Лабораторная работа 18

ПСКП

ПОИТ-3

**Задание 01**

1. Разработайте приложение **18-01**, представляющее собой EXPRESS-HTTP-сервер, прослушивающий порт ***3000***.
2. Приложение **18-01** предназначено для предоставления аутентифицированного доступа к ресурсу.
3. Приложение **18-01** должно реализовывать стратегию аутентификации через социальные сети, то есть OAuth с помощью пакета **passport** (провайдер любой: *Google*, *VK*, *Twitter*, *Facebook*, *Github* и т.д.).
4. Приложение **18-01** должно реагировать на запросы к следующим ресурсам.

|  |  |
| --- | --- |
| GET /login | Вернуть страницу с кнопкой для аутентификации через выбранную соцсеть. |
| GET /auth/{provider\_name}/ | Производить аутентификацию. Перенаправлять пользователя на на сайт выбранного провайдера для аутентификации. По завершении провайдер должен переадресовать пользователя обратно в приложение по адресу /auth/{provider\_name}/callback (redirect URL). |
| GET /auth/{provider\_name}/callback | Переадресовать пользователя на /resource в случае, если доступ был одобрен, в противном случае – на /login. |
| GET /logout | Отключить аутентифицированный доступ к ресурсу. |
| GET /resource | Если пользователь аутентифицирован, то отправить сообщение RESOURCE и информацию об аутентифицированном пользователе (идентификатор, имя пользователя и др.).  При попытке неаутентифицированного доступа выполнить переадресацию на /login. |
| Остальные URI | Сообщение со статусом 404 |

1. Тестирование выполнить с помощью браузера.

**Задание 02** Ответьте на следующие вопросы

1. Опишите общую схему OAuth-аутентификации.

OAuth (открытый стандарт авторизации) – это протокол, который позволяет пользователям предоставлять ограниченный доступ к своим данным третьим сторонам без необходимости раскрытия своих учетных данных. Он широко используется во множестве веб-приложений и API для обеспечения безопасного обмена информацией между различными сервисами.

Общая схема OAuth-аутентификации включает в себя следующие участники:

1. Владелец ресурса (пользователь): это человек, который имеет учетную запись на сервисе, позволяющем доступ к его ресурсам (например, социальная сеть или почтовый провайдер).
2. Клиент: это приложение или сервис, запрашивающее доступ к ресурсам пользователя. Клиент может быть мобильным приложением, веб-приложением или другим сервисом, который хочет получить доступ к определенным данным пользователя.
3. Сервер авторизации: это сервис, ответственный за аутентификацию пользователя и выдачу временных авторизационных токенов клиенту. Примером сервера авторизации может быть сервис авторизации Google или Facebook.
4. Сервер ресурсов: это сервис, владеющий ресурсами пользователя. Когда клиент получает авторизационный токен, он может использовать его для получения доступа к данным пользователя на сервере ресурсов.

Шаги процесса OAuth-аутентификации обычно выглядят следующим образом:

1. Клиент регистрируется на сервере авторизации и получает идентификатор клиента и секретный ключ.
2. Клиент направляет пользователя на сервер авторизации для аутентификации. Это может быть реализовано путем перенаправления пользователя на страницу авторизации или показа встроенной формы входа.
3. Пользователь вводит свои учетные данные на сервере авторизации.
4. Сервер авторизации проверяет учетные данные пользователя и запрашивает у него разрешение на доступ к его ресурсам.
5. Если пользователь дает согласие, сервер авторизации генерирует авторизационный код или токен и перенаправляет пользователя обратно на клиента, предоставляя авторизационный код в качестве параметра URL или отправляя его веб-службе клиента напрямую.
6. Клиент использует полученный авторизационный код для обмена на авторизационный токен у сервера авторизации.
7. После получения авторизационного токена клиент может использовать его для доступа к ресурсам пользователя на сервере ресурсов. Клиент отправляет токен в каждом запросе к серверу ресурсов для проверки и авторизации доступа.

Таким образом, OAuth позволяет пользователям контролировать доступ к своим данным и позволяет клиентам получать временные авторизационные токены для доступа к этим данным без необходимости раскрытия учетных данных пользователей. Это делает процесс аутентификации и авторизации более безопасным и удобным для всех участников.