УниверситетИТМО

Факультетпрограммнойинженерииикомпьютернойтехники

«Бизнес-логикапрограммныхсистем»

# Лабораторнаяработа№2

Вариант331318

Студенты:

Чжоу Хунсян

Группа:

Р33131

Преподаватель: КулиничЯрославВадимович

# Задание

Доработать приложение из лабораторной работы #1, реализовав в нём управление транзакциями и разграничение доступа к операциям бизнес-логики в соответствии с заданной политикой доступа.

**Управление транзакциями необходимо реализовать следующим образом:**

1. Переработать согласованные с преподавателем прецеденты (или по согласованию с ним разработать новые), объединив взаимозависимые операции в рамках транзакций.
2. Управление транзакциями необходимо реализовать с помощью Spring JTA.
3. В реализованных (или модифицированных) прецедентах необходимо использовать программное управление транзакциями.
4. В качестве менеджера транзакций необходимо использовать Bitronix.

**Разграничение доступа к операциям необходимо реализовать следующим образом:**

1. Разработать, специфицировать и согласовать с преподавателем набор привилегий, в соответствии с которыми будет разграничиваться доступ к операциям.
2. Специфицировать и согласовать с преподавателем набор ролей, осуществляющих доступ к операциям бизнес-логики приложения.
3. Реализовать разработанную модель разграничений доступа к операциям бизнес-логики на базе Spring Security. Информацию об учётных записах пользователей необходимо сохранять в реляционую базу данных, для аутентификации использовать JWT.

Проект

[GitHub](https://github.com/Tolia-GH/ITMO-Labs/tree/main/BusinessLogic/Lab2)

# Описание

Структура проэкта:

文本

中度可信度描述已自动生成

В проекте несколько ключевые классы, которые реализуются в ходе выполнении работы:

1.SecurityConfig

Здесь во-первых делает конфигуляцию при @EnableWebSecurity, @EnableGlobalMethodSecurity чтобы spring security применялась и основные функции как @PreAuthorize нормально работали в проекте.

Во-вторых здесь определяется некоторые url, к которым присоедить можно без аутентификации.

Во-третьих добавиться фильтр для проверки token.

2.TransactionManagementConfig:

Добавляет bitronix

2.DataSourceConfig:

Добавляет XA Trasaction

3.JwtUtils:

Инструмент для оформлении token

4.MyLoginUserDetailsService

Своя логика для аутентификации и авторизации

5.AccountsDetails extends UserDetails. Свой класс, где сохраняет все необходимые информации о пользователи.

6.JwtRequestsFilter

Проверка существование token и сравняет token с тем, что сохраняется в xml, чтобы делать авторизацию.

# Результат

文本

描述已自动生成

# Вывод

В ходе выполнении работы изучали процесс авторизации и аутентификации spring security и как писать свою логику авторизации и аутентификации в рамке spring security.

Научиться как использовать jwt token с spring security. Получил лучшее понятие о token.

Понять что такое транзакция и как её использовать при помощи Bitronix.