## Лабораторная работа №3.

На основе лабораторной работы №2 реализовать механизм ролевой политики.

Требования к выполнению работы.

1. Создать от главной ветки новую с наименованием «lb3».
2. Переключиться на новую ветку.
3. Опубликовать новую ветку.
4. Создать 4 миграции создающих таблицы:
   1. Roles – таблица со списком ролей
   2. Permissions – таблица со списком разрешений
   3. UsersAndRoles – таблица связывающая пользователя с ролями
   4. RolesAndPermissions – таблица связывающая роли с разрешениями
5. Обеспечить функциональные возможности:
   1. Пользователь может обладать несколькими ролями
   2. Роли могут обладать несколькими разрешениями
   3. Роли имеют возможность мягкого удаления
   4. Разрешения имеют возможность мягкого удаления
6. Обеспечить полноту данных.
   1. Роли имеют:
      1. Наименование
      2. Описание
      3. Шифр
      4. Служебные поля

Наименование и шифр роли обязательны и уникальны.

* 1. Разрешения имеют:
     1. Наименование
     2. Описание
     3. Шифр
     4. Служебные поля

Наименование и шифр разрешения обязательны и уникальны.

* 1. Роли пользователей имеют:
     1. Ссылку на пользователя
     2. Ссылку на роль
     3. Служебные поля
  2. Разрешения ролей имеют:
     1. Ссылку на роль
     2. Ссылку на разрешение
     3. Служебные поля

1. Выполнить созданные миграции
2. Создать модель Role соблюдая требования стилизации кода.
3. Создать DTO класс роли.
4. Создать DTO класс коллекции ролей.
5. Создать классы запросов на создание и изменение роли. Запросы выполняют проверку на уникальность данных. Выполнить запросы могут только авторизованные пользователи. Проверка авторизации обеспечивается на уровне класса запроса.
6. В созданные классы запросов добавить метод возвращающий экземпляр соответствующего DTO ресурса.
7. Создать 7 маршрутов для сущности согласно требованиям маршрутов управления ролевой политикой.
8. Создать класс контроллер для управления сущностью с методами согласно маршрутам ролей.
9. Все методы принимают в качестве аргумента экземпляр соответствующего класса запроса.
10. Выполнить этапы 8-15 для сущностей:
    1. Permissions
    2. UsersAndRoles
    3. RolesAndPermissions
    4. Users
11. В модель пользователей добавить метод roles().
12. Добавленный метод возвращает коллекцию имеющихся у пользователя ролей.
13. Добавить коллекцию roles в класс DTO ресурса пользователя возвращаемый при запросе пользователя.
14. В модель ролей добавить метод permissions().
15. Добавленный метод возвращает коллекцию имеющихся у роли разрешений.
16. Добавить коллекцию permissions в класс DTO ресурса пользователя возвращаемый при запросе роли.
17. Реализовать создание и удаление связки пользователя с ролью.
18. Реализовать создание и удаление связки роли с разрешением.
19. Создать сид со списком ролей:
    1. Admin
    2. User
    3. Guest;
20. Создать сид со списком разрешений: [entity] – наименование модели [user/role/permission].
    1. get-list-[entity]
    2. read-[entity]
    3. create-[entity]
    4. update-[entity]
    5. delete-[entity]
    6. restore-[entity]
21. Выполнить сиды.
22. Настроить создать связки ролей и разрешений:
    1. Админ может все
    2. Пользователь может получить список пользователей, читать и обновлять свои данные
    3. Гость может только получить список пользователей.
23. Реализовать проверку доступа пользователей к выполняемым операциям.
24. В случае отсутствия у пользователя разрешения на выполняемую операцию, приложение возвращает статус 403 с сообщением, которое содержит в себе наименование необходимого разрешения.
25. Зафиксировать изменения и опубликовать их в ветку «lb3».

Требования к маршрутам управления пользователями:

|  |  |  |  |
| --- | --- | --- | --- |
| Операция | URL | Метод | Открытость |
| Получение списка пользователей | api/ref/user/ | GET | Только авторизованные |
| Получение ролей пользователя | api/ref/user/{id}/role | GET | Только авторизованные |
| Присвоение ролей пользователю | api/ref/user/{id}/role | POST | Только авторизованные |
| Жесткое удаление роли у пользователя | api/ref/user/{id}/role/{id} | DELETE | Только авторизованные |
| Мягкое удаление роли у пользователя | api/ref/user/{id}/role/{id}/soft | DELETE | Только авторизованные |
| Восстановление мягко удаленной роли у пользователя | api/ref/user/{id}/role/{id}/restore | POST | Только авторизованные |

Требования к маршрутам управления ролевой политикой (Роли):

|  |  |  |  |
| --- | --- | --- | --- |
| Операция | URL | Метод | Открытость |
| Получение списка ролей | api/ref/policy/role | GET | Только авторизованные |
| Получение конкретной роли | api/ref/policy/role/{id} | GET | Только авторизованные |
| Создание роли | api/ ref/policy/role | POST | Только авторизованные |
| Обновление роли | api/ ref/policy/role/{id} | PUT | Только авторизованные |
| Жесткое удаление роли | api/ ref/policy/role/{id} | DELETE |  |
| Мягкое удаление роли | api/ ref/policy/role/{id}/soft | DELETE | Только авторизованные |
| Восстановление мягко удаленной роли | api/ ref/policy/role/{id}/restore | POST | Только авторизованные |

Требования к маршрутам управления ролевой политикой (Разрешения):

|  |  |  |  |
| --- | --- | --- | --- |
| Операция | URL | Метод | Открытость |
| Получение списка разрешений | api/ref/policy/permission | GET | Только авторизованные |
| Получение конкретного разрешения | api/ref/policy/ permission/{id} | GET | Только авторизованные |
| Создание разрешения | api/ ref/policy/permission | POST | Только авторизованные |
| Обновление разрешения | api/ ref/policy/permission{id} | PUT | Только авторизованные |
| Жесткое удаление разрешения | api/ ref/policy/permission/{id} | DELETE |  |
| Мягкое удаление разрешения | api/ ref/policy/permission/{id}/soft | DELETE | Только авторизованные |
| Восстановление мягко удаленного разрешения | api/ ref/policy/permission/{id}/restore | POST | Только авторизованные |

Требования к служебным полям:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | Тип | NOT NULL | Comment | Default |
| created\_at | Timestamp | True | Время создания записи | SYSDATE |
| created\_by | Integer | True | Идентификатор пользователя создавшего запись |  |
| deleted\_at | Timestamp | False | Время мягкого удаления записи |  |
| deleted\_by | Integer | False | Идентификатор пользователя удалившего запись |  |

Критерии приемки:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Критерий приемки | Обязательность | Баллы за выполнение критерия |
| 1 | Все требования выполнены. Все требуемые методы реализованы согласно заданию, работают корректно и без ошибок. | Да | 40 |
| 2 | Код чистый, типизированный и оптимизированный. Файлы не содержат закомментированные участки кода. Наименования используемых методов и переменных интуитивно понятны. Все используемые методы дополнены комментариями. Выполняется принцип SOLID. Файлы не содержат не используемых методов и переменных. | Да | 15 |
| 3 | Используемые маршруты сгруппированы по префиксу | Да | 3 |
| 4 | Реализована проверка контроля доступа к методам | Да | 20 |
| 5 | Реализована оптимизированная система управления связками пользователей с ролями и ролей с пользователями | Да | 10 |
| 6 | Все методы вызываются соответствующими методами запросов | Да | 4 |
| 7 | Все методы возвращают корректные статусы | Да | 2 |
| 8 | Статус 403 возвращает сообщение с необходимым разрешением | Нет | 8 |
| 9 | Добавлены сиды для связок ролей и разрешений | Нет | 6 |
| 10 | Добавлены сиды для связок пользователей с ролями | Нет | 6 |
| 11 | Данные из методов возвращаются пользователям в формате JSON. | Да | 4 |
| 12 | Код покрыт тестами. | Нет | 8 |
| 13 | Выполнение всех требований в работе с git | Нет | 4 |
| Всего | | | 130 |
| Необходимо для сдачи лабораторной | | | 90 |

Штрафы.

|  |  |  |
| --- | --- | --- |
| № | Штраф | Баллы за нарушение |
| 1 | Метод не реализован. | 5 |
| 2 | Метод работает не корректно или не работает. | 3 |
| 3 | Метод вызывается http запросом неверного типа. | 1 |
| 4 | Код грязный. | 2 |
| 5 | Нарушена или не указана типизация переменных или методов (обсуждаемо). | 1 |
| 6 | Код не оптимизирован (обсуждаемо). | 1 |
| 7 | Изменяемые содержат закомментированные участки кода. | 2 |
| 8 | Наименование используемых методов и переменных не интуитивно понятны (Обсуждаемо). | 1 |
| 9 | Нарушение требований CodeStyle. | 1 |
| 10 | Нарушение требований SOLID. | 2 |
| 11 | Миграция или сид не соответствует таблице или модели | 2 |
| 12 | Класс запроса не содержит валидации данных. | 2 |
| 13 | Пользователь может видеть служебные поля или пароли | 10 |
| 14 | Неправильная работа с git. | 2 |
| 15 | Файлы содержат не используемые методы или переменные. | 1 |
| 16 | Реализованные методы не содержат комментариев. | 1 |

Штрафы назначаются за каждое обнаруженное нарушение.

Например, если код содержит 2 метода без комментария, то штраф будет в двойном размере.