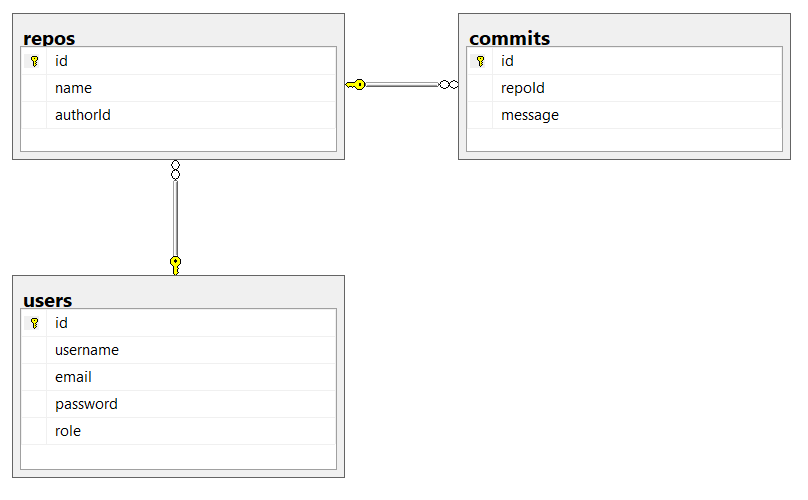
Лабораторная работа 14

ПвИ

**Задание 01**

1. Разработайте приложение **14-01**, представляющее собой EXPRESS-HTTP-сервер, прослушивающий порт ***3000***.
2. Приложение **14-01** предназначено для демонстрации управления доступом на основе ролей.
3. В приложении **14-01** должны быть определены 3 сущности (User, Repo и Commit). В качестве СУБД использовать MS SQL Server или MySQL. Для работы с базой данных использовать пакет **sequelize** или **prisma**.



1. Приложение **14-01** должно реагировать на запросы к следующим ресурсам (использовать в качестве основы приложение **12-02**).

|  |  |  |
| --- | --- | --- |
| Метод  запроса | URI | Назначение |
| GET | /login | То же, что и в приложении **12-02.** |
| POST | /login |
| GET | /register |
| POST | /register |
| GET | /logout |
| GET | api/ability | набор привилегий текущего пользователя в формате json |
| GET | api/user | получить список всех пользователей (без пароля) в json-формате |
| GET | api/user/:id | id – идентификатор пользователя, получить информацию о пользователе (без пароля) с идентификатором id |
| GET | api/repos | получить список всех репозиториев в json-формате |
| GET | api/repos/:id | id – идентификатор репозитория, получить информацию об указанном репозитории |
| POST | api/repos | добавить новый репозиторий, данные в json-формате |
| PUT | api/repos/:id | id – идентификатор репозитория, корректировать информацию об указанном репозитории, данные в json-формате |
| DELETE | api/repos/:id | id – идентификатор репозитория, удалить указанный репозиторий |
| GET | api/repos/:id/commits | id – идентификатор репозитория, получить список всех коммитов к указанному репозиторию в json-формате |
| GET | api/repos/:id/commits/:commitId | id – идентификатор репозитория, commitId – идентификатор коммита, получить информацию об указанном коммите |
| POST | api/repos/:id/commits/ | добавить новый коммит к указанному репозиторию, данные в json-формате |
| PUT | api/repos/:id/commits/:commitId | id – идентификатор репозитория, commitId – идентификатор коммита, корректировать информацию об указанном коммите, данные в json-формате |
| DELETE | api/repos/:id/commits/:commitId | id – идентификатор репозитория, commitId – идентификатор коммита, удалить указанный коммит |
| \* | Остальные URI | Вернуть ответ со статусом 404 или 405 (если неподдерживаемый метод) |

1. Добавить в приложение обработку ошибок. В случае возникновения ошибки возвращать клиенту сообщение с соответствующим статус кодом и пояснением.
2. В приложении **14-01** должна быть поддержка 3 ролей: гость, зарегистрированный пользователь и администратор.
3. Приложение **14-01** должно предоставлять доступ к API по приведенным правилам. Для управления правами использовать пакет **casl**.

|  |  |
| --- | --- |
| Роль | Привилегии |
| Гость | * просматривать свои привилегии (/ability), коммиты и репозитории |
| Зарегистрированный пользователь | * просматривать свои привилегии (/ability), информацию о себе, коммиты и репозитории * создавать репозитории * создавать коммиты к своим репозиториям * редактировать свои репозитории и коммиты |
| Администратор | * просматривать свои привилегии (/ability), информацию о всех пользователях, коммиты и репозитории * редактировать любые репозитории и коммиты * удалять любые репозитории и коммиты |

1. Определение привилегий осуществлять в глобальном middleware.
2. Тестирование выполнить с помощью браузера и Postman.

**Задание 02** Ответьте на следующие вопросы

1. Поясните понятия «авторизация», «роль», «привилегия», «ACL».

**Авторизация** - процесс, определяющий, что пользователь имеет право делать после успешного входа в систему. **Роль** - набор правил, определяющих доступы для группы пользователей. **Привилегия** - конкретное право на выполнение действия или доступ к ресурсу. **ACL (список контроля доступа)** - метод, используемый для определения того, кто может или не может использовать определенные ресурсы системы.

1. Что такое casl?

**CASL** - фреймворк для управления доступом, позволяющий определять правила доступа в приложениях, используя удобный и гибкий синтаксис.

1. Перечислите известные модели управления доступом.

DAC (Дискреционное управление доступом),

MAC (Принудительное управление доступом),

RBAC (Управление доступом на основе ролей),

ABAC (Управление доступом на основе атрибутов).