МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ЛАБОРАТОРНОГО ЗАНЯТИЯ

*Дисциплина*: Тестирование программного обеспечения.

*Тема*: «Нагрузочное тестирование и тестирование производительности веб-приложений».

*Цель работы*: научиться создавать простые сценарии тестирования и выполнять их с помощью JMeter.

Основные задачи лабораторной работы:

* научиться простейшие сценарии;
* научиться создавать сценарии с помощью семплера;
* научиться записывать и визуализировать результаты отработанных тестов.

Задания к лабораторной работе:

В качестве тестируемого сайта предлагается http://student.miu.by/.

**Задание 1**. **Создать простейший сценарий, с помощью которого можно узнать среднее время ответа сервера.**

Нам понадобится *группа пользователей*, которые будут *заходить на страницу student.miu.by*, и *сборщик результатов* выполненного теста.

1. Добавить группу пользователей через контекстное меню Test plan *(Add > Threads (Users) > Thread Group).* Количество циклов *(Loops Count)* изменить на 10.
2. Добавить запрос страницы через контекстное меню группы пользователей *(Add > Sampler > HTTP Request)*. В группе полей Web Server установить значение Server Name or IP на student.miu.by.
3. Добавить сборщик результатов через контекстное меню группы пользователей *(Add > Listener > Aggregate Report)*.

Сохраните сценарий, затем запустите *(Run > Start, CTRL-R)*. Откройте свойства Aggregate Report. Среднее время ответа указано в столбце Average. Количество пройденных циклов указано в столбце Samples. Сохраните результат в файл, нажав Save Table Data.

Если необходимо очистить результат предыдущих запусков, то нажмите в главном меню *Run > Clear* или сочетание клавиш *CTRL-E*.

**Задание 2. Записать сценарий с помощью семплера. Сценарий: просмотр рейтинга студентов с 1 по 6й курс.**

Записывать сценарий необходимо для *группы пользователей*. Для записи будем использовать встроенный в JMeter *HTTP прокси-сервер* и браузер.

1. Добавьте *группу пользователей*.
2. Добавить прокси-сервер через контекстное меню “Workbench” *(Add > Non-Tests Element > HTTP(S) Test Script Recorder)*. В группе полей Test Plan content установить значение Target Controller на “Test Plan > Thread Group”. Чтобы исключить запись запросов скриптов, стилей и картинок, в группе полей URL patterns to Exclude нажмите кнопку Add suggested Excludes.

Нажать кнопку Start.

1. Переключиться на браузер. Настроить его на прокси-сервер с доменом localhost и портом 8080.
2. В браузере зайти на страницу student.miu.by, затем последовательно переключаться с 1-го по 6-й курс.

После, верните прежние настройки браузера.

1. Переключиться на JMeter. Нажать кнопку Stop в настройках прокси-сервера.

В результате для группы пользователей будет записано семь действий – HTTP-запросов.

Для Thread Group добавьте Summary Report *(Add > Listeners > Summary Report)*. Запустите сценарий. Сохраните полученный результат *(Save Table Data)*.

**Задание 3. Описать следующий сценарий. Группа студентов входит в «учебную карточку студента». Они плутают среди страниц «приказы», «график оценок» и «рейтинг в группе». Каждый просматривает эти страницы 5 раз, а затем покидает «учебную карточку», нажав «выход».**

1. Добавить группу числом 10 пользователей. Значение Ramp-Up Period установить 0.
2. Добавить группе настройки по умолчанию для всех HTTP-запросов *(Add > Config Element > HTTP Request Defaults)*. В поле Server Name or IP указать student.miu.by.
3. Для просмотра результатов добавить Aggregate Graph.
4. Добавить HTTP Request с именем «Главная». Path: /
5. Добавить HTTP Request с именем «Вход».

Значение Path: /learning-card.html.

Значение Method: POST.

Значение Send Parameters With the Request:

|  |  |
| --- | --- |
| Name: | Value: |
| act | regnum |
| id | id |
| regnum | {НОМЕР СТУДЕНЧЕСКОГО} |

1. Добавить HTTP Request с именем «Выход». Значение Path: /rating/~/log.out.html

Теперь пользователи могут войти и выйти из «учебной карточки студента». Но важно убедиться, что вход прошел успешно. На примере student.miu.by признаком успешного входа будет отсутствие формы ввода номера студенческого билета. Чтобы проверить её наличие необходимо:

1. Добавить в шаг «Вход» XPath Assertion *(Add > Assertions > XPath Assertion)*.
2. В группе полей XPath Parsing Options установить флаг Use Tidy.
3. В поле XPath Assertion написать выражение: //form[@action="/learning-card.html"]
4. Отметить флаг «True if nothing matches»
5. В настройках группы пользователей отметить вариант «Stop Test».

Чтобы сымитировать хаотичное посещение страниц приказов, графика оценок и рейтинга нужно:

1. Добавить к группе пользователей Loop Controller *(Add > Logic Controller > Loop Controller)* перед шагом «Выход». *Перетащить мышкой контроллер перед шагом «выход»*. Количество циклов *(Loops)* установить на 5.
2. К Loop Controller добавить Random Order Controller *(Add > Logic Controller > Random Order Controller)*.
3. Внутри Random Order Controller создать 3 HTTP Request с значениями полей:

|  |  |
| --- | --- |
| Name | Path |
| Рейтинг в группе | /rating/~/my.group.html |
| Приказы | /learning-card/~/my.orders.html |
| График оценок | /learning-card/~/my.graph.html |

Запустите сценарий. Сохраните результаты и полученный график из Aggregate Graph.

**Контрольные вопросы:**

1. Почему первые шаги в сценарии проходят дольше последующих?
2. Как смоделировать запрос веб-страницы?
3. Как смоделировать нагрузку от нескольких пользователей одновременно?
4. Как регулировать «плавность» нагрузки?
5. Как организовать автоматическую запись сценария?
6. Как зациклить часть сценария?
7. Как выполнить шаги в случайном порядке?
8. Как проверить наличие определенных элементов на странице?