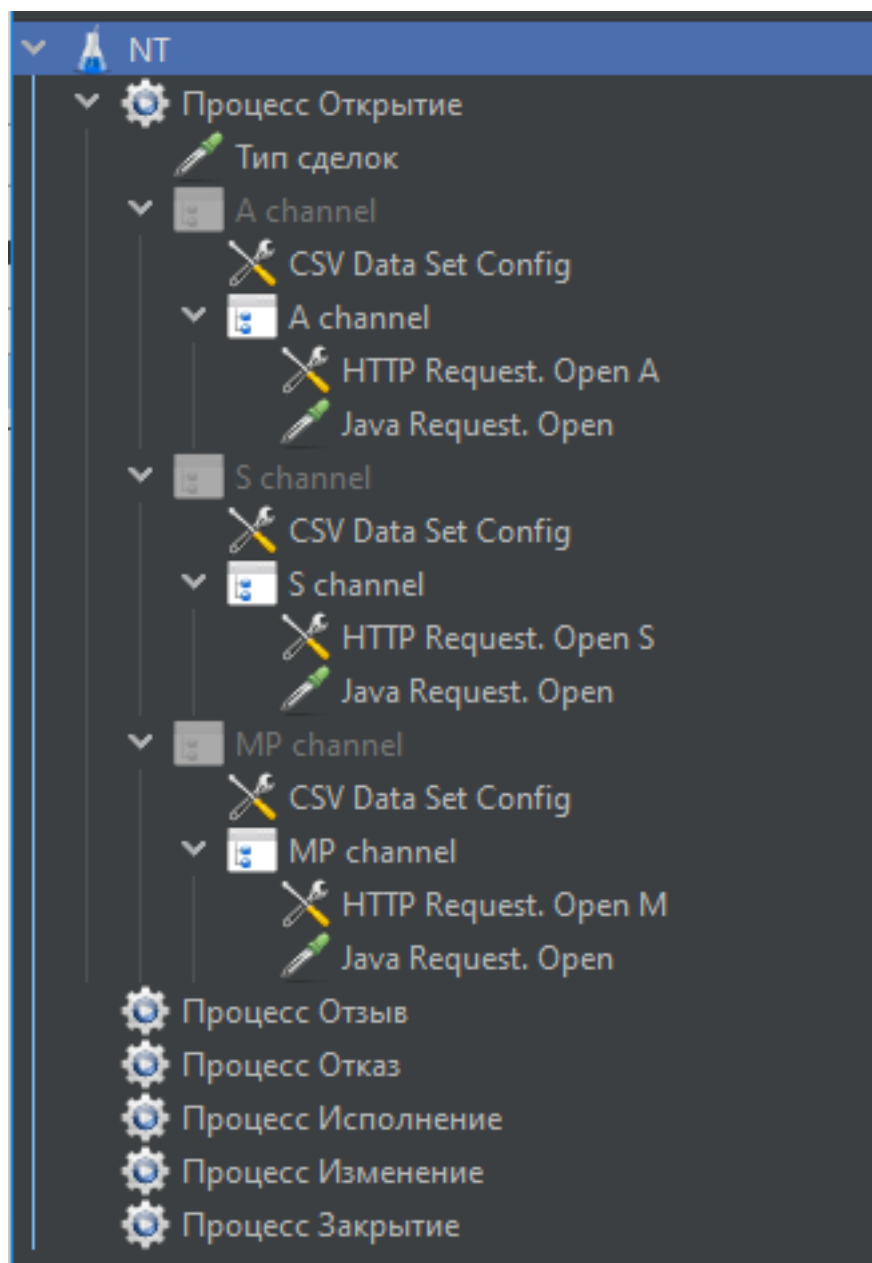


Рисунок 1. Структура теста

После разработки структуры теста необходимо добавить заглушки. Так как ранее нагрузочное тестирование системы не проводилось, часть внешних сервисов на нагрузочном стенде не настроена, это может повлиять на корректность создания операций процесса.

Заглушка выступает в качестве небольшого фрагмента кода, заменяющего другой компонент во время тестирования. Ключевым преимуществом использования заглушки является возможность получить согласованные результаты, чтобы упростить написание тестов.

Заглушка для сервиса «Notification» имитирует положительный ответ сервиса после обработки запроса от тестируемой системы. Данная заглушка необходима для процессоров по созданию операций уведомлений.

Заглушка для сервиса «Task» имитирует негативный ответ сервиса, в виду специфики реализации обработчиков по созданию задач, данный ответ необходим для прекращения выполнения метода по созданию задачи.

Описанные заглушки разрабатывались в системе «WireMock», они подключаются к проекту

через систему «OpenShift».

Также разработана локальная заглушка в программном коде теста, имитирующая недоступность базы данных. Данная заглушка применяется в стресс тесте.

Разработанные заглушки представлены в таблице 1.

Таблица 1.

Заглушки системы

Наименование сервиса	endpoint	Время ответа, мс	Тело ответа
«Notification»	https://localhost/lib/api/v2/notification	3	{ "status": "ok", "code": 200 }
«Task»	https://localhost/lib/task-ui	3	{ "id": get.id, "errorMessage": "Ошибка при обрабо }
БД	Jbs:oracle:thin:9901323.agr.al.ru:9838	3	class Connect: DatabaseConn { public List<string> GetD { return new List<string>(){ "answer500", "ansv } }

Следующим этапом настраивается «CI/CD» в системе «Jenkins». Для удобства запуска тестов произведена настройка возможности выбора теста и времени для его запуска. Также есть возможность остановки выполнения теста (рис.2)

Выбор теста

- ☐ Стресс тест
- ☐ Нагрузочный тест
- ☒ Тест надежности
- ☐ Тест производительности

запустить

д.м.г. 00:00

ОСТАНОВИТЬ

Рисунок 2. Выбор теста

Список литературы:

1. Электронный ресурс <https://habr.com/ru/companies/itsumma/articles/682022/>