

ВНЕДРЕНИЕ СИСТЕМЫ УПРАВЛЕНИЯ ТЕСТОВЫМИ ДАННЫМИ В ПРОЕКТ ПО АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ

С.С. Кириллов, старший специалист по тестированию программного обеспечения
«КонсультантПлюс»
(Россия, г. Москва)

DOI: 10.24412/2500-1000-2022-6-1-150-153

Аннотация. в рамках данной статьи освещается процесс внедрения системы управления тестовыми данными в действующий проект по автоматизации тестирования. Проведен анализ существующих подходов к организации управления тестовыми данными, а также предложена методика, которая позволила добиться независимости тестов друг от друга по данным, что в конечном итоге дало возможность развернуть систему параллельного запуска автоматизированных тестов. В качестве объекта исследования выступил проект по автоматизации тестов пользовательского интерфейса веб-приложения в связке с языком программирования Java.

Ключевые слова: автоматизированное тестирование, тестирование программного обеспечения, тестовые данные, selenium webdriver, selenide, web, браузер, пользовательский интерфейс, java.

Обоснование актуальности темы

Согласно ряду исследований, описывающих природу недетерминированного поведения автоматизированных тестов, в число наиболее частых причин их нестабильности входят: зависимость тестов от порядка их запуска, а также параллельные операции в тестах [1,2,3].

В рамках данного исследования команда тестирования столкнулась с аналогичными проблемами. Так, с увеличением количества автоматизированных тестов в проекте, время их выполнения начало возрастать. Одним из решений данной проблемы стало внедрение параллельного запуска тестов. Однако, работа проекта в параллельном режиме осложнялась использованием общих тестовых данных, что негативно сказывалось на стабильности проекта. Поэтому одной из приоритетных задач, при построении эффективного инструмента автоматизации, стало обеспечение независимости тестов друг от друга по данным.

Анализ текущих вариантов с их плюсами и минусами

Можно выделить несколько уровней создания и управления тестовыми данными в области автоматизации тестирования пользовательского интерфейса:

- **Графический интерфейс веб-приложения**

Добавление тестовых данных с помощью графического интерфейса веб-приложения является наиболее простым решением. Например, с помощью инструментов на базе Selenium WebDriver можно добавить скрипт, который будет регистрировать пользователя с необходимым набором атрибутов непосредственно в браузере. Данный подход имеет ряд существенных недостатков: низкая скорость и нестабильность работы теста. Использование графического интерфейса возможно, когда отсутствует доступ к другим уровням взаимодействия с тестируемым приложением.

- **Программный интерфейс приложения**

Наличие доступа к программному интерфейсу приложения позволяет избежать недостатков, характерных при использовании графического интерфейса. С помощью такой прослойки можно быстрее и надежнее взаимодействовать с серверной частью приложения, минуя графический интерфейс. Для эффективного использования программного интерфейса, необходимо наличие актуальной и детализированной документации. Кроме того, требуется

более высокий уровень компетенции от тех, кто использует данный инструмент.

- **База данных**

Взаимодействие напрямую с базой данных является наиболее быстрым способом генерации тестовых данных. Благодаря такому подходу появляется возможность полного контроля над параметрами системы. Между тем, такая свобода действий имеет ряд ограничений. Необходимо хорошее знание схемы базы данных и как она наполняется из тестируемой системы.

Кроме того, необходимо отслеживать изменения в структуре таблиц базы данных, чтобы оперативно вносить изменения в тестовое окружение.

Пути решения

Проанализировав подходы к управлению тестовыми данными, а также рассмотрев структуру существующего тестового проекта, была разработана методика комплексного управления тестовыми данными. Схема реализации методики представлены на Рис 1.

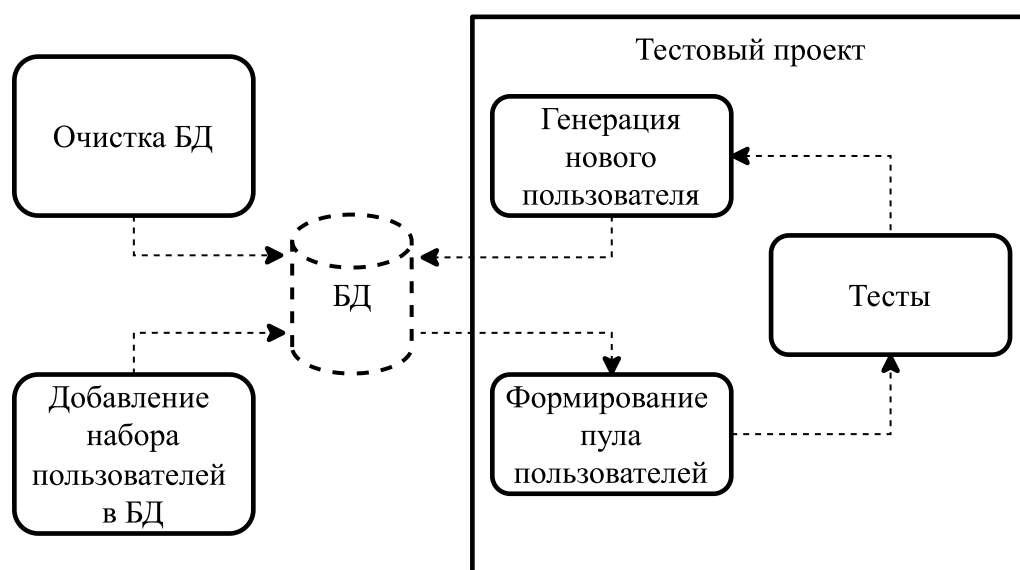


Рис. 1. Схема внедрения управления тестовыми данными в проекте

Рассмотрим представленную схему работы с тестовыми данными подробнее.

Восстановление тестовых данных

Для получения исходного состояния тестовой базы данных, перед запуском автоматизированных тестов вызывается скрипт для ее очистки с сохранением структуры таблиц. Далее в цикле создается необходимое количество стандартных пользователей под каждый браузер, что позволяет масштабировать запуск тестов. Количество тестовых пользователей (Пт) для проекта рассчитано по следующей формуле:

$$Пт = Кп \times 1,5 \times Бр ,$$

где Кп – количество потоков при параллельном запуске;

Бр – количество браузеров, запускаемых одновременно;

Набор пользователей сформирован с небольшим запасом, что позволяет избежать ожиданий, например, при запуске дополнительных тестов с использованием механизма поставки тестовых данных, реализованный в TestNG [4].

Создание пула пользователей

Как было сказано ранее, перед запуском тестов база данных наполняется стандартными пользователями, которые имеют идентификатор, соответствующий браузеру, в котором стартуют тесты. Для того, чтобы тесты не использовали одинаковых пользователей при параллельном запуске, в тестовый проект был интегрирован дизайн-паттерн «пул объектов» [5]. Реализация данного паттерн представляет собой очередь из пользователей, сформированную из существующего набора в базе данных.

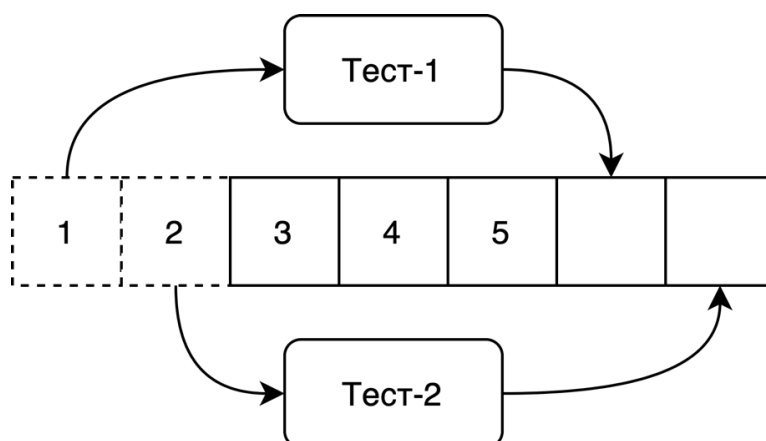


Рис. 2. Схема организации работы пула пользователей в тестовом проекте

Каждый тест при старте получает из пула свободного пользователя, а по окончании работы, возвращает его в конец очереди. Таким образом, в любой момент времени тесты независимы по данным и не мешают друг другу.

Генерация тестовых данных непосредственно в тесте

Часть тестов в проекте использует специфичный набор атрибутов, который не позволяет использовать стандартных поль-

зователей из пула. Решением данной проблемы стало создание пользователя непосредственно в тесте или перед запуском определенного класса тестов. В качестве инструмента выступил дизайн-паттерн «Строитель» [6], который дает возможность создавать в базе данных нового пользователя с необходимым набором свойств. Ниже приведен фрагмент кода на основе данного дизайн-паттерна с использованием языка программирования Java.

```
User user = User.add()
    .setName(userName)
    .setEmail(userEmail)
    .hasAdmin(adminName)
    .create();
```

Реализация данного паттерна в тестовом проекте, содержит более 15 характеристик пользователя, которые можно комбинировать в зависимости от цели теста.

В заключении важно отметить, что независимость тестов является одним из ключевых элементов построения эффективной системы автоматизации тестирова-

ния. Благодаря проведенному комплексу мероприятий, удалось в короткие сроки внедрить в существующий тестовый проект систему управления тестовыми данными и тем самым изолировать тесты друг от друга. В дальнейшем это позволило выстроить масштабируемую систему параллельного запуска тестов.

Библиографический список

1. Fowler M., Eradicating Non-Determinism in Tests [Электронный ресурс] URL: <https://martinfowler.com/articles/nonDeterminism.html> (дата обращения 19.06.2022)
2. Luo Q., Hariri F., Eloussi L. and Marinov D., An empirical analysis of flaky tests. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2014). Association for Computing Machinery, New York, NY, USA, 643–653. DOI: <https://doi.org/10.1145/2635868.2635920>
3. Romano A., Song Z., Grandhi S., Yang W. and Wang W., An Empirical Analysis of UI-Based Flaky Tests, 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 1585-1597, <https://doi.org/10.1109/ICSE43902.2021.00141>

4. TestNG – is a testing framework [Электронный ресурс] URL: <https://testng.org/> (дата обращения 19.06.2022)

5. Grand M., Patterns in Java - Object Pool, Volume 1, John Wiley & Sons, 1998.

6. Gamma E., Helm R., Johnson R, Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, 1994.

IMPLEMENTING A TEST DATA MANAGEMENT SYSTEM IN AN EXISTING TEST AUTOMATION PROJECT

S.S. Kirillov, *Senior software tester*

«ConsultantPlus»

(Russia, Moscow)

***Abstract.** This article highlights the process of implementing a test data management system in an existing test automation project. An analysis of existing approaches to the organization of test data management was carried out, and a methodology was proposed that allowed to achieve independence of tests from each other in terms of data, which eventually made it possible to deploy a system of running automated tests in parallel. The object of research was a project to automate tests of the user interface of a Web-application in conjunction with the Java programming language.*

***Keywords:** automated testing, software testing, test data, selenium webdriver, selenide, web, browser, ui, java.*