Theory 13
API Routes, Middleware, JWT, Auth Flow

WEEK 04

Nội dung chính

- ❑ XÂY DỰNG ĐƯỢC API ROUTES 03
- ❑ CẤU HÌNH ĐƯỢC MIDDLEWARE 04
- ❑ ÁP DỤNG ĐƯỢC JWT ĐỂ XÁC THỰC 05
- ❑ HIỂU ĐƯỢC LUỒNG XÁC THỰC 06

API Routes

- **API Routes** là cách để tạo ra các endpoint backennd trong ứng dụng NextJS
- Thường sử dụng cho các chức năng: **Auth, CRUD, gửi email, upload file,...**

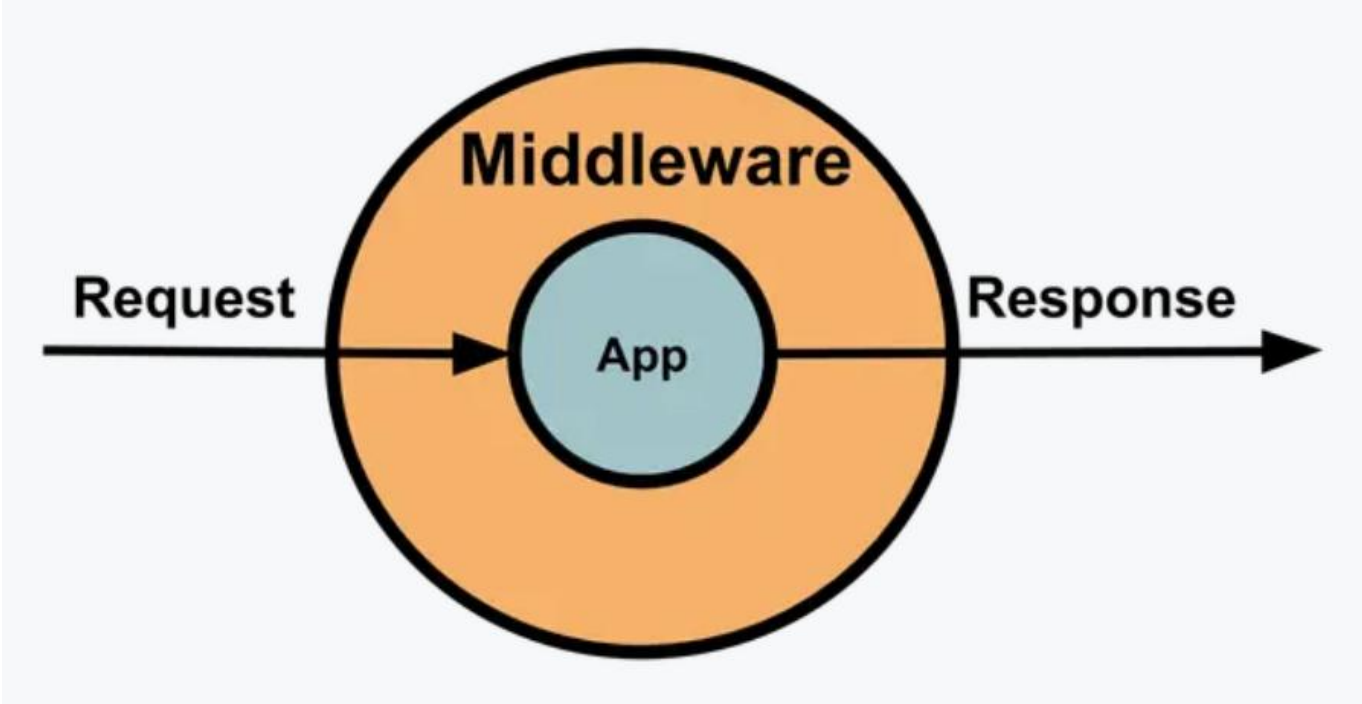
API Routes

- **Cấu trúc**: /app/api/products/route.js # GET /api/products
- **Ví dụ**:

```
// /app/api/hello/route.js
export async function GET() {
    return Response.json({ message: “Hello API Route!” });
}
```

Middleware

- **Middleware** là tầng trung gian, chạy trước khi request truy cập vào route.
- Thường được sử dụng để: **Kiểm tra login, phân quyền, redirect, logging,...**



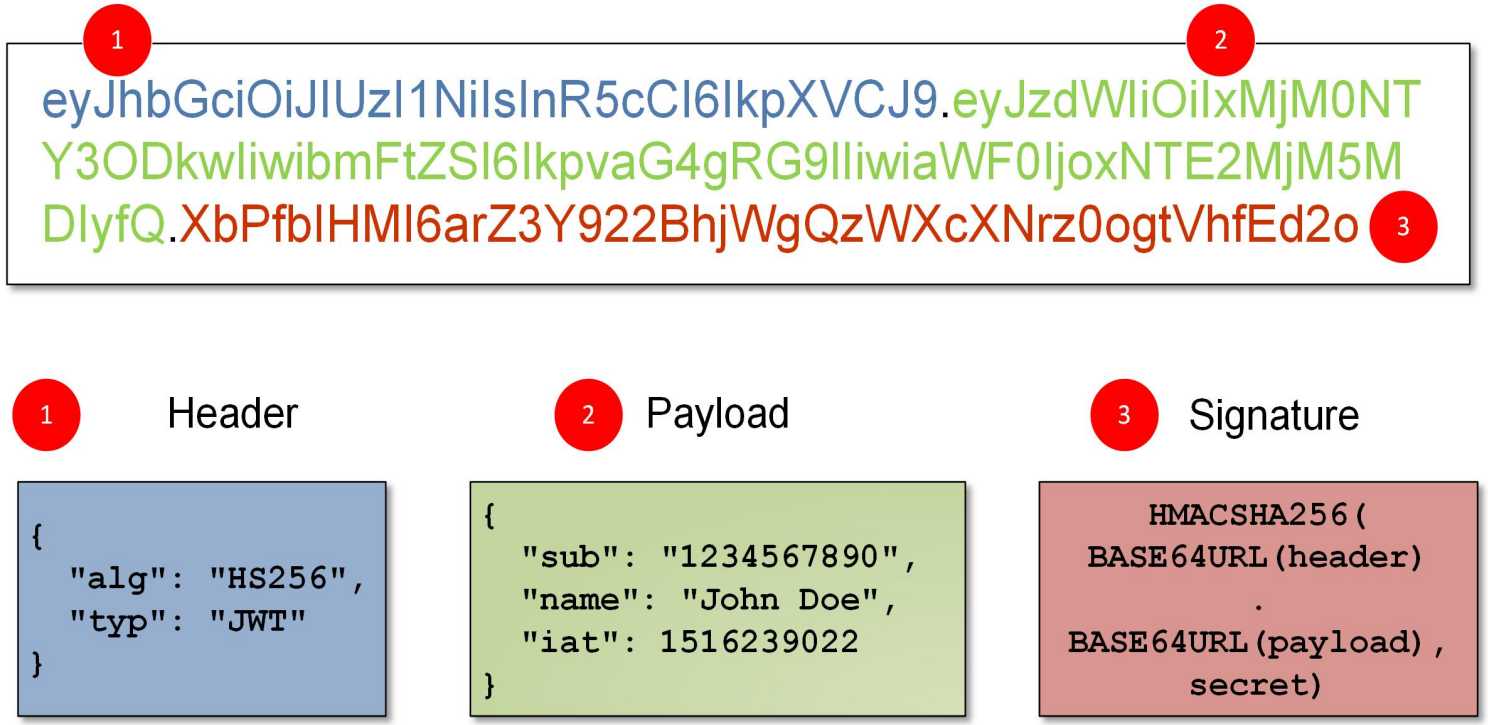
Middleware

❑ Tạo Middleware:

```
import { NextResponse } from 'next/server';
import { verifyToken } from '@/lib/jwt';

export function middleware(req) {
    const token = req.cookies.get('token')?.value;
    const isAuth = token && verifyToken(token);

    if (!isAuth && req.nextUrl.pathname.startsWith('/dashboard')) {
        return NextReponse.redirect(new URL('/login', req.url));
    }

    return NextResponse.next();
}
```

JWT - JSON Web Token

- **JWT** là chuỗi token chứa payload (dữ liệu) đã được mã hóa.
- Gồm 3 phần: **Header, Payload, Signature**



JWT - JSON Web Token

- **Cài đặt JWT:**
  - npm install jsonwebtoken

JWT - JSON Web Token

❑ Tạo file xử lý JWT:

```
// lib/jwt.js
import jwt from 'jsonwebtoken';

const SECRET = process.env.JWT_SECRET;

export function signToken(payload) {
    return jwt.sign(payload, SECRET, { expiresIn: '1h' });
}

export function verifyToken(token) {
   // ...
}
```

JWT - JSON Web Token

❑ Tạo file xử lý JWT:

```
export function signToken(payload) {
    return jwt.sign(payload, SECRET, { expiresIn: '1h' });
}

export function verifyToken(token) {
  try {
      return jwt.verify(token, SECRET);
  } catch {
      return null;
  }
}
```

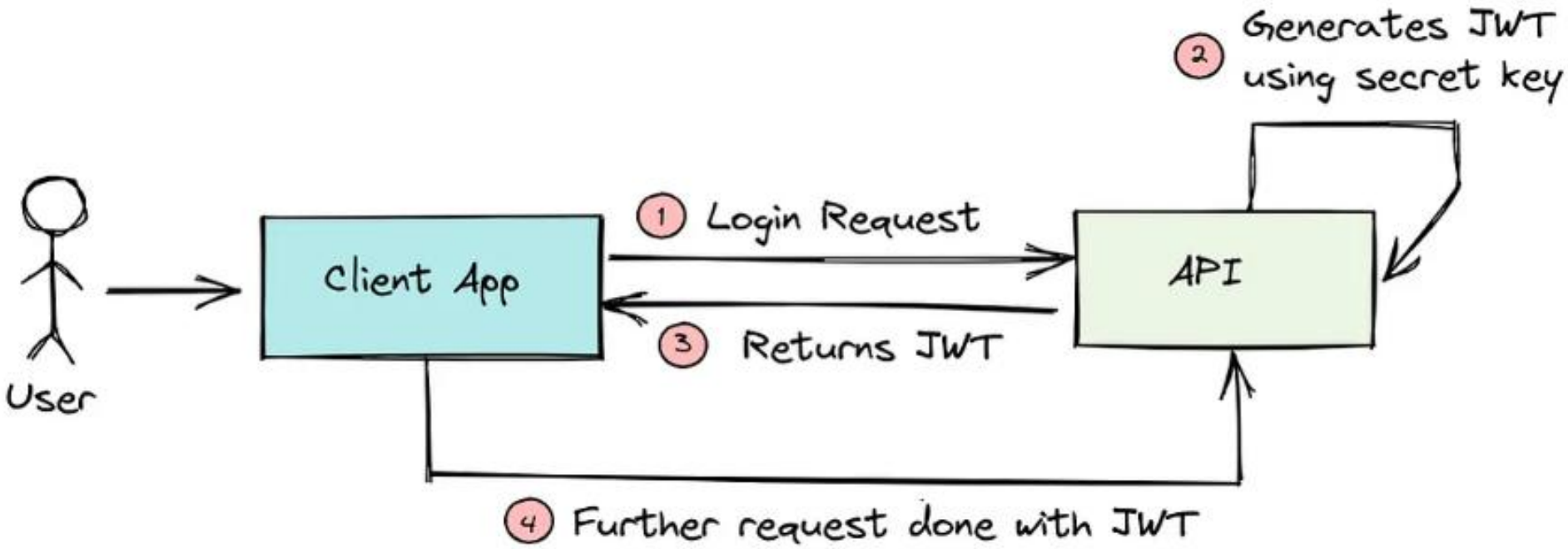
Auth Flow

❑ **Quy trình hoạt động:**

1. Người dùng login -> /api/login
2. API kiểm tra tài khoản, tạo JWT, lưu vào cookies
3. Middleware kiểm tra JWT để bảo vệ /dashboard, /articles,...
4. Người dùng logout -> xóa cookie

Auth Flow

❑ Quy trình hoạt động:



Auth Flow

- **Ví dụ API login:**
  - import thư viện cần thiết để cấu hình API login:

```
// app/api/login/route.js
import { signToken } from '@/lib/jwt';
import { NextResponse } from 'next/server';
```

Auth Flow

❑ Ví dụ API login:

```
export async function POST(req) {
    const { username, password } = await req.json();

    // Simulation
    if (username === 'admin' && password === '123') {
        const token = signToken({ username });
        const res = NextResponse.json({ success: true });
        res.cookies.set('token', token, { httpOnly: true });
        return res;
    }

    return NextResponse.json({ error: 'Invalid credentials' }, { status: 401 });
}
```

Tóm tắt bài học

- ❑ Xây dựng API Routes
- ❑ Cấu hình Middleware
- ❑ Áp dụng JWT để xác thực
- ❑ Luồng xác thực (Auth Flow)