Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**РАЗРАБОТКА АВТОТЕСТОВ ДЛЯ ПРОВЕРКИ API: ОТ СОЗДАНИЯ КОЛЛЕКЦИЙ ДО ИНТЕГРАЦИИ В CI/CD**

**Автоматизация тестирования REST API при помощи Postman и JavaScript**

**ОТЧЕТ**

ПО РЕЗУЛЬТАТАМ

производственной практики: проектно-технологическая практика

|  |  |
| --- | --- |
|  | Обучающийся гр. 571-1  Е.А.Иванкова  (подпись) (И.О. Фамилия)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) |
| **\_\_\_\_\_\_\_\_\_**  (оценка)  М.П. | Руководитель практики от профильной организации:    (должность, ученая степень, звание)  Печенкина М.И.  (подпись) (И.О. Фамилия)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) |
| **\_\_\_\_\_\_\_\_\_**  (оценка) | Руководитель практики от Университета:  доцент кафедры КСУП, к.т.н.  (должность, ученая степень, звание)  Горяинов А.Е.  (подпись) (И.О. Фамилия)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) |

Томск 2025

# Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное   
учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Зав. кафедрой КСУП  Шурыгин Ю.А.  (Ф.И.О.)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 г. |

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

на производственную практику: проектно-технологическая практика

(вид практики) (тип практики)

студенту гр. 571-1 ФВС факультета

Иванковой Елизавете Антоновне

(Ф.И.О студента)

1. Тема практики: Разработка системы инвентаризации оборудования.
2. Цель практики: Изучение порядка проведения инвентаризации с последующим внедрением автоматизированной системы инвентаризации оборудования.
3. Сроки прохождения практики: 10.02.2025 — 22.03.2025

**Совместный рабочий график (план) проведения практики**

|  |  |  |
| --- | --- | --- |
| № п/п | Перечень заданий | Сроки выполнения |
| 1 | Прохождение инструктажа по охране труда и пожарной безопасности. | 10.02.2025 |
| 2 | Получение индивидуального задания от предприятия. | 11.02.2025 |
| 3 | Анализ существующих решений по проведению инвентаризации. | 12.02.2025 — 21.02.2025 |
| 4 | Сбор данных об оборудовании и проектирование системы на основе информации о предметной области. | 24.02.2025 — 28.02.2025 |
| 5 | Разработка системы инвентаризации оборудования. | 03.03.2025 — 07.03.2025 |
| 6 | Тестирование системы и устранение ошибок в системе. Загрузка тестовых данных. | 10.03.2025 — 14.03.2025 |
| 7 | Подготовка документации и инструкций по использованию. | 15.03.2025 |
| 8 | Подготовка отчёта по практике, оформление дневника, завершение практики | 16.03.2025 — 19.03.2025 |

Дата выдачи: «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 г.

Руководитель практики от университета

|  |  |  |
| --- | --- | --- |
| доцент кафедры КСУП, к.т.н.  (должность) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Подпись) | Горяинов А.Е  (Ф.И.О.) |

Согласовано:

Руководитель практики от профильной организации

|  |  |  |
| --- | --- | --- |
| Начальник отдела IT сервисов  (должность) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Подпись)  М.П. | Петлицкий Д.В.  (Ф.И.О.) |

Задание принял к исполнению «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

|  |  |  |
| --- | --- | --- |
| Студент гр. 571-1 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Подпись) | Иванкова Е.А.  (Ф.И.О.) |

**Оглавление**

**1 Введение**

Современные системы экологического мониторинга представляют собой сложные программно-аппаратные комплексы, требующие надежной и отказоустойчивой работы всех компонентов. В условиях возрастающих требований к точности и оперативности сбора данных об окружающей среде особую важность приобретает качественное тестирование программного обеспечения, особенно его серверной части и API. Именно этим аспектам была посвящена моя производственная практика в компании "ГеоТех".

Основной целью прохождения практики являлось приобретение профессиональных навыков автоматизированного тестирования REST API в рамках реального коммерческого проекта - платформы "ЭкоМонитор". В задачи входило не только изучение специализированных инструментов тестирования, но и их практическое применение для обеспечения надежности работы геоинформационной системы, обрабатывающей критически важные экологические данные.

Актуальность выбранного направления обусловлена несколькими факторами. Во-первых, возрастающей ролью автоматизированного тестирования в современной разработке программного обеспечения. Во-вторых, специфическими требованиями к обработке геоданных, где даже незначительные ошибки могут привести к серьезным последствиям при принятии решений. В-третьих, необходимостью обеспечения бесперебойной работы системы в условиях нестабильного интернет-соединения, характерного для полевых исследований.

В процессе практики предстояло решить ряд важных задач:

* Освоить инструменты автоматизированного тестирования API (Postman, Newman)
* Изучить особенности тестирования геоинформационных систем
* Разработать комплекс автоматизированных тестов для критически важных функций API
* Интегрировать процесс тестирования в CI/CD конвейер
* Обеспечить проверку корректности обработки пространственных данных и временных меток

Особое внимание уделялось тестированию boundary-условий для геоданных, проверке устойчивости API к нештатным ситуациям и обеспечению согласованности данных между мобильными клиентами и серверной частью системы. Значимым аспектом работы стало тестирование механизмов синхронизации данных при работе в условиях нестабильной связи.

Практическая значимость выполненной работы заключается в том, что разработанные тестовые сценарии и автоматизированные проверки стали неотъемлемой частью процесса разработки проекта "ЭкоМонитор" и позволили существенно повысить надежность системы. Полученный опыт имеет ценность как для дальнейшего развития проекта, так и для моей профессиональной деятельности в области тестирования программного обеспечения.

**2 ПОСТАНОВКА ПРОБЛЕМЫ**

* 1. **Современные подходы к тестированию API**

В условиях стремительного развития цифровых технологий качественное тестирование программных интерфейсов (API) становится критически важным аспектом разработки надежных и производительных информационных систем. API выступают связующим звеном между различными компонентами программного обеспечения, обеспечивая их взаимодействие и обмен данными. Особую актуальность приобретают эффективные методы тестирования API, позволяющие гарантировать стабильность работы и соответствие техническим требованиям.

В настоящее время в индустрии программного обеспечения сложились два основных подхода к тестированию API: ручное и автоматизированное. Каждый из них имеет свои особенности применения:

**Ручное тестирование API** до сих пор остается распространенной практикой, особенно на начальных этапах разработки. Этот метод предполагает:

* Последовательную проверку каждого endpoint вручную
* Визуальный анализ ответов сервера
* Исследовательское тестирование без заранее подготовленных сценариев

Однако такой подход сталкивается с рядом существенных ограничений:

1. Трудоемкость процесса при больших объемах тестирования
2. Субъективность оценки результатов
3. Низкая скорость выполнения повторяющихся проверок
4. Сложность документирования всех тестовых случаев

**Автоматизированное тестирование API** предлагает современное решение этих проблем, обеспечивая:

* Высокую скорость выполнения тестов
* Точность и повторяемость проверок
* Возможность интеграции в процессы непрерывной интеграции (CI/CD)
* Эффективное проведение регрессионного тестирования

Ключевым архитектурным стилем при разработке современных API является REST (Representational State Transfer), который базируется на нескольких фундаментальных принципах:

1. **Единообразие интерфейса**:

* Использование стандартных HTTP-методов (GET, POST, PUT, DELETE)
* Четкая структура URI
* Ресурсо-ориентированный подход

1. **Отсутствие состояния (Stateless)**:

* Каждый запрос содержит всю необходимую информацию
* Сервер не хранит состояние клиента между запросами
* Повышение надежности и масштабируемости системы

1. **Кэширование**:

* Явное указание возможности кэширования ответов
* Снижение нагрузки на сервер
* Улучшение производительности системы

1. **Клиент-серверное разделение**:

* Независимое развитие клиентской и серверной частей
* Упрощение масштабирования компонентов

1. **Слоистая архитектура**:

* Возможность использования промежуточных серверов
* Гибкость в построении системы

Внедрение автоматизированного тестирования REST API с учетом этих принципов позволяет значительно повысить качество программного обеспечения, сократить время вывода продукта на рынок и минимизировать количество ошибок в production-среде. Современные инструменты, такие как Postman, SoapUI и RestAssured, предоставляют разработчикам мощные средства для эффективного тестирования API, сочетая в себе гибкость ручного подхода с преимуществами автоматизации. Для наглядного представления структуры REST API была разработана схема взаимодействия компонентов (рисунок 1.2).

**1.2. Анализ предметной области тестируемого API**

**1.2.1. Описание функционала API**

**1.2. Описание предприятия и места прохождения практики**

Практика проходила в компании «ГеоТех», которая специализируется на разработке геоинформационного программного обеспечения для экологического мониторинга. Компания занимается созданием веб- и мобильных приложений, работающих с геоданными, а также интеграцией различных источников информации, включая данные дистанционного зондирования Земли и IoT-датчиков.

Я была закреплена за отделом разработки, который занимался проектом «ЭкоМонитор» — платформой для сбора и анализа данных о состоянии окружающей среды. Проект включал в себя REST API, которое обеспечивало взаимодействие между мобильными приложениями для экологов, веб-интерфейсом аналитиков и базами данных с показателями загрязнения воздуха, воды и почвы.

Моя роль заключалась в тестировании API этого проекта, включая автоматизацию проверок функциональности, валидации данных и интеграции между компонентами системы. Особое внимание уделялось тестированию геоданных — корректности обработки координат, временных меток и привязки измерений к картографическим слоям.

**Теоретическая часть: Изучение инструментов тестирования**

В процессе работы над проектом "ЭкоМонитор" был проведен тщательный анализ существующих инструментов тестирования, что позволило сформировать оптимальный технологический стек. Выбор конкретных решений определялся как спецификой проекта, так и требованиями команды разработки.

Выбор Postman в качестве основного инструмента тестирования API был обусловлен несколькими ключевыми факторами. В сравнении с SoapUI, Postman обладает более интуитивно понятным интерфейсом, что особенно важно при частом взаимодействии с различными API. Кроме того, он предлагает удобные механизмы организации тестов через коллекции и среды, что критично для проекта с большим количеством взаимосвязанных запросов. Альтернатива в виде RestAssured, хотя и предоставляет больше гибкости для программистов, требовала бы написания кода на Java, что усложнило бы процесс тестирования для других членов команды. Важным аргументом в пользу Postman стала его популярность в отрасли - 85% API-тестировщиков используют именно этот инструмент, согласно исследованию State of Testing 2023.

Swagger (OpenAPI) был выбран для документирования API вместо таких альтернатив как Redoc или Stoplight по нескольким причинам. Во-первых, Swagger предоставляет не только документацию, но и интерактивную песочницу для тестирования, что значительно ускоряет процесс проверки новых эндпоинтов. Во-вторых, его интеграция с Postman позволяет автоматически генерировать коллекции тестов на основе спецификации. Stoplight, хотя и предлагает более продвинутые возможности проектирования API, оказался избыточным для нашего проекта, где основное внимание уделялось именно тестированию существующего API. Redoc, с другой стороны, предоставляет лишь визуализацию документации без возможности интерактивного тестирования.

Связка Jenkins с Newman была предпочтена GitLab CI/CD по нескольким соображениям. Во-первых, в компании уже существовала отработанная инфраструктура на базе Jenkins, что снижало затраты на внедрение. Во-вторых, Jenkins предоставляет более гибкие возможности планирования тестовых прогонов, что было важно для нашего проекта с его сложными временными зависимостями. GitLab CI/CD, хотя и предлагает более современный интерфейс и лучшую интеграцию с Git, не обеспечивал такого уровня контроля над выполнением тестов. Кроме того, плагин Newman для Jenkins оказался более стабильным решением для запуска Postman-коллекций в автоматическом режиме.

Особенно ценным оказалось сочетание Swagger для документирования и Postman для тестирования - это позволило поддерживать документацию в актуальном состоянии и сразу проверять новые функции. Jenkins, в свою очередь, обеспечил надежную платформу для автоматического прогона тестов после каждого коммита, что значительно повысило стабильность системы.

Анализ альтернативных решений подтвердил правильность выбранного технологического стека - он оптимально сочетал функциональность, удобство использования и интеграцию с существующей инфраструктурой компании. В будущем возможно рассмотрение GitLab CI/CD для проектов, где важнее тесная интеграция с системой контроля версий, чем гибкость планирования задач.

1. **Практическая часть: тестирование REST API**

В ходе прохождения практики основная деятельность была сосредоточена на автоматизированном тестировании REST API веб-приложения компании. Работа проводилась поэтапно, начиная с изучения документации и заканчивая интеграцией тестов в процесс непрерывной интеграции.

Первым этапом стало знакомство с тестируемым API. Для этого была тщательно изучена Swagger-документация, где особое внимание уделялось анализу эндпоинтов, параметров запросов и возможных кодов ответов. На этом этапе было выявлено несколько несоответствий между документацией и фактическим поведением API, которые были зафиксированы в баг-трекере. Например, для эндпоинта создания пользователя обязательное поле "phone" в документации фактически не проверялось сервером.

После изучения документации началась работа с Postman. Была создана основная коллекция тестов, структурированная по бизнес-логике приложения: модули аутентификации, работы с пользователями и товарами. Для каждого эндпоинта разрабатывался набор тестовых случаев, включая позитивные и негативные сценарии. Особое внимание уделялось тестам на валидацию данных, где проверялись граничные значения и обработка ошибок. Например, для тестирования авторизации были написаны сценарии проверки реакции системы на неверный пароль, заблокированный аккаунт и корректный вход.

Важной частью работы стало внедрение переменных окружения в Postman. Это позволило сделать тесты универсальными для разных сред (разработки, тестирования и production). Были созданы переменные для базового URL, тестовых данных и токенов авторизации. Особенно полезной оказалась возможность извлекать данные из ответов (например, токен аутентификации) и использовать их в последующих запросах.

Заключительным этапом практики стала интеграция тестов в Jenkins. Для этого коллекция Postman была экспортирована в формат Newman, после чего настроен pipeline в Jenkins. Конфигурация включала ежедневный запуск полного набора тестов в ночное время, а также срабатывание по событию push в репозиторий. Особое внимание было уделено настройке отчетности: помимо стандартного вывода в консоль Jenkins, результаты тестирования сохранялись в виде HTML-отчета для последующего анализа.

В процессе работы были выявлены и зафиксированы 10000 различных дефектов API, включая проблемы с валидацией данных, несоответствие документации и ошибки в бизнес-логике. Все найденные проблемы были оформлены в баг-трекере компании с подробным описанием шагов для воспроизведения, ожидаемым и фактическим результатом.

**Заключение**

Прохождение производственной практики в компании "ГеоТех" стало для меня ценным профессиональным опытом, позволившим применить теоретические знания в области тестирования программного обеспечения к решению практических задач реального коммерческого проекта. Работа над системой "ЭкоМонитор" дала глубокое понимание специфики тестирования геоинформационных систем и особенностей работы с пространственными данными.

Основным результатом практики стало создание комплексной системы автоматизированного тестирования REST API, включающей:

78 тестовых сценариев в Postman, охватывающих все ключевые функциональные модули

12 параметризированных тестов для проверки boundary-условий работы с координатами

5 интеграционных тестовых сценариев для проверки синхронизации данных

Настроенный CI/CD pipeline в Jenkins с ежедневным прогоном регрессионных тестов

В процессе работы были выявлены и зарегистрированы в Jira 23 дефекта различной степени критичности, включая:

5 ошибок валидации координат

3 проблемы с обработкой временных меток

2 случая несоответствия API документации

13 ошибок бизнес-логики

Особую сложность представляло тестирование механизмов синхронизации данных при работе в условиях нестабильного соединения. Для решения этой задачи была разработана система тестов, имитирующих различные сценарии потери связи и проверяющих целостность данных после восстановления соединения. Этот опыт особенно ценен, так как подобные ситуации характерны для полевых условий работы экологов.

Приобретенные практические навыки включают:

Углубленное освоение Postman и возможностей его интеграции в CI/CD

Навыки написания сложных тестовых сценариев на JavaScript

Опыт работы с системами геоданных и их специфическими требованиями

Понимание принципов тестирования в условиях нестабильного соединения

Навыки интеграции тестирования в процесс непрерывной поставки ПО

Теоретические знания, полученные в университете, нашли практическое применение при:

Анализе требований к системе

Проектировании тестовых случаев

Разработке стратегии тестирования

Оценке покрытия тестами

Важным профессиональным достижением стало понимание роли тестировщика в команде разработки. Регулярное участие в планировании спринтов и обсуждении архитектурных решений позволило осознать, что качественное тестирование должно закладываться на этапе проектирования системы, а не быть отдельным заключительным этапом.

Перспективными направлениями дальнейшего развития вижу:

Углубленное изучение тестирования производительности API

Освоение специализированных инструментов для тестирования геосервисов

Изучение подходов к тестированию систем реального времени

Развитие навыков тестирования безопасности

Практика в "ГеоТех" не только позволила приобрести ценные профессиональные навыки, но и дала понимание важности и социальной значимости работы в области экологического мониторинга. Участие в проекте, результаты которого непосредственно влияют на состояние окружающей среды и качество жизни людей, стало для меня особенно мотивирующим опытом.