JSON Web Token

Смирнов Александр
Тухватуллина Гузель
гр. 5040102/50201

Содержание


Что такое JWT?
Как передавать токены
Виды JWT
Структура JWT
Валидация токенов
Black-list токенов
Использование JWT
Обновление JWT
Плюсы и минусы

Что такое JWT?

JWT (Json Web Token) — ключ аутентификации пользователя. Используется для запросов к защищенным методам API.

Для чего нужны JWT: чтобы не передавать учетные данные пользователя с каждым запросом к серверу.

Роль пароля и токенов

- Пароль используется только один раз — при аутентификации.
- После успешного входа сервер выдает пару токенов: Access и Refresh.
- Клиент не хранит пароль, только токены.
- Сервер не хранит пароль, только его криптографический хэш.
- Повторный ввод пароля требуется только в особых случаях:
  - Истек срок жизни обоих токенов.
  - Пользователь вышел из системы (logout).
  - Пароль был изменен или учетная запись сброшена.

Почему JWT? Сравнение с серверными сессиями

Ограничения серверных сессий:

- Привязка к домену: Cookie с session_id не пересылаются между разными доменами (auth.example.com, api.example.com).
- Проблемы масштабирования: Требуется общая БД (session store) для всех серверов.
- Зависимость от клиента: Удобны только для браузеров, которые автоматически отправляют cookie. Неудобны для мобильных приложений и API-клиентов.

Как передавать токены

от сервера к клиенту:

1. в теле запроса
2. либо используя заголовок «Set-Cookie»:

```
Set-Cookie: accessToken=<jwt>; HttpOnly; Sequre; SameSite=Strict;
```

Как передавать токены

от клиента к серверу:

1. в заголовке запроса «Authorization» с добавлением слова Bearer

```
Authorization: Bearer <jwt>
```

2. либо используя заголовок «Cookie»

```
Cookie: accessToken=<jwt>
```

Виды JWT

access token

проверяется при каждом обращении к защищенному API

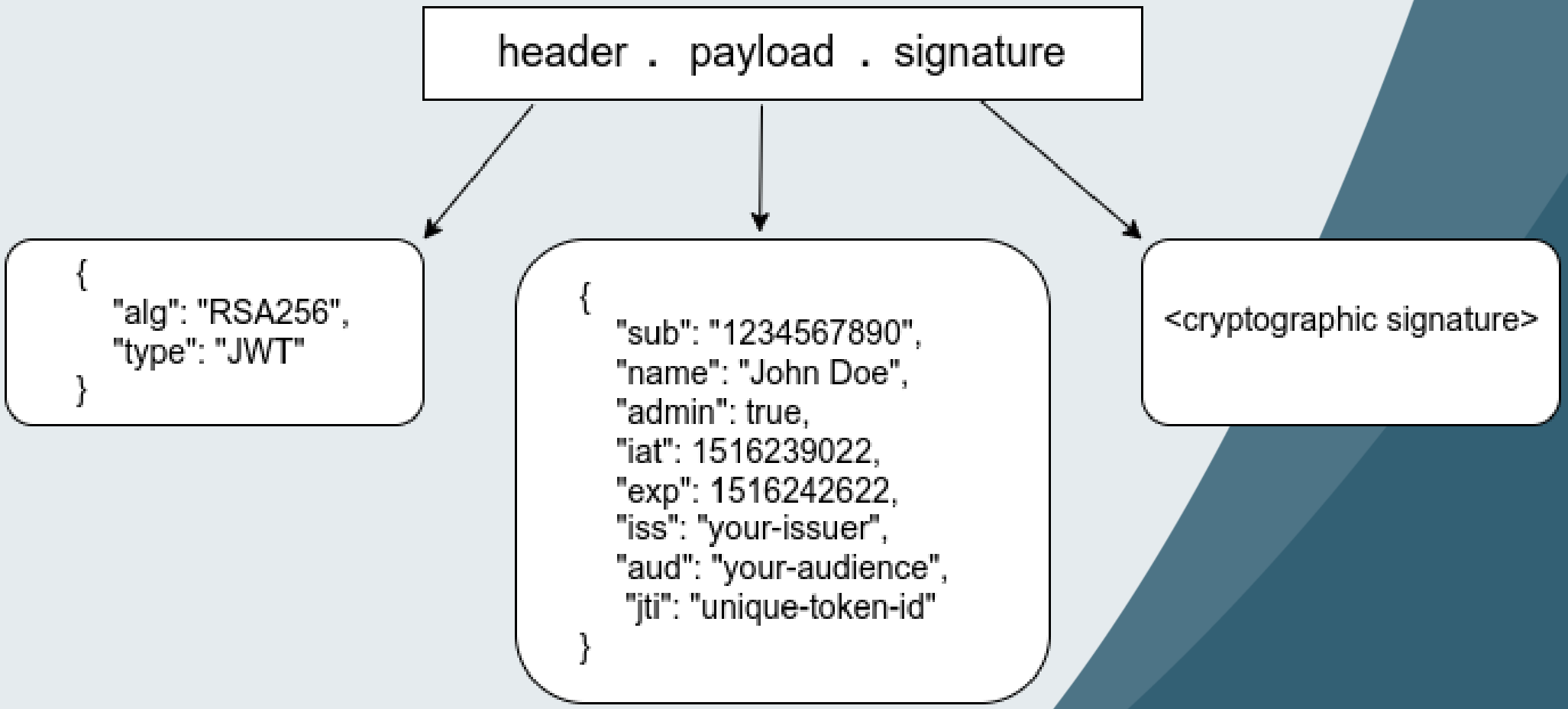
refresh token

токен для получения новой пары токенов (access и refresh)

bearer token

частный случай access токена. В рамках веб приложений эти термины можно использовать, как синонимы.

Структура JWT



Создание

Создать:

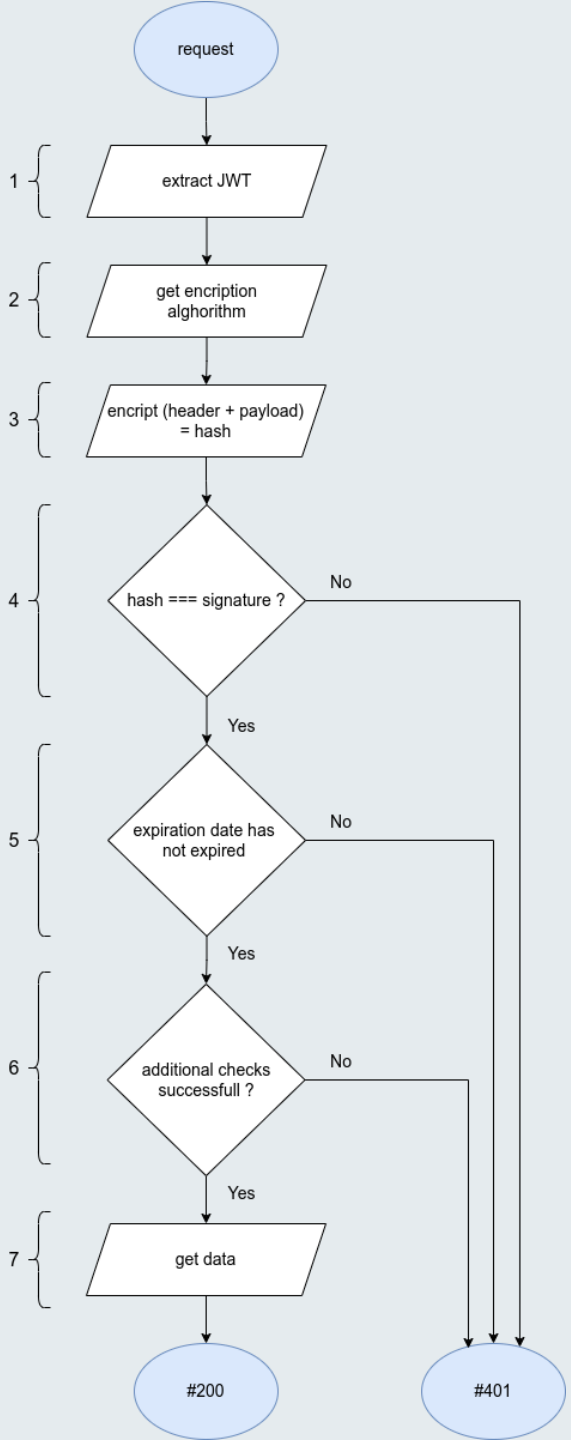
```
var jwt = require("jsonwebtoken");
var token = jwt.sign({name:"Liz"},"super-top-secret-string-of-secrets");
```

Задать заголовок:

```
res.setHeader("Authorization","Bearer "+ token);
```

Декодировать:

```
var t = jwt.decode(token,"super-top-secret-string-of-secrets");
console.log(t);
```

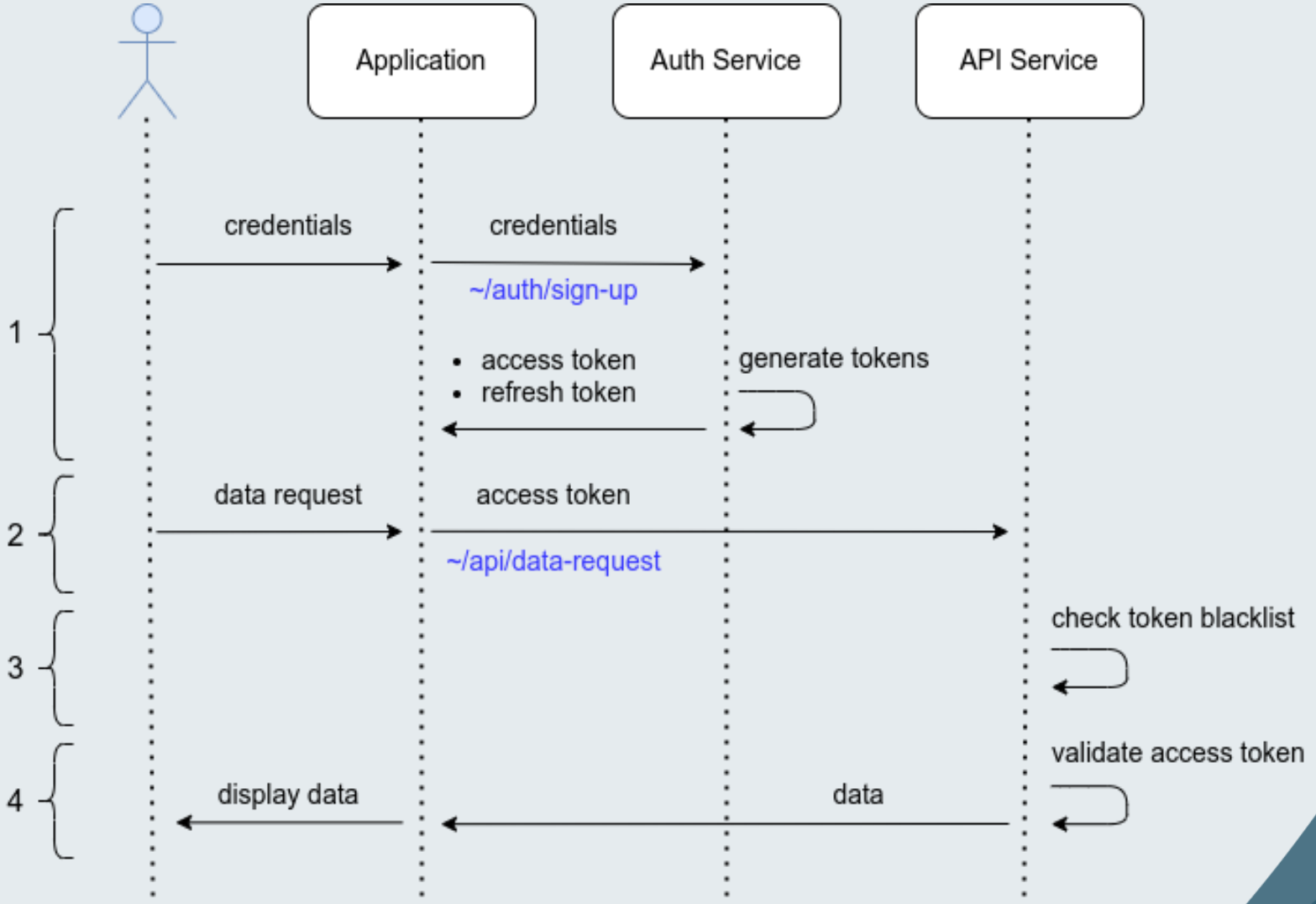


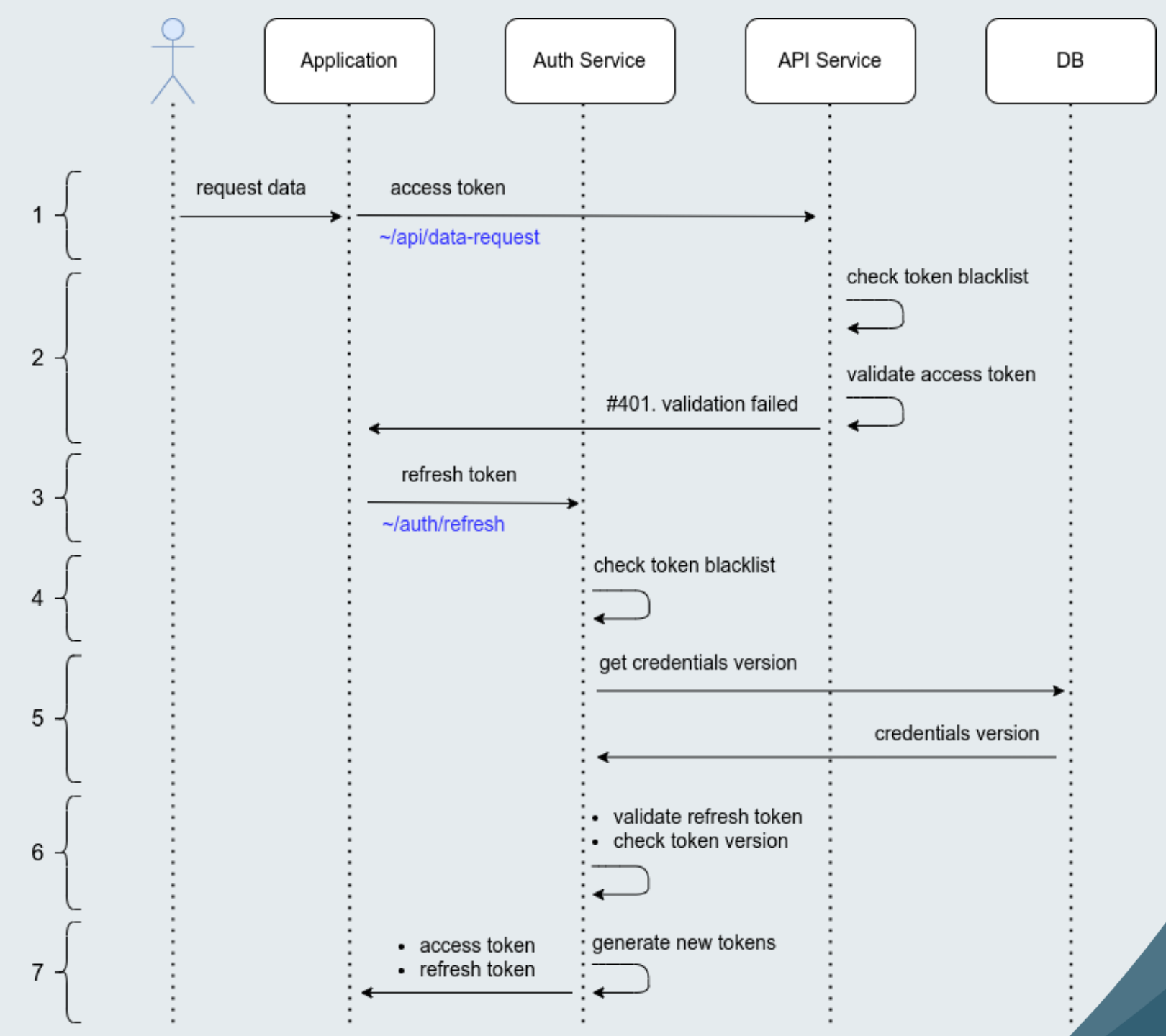
Валидация токенов

Black-list токенов

Когда мы выходим из учетной записи, или сбрасываем пароль, нам нужно отозвать ранее выданные токены, чтобы никто уже не смог зайти с ними в приложение. Для этого токены добавляются в специальный «черный список»

Использование JWT





Обновление JWT

Плюсы

1. Минимальная нагрузка на сервер благодаря отсутствию состояния.
2. Высокая скорость проверки.
3. Удобная передача между сервисами.
4. Нет необходимости хранить пароль после входа.
5. Кроссплатформенность и стандартизированность.

Минусы

1

Если токен украден — злоумышленник получает полный доступ.

2

Токены нельзя отозвать “по умолчанию”.

3

Нужно задавать срок жизни токена.

4

Refresh-токены требуют сложной защиты.

5

Валидация может быть ошибочно настроена.

Заключение

JWT — это удобный и лёгкий механизм аутентификации, который позволяет пользователю пройти вход один раз и дальше работать с приложением без повторного ввода пароля.