Процесс аутентификации

Аутентификация основана протоколе OAuth 2.0. Данный протокол предусматривает наличие четырёх ролей:

1. Владелец ресурса (Resource Owner) – сущность или пользователь получающий доступ к защищённым ресурсам.
2. Ресурсный сервер (Resource Server) – сервер, который предоставляет доступ к защищённым ресурсам. Данный сервер должен иметь возможность проверять токен выданный сервером авторизации и если токен корректный – отвечать на запрос клиентского приложения.
3. Клиентское приложение (англ. Client Applications) – приложение получающее данные от защищённого ресурсного сервера, данное приложение также запрашивает токен у сервера авторизации, который идентифицирует владельца ресурса.
4. Сервер авторизации (Authorization Server) – сервер, который управляет авторизацией выдавая токены доступа, позволяющие идентифицировать владельца ресурса при запросе данных с ресурсного сервера.



В данном приложении в качестве сервера авторизации выступают

Уровень доступа к данным.

Для большинства современны web-приложений основной функцией является предоставления доступа к данным. Интерфейс типичного приложения водится к предоставлению пользователю следующих типичных процедур для манипулирования данными: добавление (create), чтение (read), обновление (update), удаление (delete). Данный набор операций известен как CRUD-операции. Данная концепция используется не только серверами управления баз данных (СУБД), но и основным протоколом передачи данных HTTP. В таблице 1 представлены соответствия между CRUD-операциями, операторами языка SQL, типами HTTP запросов (HTTP-verbs).

Таблица 1.

|  |  |  |
| --- | --- | --- |
| Операция | Оператор языка SQL | Операция в протоколе HTTP |
| Создание (create) | INSERT | POST |
| Чтение (read) | SELECT | GET |
| Редактирование (update) | UPDATE | PUT или PATCH |
| Удаление (delete) | DELETE | DELETE |

Схожий подход используется также в нерялиционных базах (документо-ориентированные СУБД, графовых СУБД и т.д.), при этом в серверах баз данных использующихся в высоконагруженных системах широкое распространение получил подход к увеличению производительности и отказоустойчивости основанный на отказе от предоставления операции изменения. Для имитации данной операции используется комбинация трёх оставшихся команд: чтение, удаление, вставка. При в отличии от традиционной операции изменения такой подход изменяет первичный ключ (индекс в таблице) и временную метку. Такой подход используется наиболее распространённом NoSQL (not only SQL) решении в СУБД Cassandra. Данная СУБД используется во множестве высоко нагруженных систем, наиболее известной из которых является сервис Twitter, который отображает информацию опубликованную пользователем в хронологическом порядке. В силу вышеупомянутого подхода к оптимизации производительности можно с уверенностью ответить на вопрос, который беспокоит сотни миллионов пользователей Twitter-а: когда появиться функция редактирования сообщений? Ответ: она не появится до тех пор, пока Twitter не откажется от подхода оптимизации производительности, основанном на замене операции редактирования операциями чтения, удаления записи.