

ปูพื้นฐาน

โดย น้องบอส

NEXT {14}
.JS

เบื้องต้น

2024

Get started by editing `pages/index.tsx`

By ▲

NEXT.JS 13



Introducing the
Vercel Platforms Starter Kit



customer.com



customer.domain.com



janedoe.com



jane.domain.com

Routing

คือการสร้างเส้นทางให้ผู้ใช้งานเว็บไซต์ สามารถเข้าถึงได้อย่างง่ายดายผ่าน URL โดยจะแบ่งออกเป็น 2 ประเภทคือ

1. Static Route คือ เป็น Link ถาวรของหน้านั้น ๆ
2. Dynamic Route คือ Link ที่สามารถแปรผันไปตามปัจจัยต่าง ๆ ได้ เช่นการเข้าชม Detail ของสินค้าต่าง ๆ สำหรับเว็บไซต์ขายสินค้า

การดึงข้อมูล

ใน Next.js เราจะใช้คำสั่ง `fetch` ร่วมกับ `async/await` เพื่อดึงข้อมูลจากแหล่งต่าง ๆ เช่น API Endpoint โดยจะใช้คำสั่งดังต่อไปนี้

```
const res = await  
fetch('http://localhost:3000/api/v1/users')
```

ซึ่งจะได้ข้อมูลมา เราก็ต้องแปลงโดยใช้คำสั่ง `res.json()` อีกที

Component และ Layout

การใช้งาน Component และ Layout เป็นเรื่องที่ค่อนข้างละเอียด สำหรับมือใหม่ ดังนั้นจึงควรศึกษาให้เข้าใจอย่างถ่องแท้ เรามาดูกันดีกว่า

Component หรือก็คือ ส่วนประกอบ สมมติว่าเรามีหน้าเว็บอยู่ 1 หน้า ก็จะประกอบไปด้วย Navbar Header Footer และอะไรอีกหลาย ๆ อย่าง

โดย Layout ก็จะทำหน้าที่รวมสิ่งต่าง ๆ เข้าด้วยกัน และรอรับเนื้อหาที่ส่งมา

ผัง Layout

Footer

Header

Layout 1

Page 1

Page 1 เปรียบเสมือนลูก
ที่อ้างอิงเนื้อหาจากแม่
นั่นก็คือ Layout นั่นเอง

การทำ Styling

CSS Module

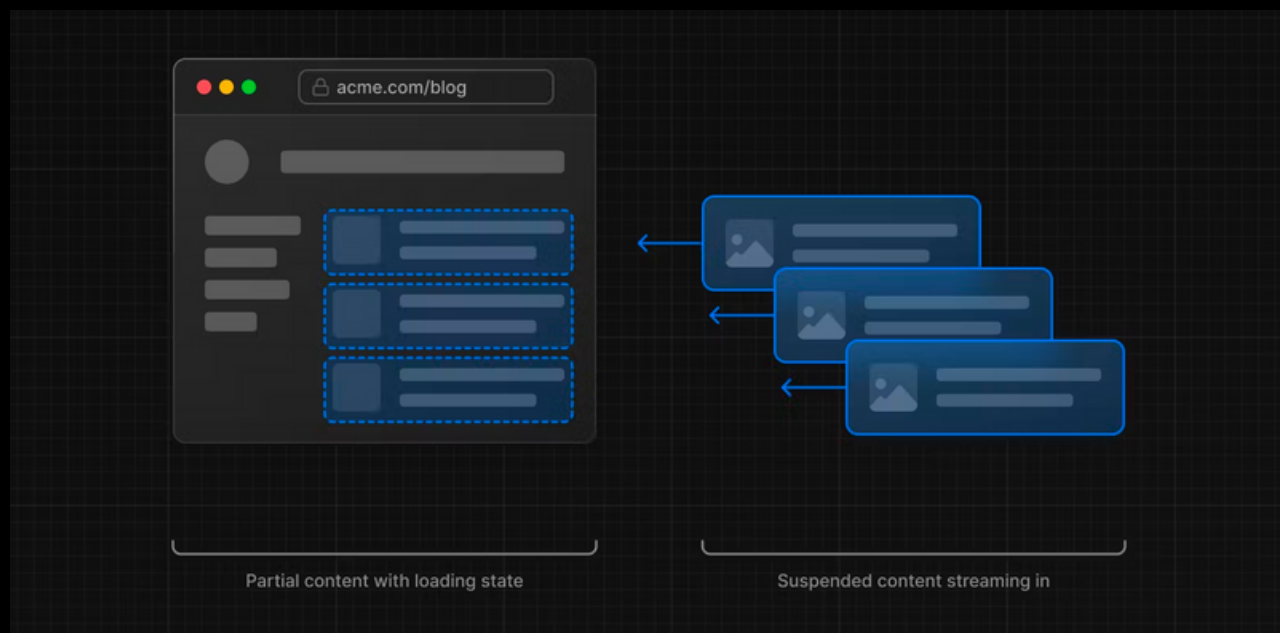
สร้างไฟล์ที่มีนามสกุล
.module.css ขึ้นมา
โดยเราจะปรับแต่ง
หน้าเว็บไซต์ ได้ในนั้น

Tailwind CSS

เป็น Library CSS ที่จะ
ช่วยในการเขียน CSS
โดยที่เราอ้างอิงจาก
Class เพื่อความง่าย

Server Component

สำหรับทำ API หรือ Backend ของเว็บไซต์ โดยที่จะเชื่อมโยงกับ Route ซึ่งจะมีการ Cache เป็นระบบของ Next.js อยู่แล้ว เราจะสามารถ Render หน้า UI จากเซิร์ฟเวอร์ได้โดยตรง ดังเช่นการ Streaming ข้อมูล



จากภาพเราจะเห็นได้ว่าการ Stream ข้อมูลจากฝั่ง Server ไปยังฝั่ง Client ซึ่งโดยส่วนใหญ่จะเป็น Media ต่าง ๆ เช่น รูปภาพ คลิปเสียง หรือวิดีโอ

Client Component

ใช้ในฝั่ง Frontend ซึ่งจะมีคำสั่งต่าง ๆ ใน React เช่น `useState()` เพื่อทำ Stateful Component อีกทั้งเรายังสามารถใช้งาน Component ของ Next.js ได้อีกด้วย โดยต้องมี 'use client'

```
app/counter.tsx
TypeScript

1 'use client'
2
3 import { useState } from 'react'
4
5 export default function Counter() {
6   const [count, setCount] = useState(0)
7
8   return (
9     <div>
10       <p>You clicked {count} times</p>
11       <button onClick={() => setCount(count + 1)}>Click me</button>
12     </div>
13   )
14 }
```

จากภาพจะมีการใช้งาน `useState()` เพื่อเก็บและคงสถานะเอาไว้ โดยไม่จำเป็นต้อง Refresh เบราเซอร์ อย่างเช่นในตัวอย่างที่จะ บวกลบไปเรื่อย ๆ เมื่อผู้ใช้มีการกดปุ่ม