

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ МИКРОСЕРВИСОВ ПРИЛОЖЕНИЯ

Рассматривается автоматизация тестирования микросервисов приложения средствами языка программирования JavaScript и Postman.

ВВЕДЕНИЕ

Современный мир все больше требует вовлеченности информационных технологий во все сферы жизнедеятельности человечества, тем самым ускоряя и совершенствуя свои процессы и технологии. В таком ритме продуктам необходимо оптимизироваться и подстраиваться под потребителя, чтобы занимать лидирующие позиции. Для осуществления такой цели жизненный цикл разработки программного обеспечения (SDLC [1]) должен быть минимизирован на всех его стадиях, но уделяя особое внимание качеству продукта.

1. ОСНОВНОЙ РАЗДЕЛ

Фаза тестирования играет важную роль в иерархии жизненного цикла разработки ПО, присутствует практически на всех этапах, пока продукт не достигнет необходимых стандартов качества. Исходя из современных тенденций разработки клиент-серверных приложений предпочтение отдается микросервисной архитектуре [2]. Из этого вытекает необходимость тестирования программного интерфейса приложения (API [3]) которое становится особенно востребованным на первых этапах проектирования, так как именно оно позволяет в полной мере убедиться в правильности работы микросервиса и интеграции отдельных сервисов, а также в соблюдении логики функционирования системы в целом. Реализация программного интерфейса приложения может быть как внутренней когда программные компоненты находятся внутри приватной сети и нет прямой возможности получить к ним доступ из вне, либо открытой – когда программные компоненты имеют прямой выход в глобальный интернет, что позволяет внешним пользователям или другим программам получать информацию, которую можно интегрировать в свои приложения. На базе таких архитектурных решений выстраиваются специальные техники тест-дизайна, которые подразумевают под собой этап процесса тестирования программного обеспечения, на котором проектируются и создаются тестовые случаи для покрытия логики как отдельного модуля так и интеграции с другими модулями в соответствии с определёнными ранее критериями качества и целями тестирования. Специ-

фикой тестирования API является то, что для программного интерфейса приложения необходима базовая техническая подготовка и необходимость использования дополнительных инструментов. На практике для API тестирования применяются такие средства как язык программирования JavaScript [4] и сервис Postman [5]. На языке JavaScript реализуются алгоритмы тестирования микросервисов, имеющие структуру вида: входные параметры, реализация вызова API интерфейса, получение результата, сравнение полученного результата с ожидаемым. Postman - сервис с открытым исходным кодом для запуска реализованных алгоритмов тестирования с возможностью группировки результатов и создания отчетов.

ЗАКЛЮЧЕНИЕ

Целесообразностью внедрения автоматизации тестирования обусловлена, в первую очередь, возможностью ускорить релизы, позволяя доставить новый функционал либо исправление критической ошибки намного быстрее. Во-вторых, автоматизация рутинных и частых проверок с большим количеством устройств, версий браузеров и операционных систем позволит снизить нагрузку на QA специалистов. При этом, в долгосрочной перспективе, снизит как расходы на тестирование, так и риски, связанные с человеческим фактором. Также не исключено является автоматизация тестирования производительности приложения в условиях одновременной работы с большим количеством данных и пользователей.

Список литературы

1. ISO/IEC 12207:1995 Information technology – Software life cycle processes [Electronic resource] – Mode of access: <https://www.iso.org/standard/21208.html>
2. Кочер П.С. Микросервисы и контейнеры Docker / пер. с англ. А. Н. Киселева. – М.: ДМК Пресс, 2019. – С.16-21.
3. API [Electronic resource] – Mode of access: <https://developer.mozilla.org/ru/docs/Glossary/API>
4. JavaScript: Учебные материалы [Electronic resource] – Mode of access: <https://developer.mozilla.org/ru/docs/Web/JavaScript>
5. The Postman API Platform [Electronic resource] — Mode of access: <https://www.postman.com/api-platform/>