РАЗРАБОТКА СИСТЕМЫ ТЕЛЕМЕТРИИ ДЛЯ МОНИТОРИНГА И АНАЛИЗА РАБОТЫ ПОЛЬЗОВАТЕЛЕЙ С ИНФОРМАЦИОННЫМИ СИСТЕМАМИ

Малько А.Ю.[1], Осипова В.В.[2]
[1]Томский политехнический университет, гр. 8ВМ31 ОИТ ИШИТР, e-mail: aym28@tpu.ru
[2]Томский политехнический университет, к.т.н., доцент ОИТ ИШИТР, e-mail: vikosi@tpu.ru

Аннотация

Разработка системы для мониторинга действий пользователей и обнаружения ошибок при актуализации данных в ИПК «Деканат». Решение включает серверную часть для хранения и анализа данных, а также инструменты для их интерпретации. В результате работы библиотека собирает, обрабатывает и отправляет информацию в систему мониторинга для дальнейшей визуализации данных.

Ключевые слова: телеметрия, системы мониторинга, OpenTelemetry, метрики, логи, трассировки, ELK.

Введение

При работе приложений возникают ситуации, когда необходимо вести сбор и мониторинг действий пользователя. Собираемая информация помогает технической поддержке быстрее выявлять и реагировать на поступающие обращения [1]. Для этого в единой информационной базе данных ТПУ были введены системные поля (created_by, date_created, modified_by, date_modified), значения которых проставляются в триггерах на актуализацию данных в таблицах. Подобные атрибуты не в полной мере могут помочь в отслеживании ошибок при редактировании данных. Например, значение в поле modified_by говорит о том, какой пользователь последним вносил изменения в таблицу, но невозможно определить, какие данные были исправлены. С целью решения такой проблемы предложено внедрить на примере ИПК «Деканат» систему для сбора и передачи информации о действиях пользователя, что является одним из видов телеметрии.

На сегодняшний день одним из готовых решений является инструмент OpenTelemetry, которое можно использовать в проекте без написания дополнительного кода [2], но по умолчанию собирает данные, связанные только с телеметрией серверов [3]. Кроме этого, подобные системы не могут обеспечить сбор специфичных данных, связанных с телеметрией пользователя, и большинство организаций создают собственные средства для реализации возникающих потребностей на базе OpenTelemetry [4, 5].

Исходя из этого возникает потребность в разработке собственного сервиса, который позволит систематически фиксировать ключевые аспекты активности пользователей. В частности, система будет регистрировать, кто выполняет операцию, когда это происходит, какая конкретно операция производится, а также какие данные вставляются, изменяются или удаляются и с какими объектами происходит взаимодействие. Также собираемая информация о действиях пользователей помогает технической поддержке быстрее выявлять и устранять ошибки, с которыми обращаются пользователи.

Целью данной работы является разработка и внедрение системы для сбора телеметрических данных о действиях пользователя в веб-приложении.

Телеметрические данные

В ходе анализа проблемы определены следующие телеметрические данные и цели их использования:

1. Время – время выполнения операции с данными.
2. Сервис – имя сервиса, в который внедряется библиотека.
3. Объект – название объекта, с которым взаимодействует пользователь.

4. Путь запроса – цепочка вызовов методов в сервисе при выполнении операции.
5. Операция – тип операции с данными (INSERT, DELETE, UPDATE).
6. Сообщение – текстовое представление объекта / сообщение об ошибке.
7. Сравнение объектов – измененные значения при редактировании данных.
8. Уровень логов – статус выполнения операции (INFO, ERROR).
9. Пользователь – имя пользователя, выполняющего действие.

Перечисленные выше телеметрические данные относятся к разным типам данных: логи, метрики, трассировки [6]. Большинство из них являются логами, так как фиксируются события, совершаемые пользователем в системе. К трассировкам можно отнести путь запроса, так как это ключевой параметр для данного типа. В качестве метрик можно выделить отдельные числовые показатели, например, количество операций за период времени, частота ошибок, среднее время выполнения операции.

После того как были выделены телеметрические данные, необходимо перейти к описанию алгоритма их сбора.

Алгоритм сбора телеметрических данных

Для создаваемой системы телеметрии разработан следующий алгоритм. Сначала на вход передается объект, с которым работает пользователь, логин пользователя и операция, которую необходимо выполнить и зафиксировать ее результат. Если операция завершена без ошибок, формируется строковое представление объекта, включая названия атрибутов с их значениями и идентификаторы связанных элементов в случае наличия вложенных объектов. Например, для добавления ИНН личности строковое представление объекта представлено на рисунке 1.

Рис. 1. Текстовое представление объекта

Затем извлекается название сервиса из конфигурационного файла приложения. Определяется, к какой таблице в базе данных относится объект, с которым работает пользователь. Если операция связана с изменением состояния объекта, фиксируются различия между предыдущим и текущим состоянием. Составляется путь запроса, исключая методы, связанные с телеметрией. Например, путь запроса при выполнении операции добавлении ИНН личности показан на рисунке 2.

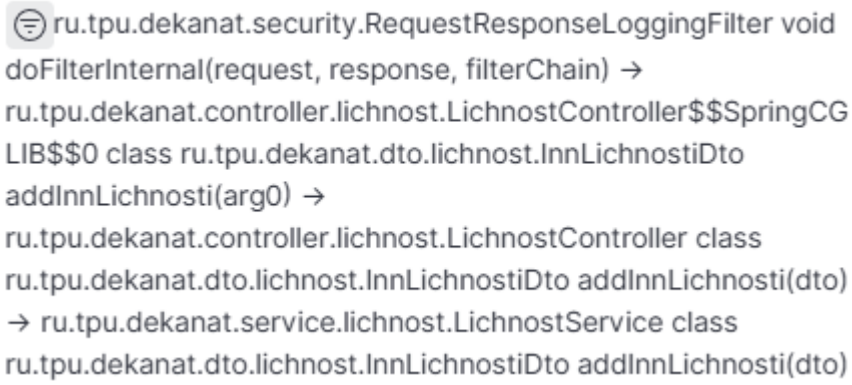

Рис. 2. Путь запроса

Следующим этапом создается запись лога с указанием: логина пользователя, типа выполняемой операции, названия таблицы, пути запроса, строкового представления объекта, даты выполнения запроса, названия исходного сервиса и уровень логирования INFO. Собранные данные отправляются в систему мониторинга. Например, собранные данные при добавлении ИНН личности отображены на рисунке 3.

call_chain.method	ru.tpu.dekanat.security.RequestResponseLoggingFilter void doFilterInternal(request, response, filterChain) → ru.tpu.dekanat.controller.lichnost.LichnostController$$SpringCGLIB$$0 class ru.tpu.dekanat.dto.lichnost.InnLichnostiDto addInnLichnosti(arg0) → ru.tpu.dekanat.controller.lichnost.LichnostController class ru.tpu.dekanat.dto.lichnost.InnLichnostiDto addInnLichnosti(dto) → ru.tpu.dekanat.service.lichnost.LichnostService class ru.tpu.dekanat.dto.lichnost.InnLichnostiDto addInnLichnosti(dto)
compare.objects	null
db_object	TPU.INN_LICHNOSTI_V
log.level	INFO
message	InnLichnosti{lichnostId=538431, inn='632569809870', dokumentId=null}
service.name	dekanat-service
sql.operation	INSERT
timestamp	2024-11-27T19:02:51.3555452
username	aym28

Рис. 3. Собранные данные при успешном выполнении

В случае если операция выбрасывает исключение, сообщение об ошибки форматируется для корректного отображения в логах. Затем происходит сбор такой же информации, как и в случае успешного выполнения, только уровень логирования ERROR. Например, в случае ошибки при добавлении ИНН личности будут оправлены данные в формате, представленном на рисунке 4.

db_object	TPU.INN_LICHNOSTI_V
log.level	ERROR
message	could not execute statement [ORA-20107: Изменение значения атрибута запрещено. Недопустимо изменять идентификатор документа для существующего ИНН. ORA-06512: на "TPU.TPU_ERRORS_P", line 50 ORA-06512: на "TPU.LICHNOST_PKG", line 763 ORA-06512: на "TPU.INN_LICHNOSTI_UPD_TRG", line 2 ORA-04088: ошибка во время выполнения триггера 'TPU.INN_LICHNOSTI_UPD_TRG'] [update tpu.inn_lichnosti_v set dokument_id=?,inn=? where lichnost_id=?]
service.name	dekanat-service
sql.operation	INSERT
timestamp	2024-11-25T15:38:50.4780992
username	aym28

Рис. 1. Собранные данные при неуспешном выполнении

Разработка системы

Реализация алгоритма выполнялась с использование Java Spring Boot и библиотеки OpenTelemetry, которая позволяет быстро и эффективно настроить телеметрию [7].

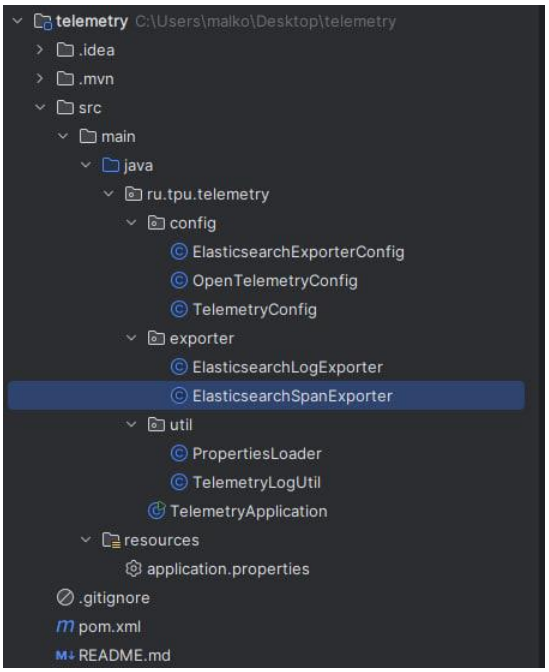

Рис. 2. Структура проекта

Система включает в себя следующие классы:

1. Конфигурационные классы, в которых происходит настройка телеметрии и указание системы мониторинга для отправки данных.

2. Классы, в которых определены методы для сбора и подготовки к отправке данных. Сбор необходимой информации осуществляется за счет рефлексии и завязан на аннотациях, используемых в Spring Framework.

Для использования реализованной библиотеки необходимо настроить приложение, указав в файле application.properties/.yaml следующие параметры:

1. package.name – главный пакет приложения, необходимый для удаления из пути запроса методов библиотеки.

2. service.name – название сервиса, из которого собираются телеметрические данные.

После того, как настроен проект и внедрена система для сбора и отправки телеметрических данных, можно визуализировать полученные данные с помощью системы мониторинга.

Визуализация данных

В качестве системы мониторинга выбран стек ELK (elasticsearch, logstash, kibana) [8]. Для примера визуализации данных выбрана операция вставки объекта ИНН личности.

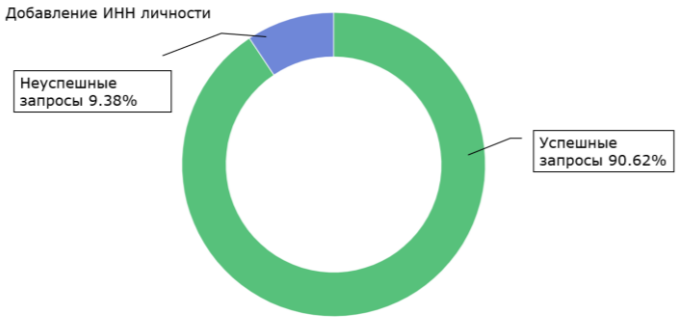

Рис. 3. Круговая диаграмма по уровню логирования

На рисунке 5 видно, что большинство запросов являются успешными, но также присутствуют и неуспешные запросы. Отсюда следует, что у некоторых пользователей возникает сложность с добавлением ИНН. Можно подробно посмотреть и проанализировать, у каких пользователей появлялись ошибки и какие данные они при этом передавали, затем можно перейти к устранению выявленных проблем.

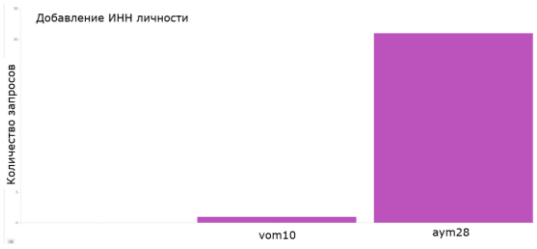

Рис. 4. Столбчатая диаграмма по количеству запросов от пользователей

На рисунке 6 видно, что чаще всего данную операцию выполняет пользователь с логином aym28, исходя из этого можно собирать статистические данные по объему и видам выполняемых работ. Подобная информация поможет технической поддержке эффективнее решать проблемы, возникающие по вине пользователей, выявлять проблемные места в приложении.

Заключение

Результатом работы является система для сбора и отправки телеметрических данных. На данный момент система внедрена в проект «Деканат». Разработанная библиотека позволяет обеспечить прозрачное и всестороннее понимание процессов, происходящих в приложении, что позволяет проводить детальный анализ и оперативно реагировать на изменения в поведении пользователей. Благодаря накопленной информации можно выявлять закономерности в использовании функционала, оптимизировать интерфейс и улучшать общую эффективность работы веб-приложения.

На данный момент система может внедряться в проекты, написанные с использованием фреймворка Spring, но в дальнейшем планируется сделать систему более универсальной.

Список использованных источников

1. Как Open Telemetry влияет на качество сервисов // True Engineering: сайт. – 2025. – [Электронный ресурс]. – URL: trueengineering.ru/blog/open_telemetry?ysclid=m8mxd2pr2v899609533 (дата обращения 20.03.2025).

2. Zero-code Instrumentation // OpenTelemetry: сайт. – 2025. – [Электронный ресурс]. – URL: opentelemetry.io/docs/zero-code/ (дата обращения 21.03.2025).

3. What is telemetry? // TechTarget: сайт. – 2025. – [Электронный ресурс]. – URL: techtarget.com/whatis/definition/telemetry (дата обращения 20.03.2025).

4. Как внедрить наблюдаемость в микросервисное приложение с помощью OpenTelemetry, Jaeger и Prometheus // Habr: сайт. – 2025. – [Электронный ресурс]. – URL: habr.com/ru/articles/865288/ (дата обращения 15.03.2025).

5. Создание системы анализа производительности приложений с помощью OpenTelemetry // Hemaks: сайт. – 2025. – [Электронный ресурс]. – URL: hemaks.org/ru/posts/building-an-application-performance-analysis-system-with-opentelemetry/ (дата обращения 16.03.2025).

6. Telemetry – definition & overview // Sumo Logic: сайт. – 2025. – [Электронный ресурс]. – URL: translated.turbopages.org/proxy_u/en-ru.ru.49365fe4-67e231d2-bb0e8c5a-74722d776562/https/www.sumologic.com/glossary/telemetry/ (дата обращения 14.03.2025).

7. Documentation // OpenTelemetry: сайт. – 2025. – [Электронный ресурс]. – URL: opentelemetry.io/docs/ (дата обращения 19.03.2025).

8. Elastic Documentation // Elasticsearch B.V.: сайт. – 2025. – URL: elastic.co/docs (дата обращения 23.03.2025).