Лабораторная работа 22

ПСКП

ПОИТ-3

**Задание 01**

1. Разработайте приложение **22-01**, представляющее собой EXPRESS-HTTP-сервер, прослушивающий порт ***3000***.
2. Приложение **22-01** предназначено для предоставления аутентифицированного http-доступа к ресурсу.
3. Приложение **22-01** должно обеспечивать FORMS-аутентификацию на основе сессий с помощью пакета **passport**.
4. Приложение **22-01** должно реагировать на запросы к следующим ресурсам.

|  |  |
| --- | --- |
| GET /login | Вернуть форму для ввода имени пользователя и пароля |
| POST /login | Производить аутентификацию  В случае успешной аутентификации переадресовывать на /resource, иначе – на /login |
| GET /logout | Отключить аутентифицированный доступ к ресурсу |
| GET /resource | Если пользователь аутентифицирован, то отправить сообщение RESOURCE и информацию об аутентифицированном пользователе  При попытке неаутентифицированного доступа выполнить переадресацию на /login |
| Остальные URI | Вернуть ответ со статусом 404 |

1. При попытке неаутентифицированного доступа возвращать ошибку с 401 статусом и пояснительным сообщением.
2. Список разрешенных данных для аутентификации (имя, пароль и т.д.) в приложении должен быть представлен в виде json-объекта, доступ к которому осуществляется с помощью функции **require**.
3. Тестирование выполнить с помощью браузера и Postman’а.

**Задание 02**

1. Разработайте приложение **22-02**, представляющее собой EXPRESS-HTTP-сервер, прослушивающий порт ***3000***. Приложение **22-02** должно использовать FORMS- аутентификацию на основе jwt-токенов (access и refresh) с помощью пакета **jsonwebtoken**.
2. Приложение **22-02** должно реагировать на запросы к следующим ресурсам.

|  |  |
| --- | --- |
| GET /login | Вернуть форму для ввода имени пользователя и пароля |
| POST /login | Производить аутентификацию  В случае успешной аутентификации возвращать пару токенов (access и refresh) и переадресовывать на /resource, иначе – на /login |
| GET /refresh-token | Принимать refresh-токен, проверять его.  Если токен валидный, то формировать новую пару access и refresh токена, помещать старый refresh-токен в «черный» список.  Если токен не валидный, то возвращать статус 401 |
| GET /logout | Отключить аутентифицированный доступ к ресурсу |
| GET /resource | Если токен валидный, то отправить сообщение RESOURCE и информацию об аутентифицированном пользователе  При попытке неаутентифицированного доступа выполнить переадресацию на GET /login |
| Остальные URI | Вернуть ответ со статусом 404 |

1. При попытке неаутентифицированного доступа возвращать ошибку с 401 статусом и пояснительным сообщением.
2. Список разрешенных данных для аутентификации (имя, пароль и т.д.) хранить в базе данных.Для работы с ней использовать пакет **sequelize**.
3. Установить время жизни access-токена 10 минут, а refresh-токена – 24 часа.
4. Access-токен сохранять в cookie. При установке указать флаги HttpOnly=true (для защиты от XSS-атак), SameSite=Strict (для защиты от CSRF-атак).
5. Refresh-токен хранить тоже в cookie. Дополнительно использовать флаг path.
6. Для «черного» списка refresh-токенов использовать Redis.
7. Добавить возможность регистрации.
8. Тестирование выполнить с помощью браузера и Postman’а.

**Задание 03**

1. Опишите схему FORMS-аутентификации с использованием сессий.
2. Опишите схему FORMS-аутентификации с использованием токенов.