**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по лабораторным и практическим работам**

**МДК 11.01**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Студент | ИСПП-35 | |  |  | |  | | --- | | *Пожидаев Г.С.* | |
|  | (Группа) | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Преподаватель | | |  |  | *Маломан Ю. С.* |
|  | |  | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2024

# **Практическая работа №3**

**Разграничение прав доступа на уровне REST API**

1. **Цель работы** 
   1. Научиться выполнять разработку web-API для доступа к БД.

Научиться работать с JWT.

1. **Контрольные вопросы**

Вопрос: Атрибуты методов REST для настройки доступа

При разработке REST API для управления доступом можно использовать следующие атрибуты:

- `@GetMapping`: Указывает HTTP-метод GET для получения ресурса.

- `@PostMapping`: Указывает HTTP-метод POST для отправки нового ресурса.

- `@PutMapping`: Указывает HTTP-метод PUT для обновления существующего ресурса.

- `@DeleteMapping`: Указывает HTTP-метод DELETE для удаления ресурса.

- `@PreAuthorize`: Позволяет применять аннотации Spring Security для проверки прав доступа.

- `@RolesAllowed`: Ограничивает доступ только определенным ролям.

- `@AuthenticationPrincipal`: Предоставляет текущего пользователя для проверки прав.

Для неавторизованных пользователей можно использовать:

- `@AnonymousAccessController`

- `@Secured({"ROLE\_ANONYMOUS"})`

Вопрос: Использование JWT

JWT (JSON Web Token) используется для передачи информации в безопасном и зашифрованном формате между клиентом и сервером. Он позволяет:

- Автоматически аутентифицировать пользователя

- Хранить информацию о состоянии сессии

- Обеспечить безопасное хранение чувствительных данных

- Улучшить производительность приложения за счет минимального количества запросов к серверу

Вопрос: Отличие между авторизацией с использованием cookie и JWT

Авторизация с использованием cookie:

- Более простая реализация

- Меньше нагрузки на сеть

- Потенциально более уязвима к атакам типа CSRF

Авторизация с использованием JWT:

- Более безопасна из-за отсутствия прямого обращения к серверу

- Не зависит от сессий на стороне сервера

- Легче масштабировать приложение

- Более сложная реализация

Вопрос: Настройки при создании токена

При создании токена JWT можно указать следующие настройки:

- Выбор алгоритма шифрования (HS256, RS256 и др.)

- Срок действия токена

- Ключи для шифрования и расшифровки

- Дополнительные claims (поля в токене)

- Параметры времени создания и истечения срока действия

Вопрос: Почему использовать метод POST для авторизации

Использование метода POST для авторизации рекомендуется по следующим причинам:

- Метод POST предназначен для отправки данных на сервер

- Он менее подвержен атакам типа CSRF по сравнению с GET-запросами

- Позволяет передавать чувствительные данные безопасно

- Улучшает защиту от повторных запросов (как физических, так и социальных)

- Поддерживает более сложную логику обработки входных данных

Использование POST обеспечивает более надежную и безопасную реализацию механизма авторизации.

1. **Вывод**
   1. В ходе лабораторной работы мы научились выполнять разработку web-API для доступа к БД.

Научились работать с JWT.