

Университет ИТМО

Факультет программной инженерии и компьютерной техники

«Бизнес-логика программных систем»

Лабораторная работа №2

Вариант 331318

Студенты:
Ляо Ихун

Группа:
Р33131

Преподаватель:
Кулинич Ярослав Вадимович

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 декларативное 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 учетных записях пользователей необходимо сохранять в файле XML, для аутентификации использовать JWT.

Правила выполнения работы:

1. Все изменения, внесённые в реализуемый бизнес-процесс, должны быть учтены в описывающей его модели, REST API и наборе скриптов для тестирования публичных интерфейсов модуля.
2. Доработанное приложение необходимо развернуть на сервере helios.

Проект

[Ссылка к проекту](#)

Описание

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

1.SecurityConfiguration:

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 присоединить можно без аутентификации.

Во-третьих добавиться фильтр для проверки token.

2.TransactionConfig:

Добавляет bitronix

3.JwtUtils:

Инструмент для оформления token

4.XmlUtils

Инструмент для чтения и записи информации о token пользователей в xml(MySecurityUsers.xml)

5.MyLoginUserDetailsService

Своя логика для аутентификации и авторизации

6.MyUserDetails

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

7.TokenFilter

Проверка существования token и сравнивает token с тем, что сохраняется в xml, чтобы делать авторизацию.

Результат

```
2023-05-09 13:27:28.234 INFO 13368 --- [main] edu.itmo.blps.DemoApplication : Starting DemoApplication using Java 17.0.6 on LAPTOP-I2LUFRCV with PID 13368
2023-05-09 13:27:28.236 INFO 13368 --- [main] edu.itmo.blps.DemoApplication : No active profile set, falling back to default profiles: default
2023-05-09 13:27:28.672 INFO 13368 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2023-05-09 13:27:28.722 INFO 13368 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 44 ms. Found 5 JPA repository in
2023-05-09 13:27:29.301 INFO 13368 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-05-09 13:27:29.308 INFO 13368 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-05-09 13:27:29.308 INFO 13368 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.56]
2023-05-09 13:27:29.392 INFO 13368 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-05-09 13:27:29.392 INFO 13368 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1124 ms
2023-05-09 13:27:29.465 INFO 13368 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-05-09 13:27:29.596 INFO 13368 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-05-09 13:27:29.627 INFO 13368 --- [main] bitronix.tm.BitronixTransactionManager : Bitronix Transaction Manager version 2.1.4
2023-05-09 13:27:29.629 WARN 13368 --- [main] bitronix.tm.Configuration : cannot get this JVM unique ID. Make sure it is configured and you only use
2023-05-09 13:27:29.629 INFO 13368 --- [main] bitronix.tm.Configuration : JVM unique ID: <172.23.0.1>
2023-05-09 13:27:29.634 WARN 13368 --- [main] bitronix.tm.journal.DiskJournal : active log file is unclear, did you call BitronixTransactionManager.shutdown
2023-05-09 13:27:29.654 INFO 13368 --- [main] bitronix.tm.recovery.Recoverer : recovery committed 0 dangling transaction(s) and rolled back 0 aborted tran
2023-05-09 13:27:29.733 INFO 13368 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHN000204: Processing PersistenceUnitInfo [name: default]
2023-05-09 13:27:29.775 INFO 13368 --- [main] org.hibernate.Version : HHN000412: Hibernate ORM core version 5.6.3.Final
2023-05-09 13:27:29.905 INFO 13368 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2023-05-09 13:27:30.005 INFO 13368 --- [main] org.hibernate.dialect.Dialect : HHN000400: Using dialect: org.hibernate.dialect.PostgreSQLDialect
2023-05-09 13:27:30.141 INFO 13368 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHN000490: Using JtaPlatform implementation: [org.springframework.boot.orm
2023-05-09 13:27:30.480 INFO 13368 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2023-05-09 13:27:30.571 WARN 13368 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries
2023-05-09 13:27:31.042 INFO 13368 --- [main] o.s.s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.context.request
2023-05-09 13:27:31.579 INFO 13368 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-05-09 13:27:31.586 INFO 13368 --- [main] edu.itmo.blps.DemoApplication : Started DemoApplication in 3.619 seconds (JVM running for 4.105)
```

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.