LESSON 22

SPRING BOOT: SECURITY

SPRING BOOT: SECURITY

- Введение
- Аутентификация
- Авторизация
- OAuth 2.0
- Как происходит процесс

INTRODUCTION

Spring Security — среда для аутентификации и авторизации пользователей. Фреймворк применяется для защиты приложений на Spring. В нем представлены базовые инструменты безопасности, которые без труда расширяются для решения разных задач.

Самым фундаментальным объектом является **SecurityContextHolder**. В нем хранится информация о текущем контексте безопасности приложения, который включает в себя подробную информацию о пользователе, работающим с приложением.

AUTHENTICATION

Аутентифика́ция — процедура проверки подлинности, например: проверка подлинности пользователя путём сравнения введённого им пароля с паролем, сохранённым в базе данных пользовательских логинов.

AUTHORIZATION

Авторизация - это предоставление определённому лицу или группе лиц прав на выполнение определённых действий, а также процесс проверки данных прав при попытке выполнения этих действий.
В более простых приложениях аутентификации может быть достаточно: как только пользователь проходит аутентификацию, он может получить доступ ко всем частям приложения.

Но у большинства приложений есть концепция разрешений (или ролей). Представьте: клиенты, имеющие доступ к общедоступному интерфейсу вашего интернет-магазина, и администраторы, имеющие доступ к отдельной области администрирования.

Оба типа пользователей должны войти в систему, но сам факт аутентификации ничего не говорит о том, что им разрешено делать в вашей системе. Следовательно, вам также необходимо проверить разрешения аутентифицированного пользователя, т.е. вам необходимо авторизовать пользователя.

OAUTH2

OAuth2 — это протокол авторизации, который позволяет предоставить третьей стороне ограниченный доступ к защищенным ресурсам пользователя без необходимости передавать ей (третьей стороне) логин и пароль.

OAuth2 определяет 4 роли:

- Владелец ресурса
- Ресурсный сервер
- Сервер авторизации
- Клиент (приложение)

EXPECTED PROTOCOL FLOW

Упрощенный поток:

1. Запрос авторизации отправляется от клиента к серверу (действующему как владелец ресурса) с использованием предоставления авторизации пароля.

2. Токен доступа возвращается клиенту (вместе с токеном обновления)

3. Затем токен доступа отправляется от клиента к серверу (действующему в качестве сервера ресурсов) при каждом запросе на доступ к защищенному ресурсу.

4. Сервер отвечает требуемыми защищенными ресурсами

EXPECTED PROTOCOL FLOW