ปั้น AI Agent ด้วย DartanticAI

คู่มือสร้างเอเจนต์จากพื้นฐานสู่การใช้งานจริง

อนุชิต ชโลธร

สำนักพิมพ์ก๊อปวาง

ปั้น AI Agent ด้วย Dartantic AI

คู่มือสร้างเอเจนต์จากพื้นฐานสู่การใช้งานจริง

อนุชิต ชโลธร

คำนำ

หนังสือ **ปั้น AI Agent ด้วย Dartantic AI** เล่มนี้จัดทำขึ้นเพื่อเป็นแนวทางเชิงปฏิบัติสำหรับนักพัฒนาที่ต้องการสร้างและประยุกต์ใช้ระบบปัญญาประดิษฐ์ (AI) ด้วยภาษา Dart ในยุคที่ระบบอัจฉริยะมีบทบาทสำคัญต่อการพัฒนาซอฟต์แวร์ ตั้งแต่การประมวลผลภาษาธรรมชาติ การวิเคราะห์ข้อมูล ไปจนถึงการสร้าง AI Agent ที่สามารถทำงานอัตโนมัติได้อย่างมีประสิทธิภาพ ความเข้าใจและความสามารถในการนำ AI ไปใช้งานจริงจึงกลายเป็นทักษะที่จำเป็นอย่างยิ่ง

แม้ Dart จะเป็นที่รู้จักอย่างแพร่หลายในงานพัฒนาแอปพลิเคชันข้ามแพลตฟอร์ม แต่ด้วยประสิทธิภาพของภาษา ระบบนิเวศที่แข็งแกร่ง และความสามารถในการผสานรวมกับบริการและโมเดล AI สมัยใหม่ Dart จึงมีศักยภาพสูงในการพัฒนาระบบอัจฉริยะ หนังสือเล่มนี้มุ่งแสดงให้เห็นว่า Dart และ DartanticAI สามารถนำมาใช้สร้าง AI Agent ได้อย่างเป็นระบบ ยืดหยุ่น และเหมาะสมกับการใช้งานจริง

เนื้อหาภายในเล่มออกแบบในรูปแบบเชิงปฏิบัติ โดยแต่ละบทนำเสนอแนวคิด ขั้นตอน และตัวอย่างโค้ดที่สามารถนำไปใช้งานและ

ต่อยอดได้ทันที ครอบคลุมตั้งแต่พื้นฐานของ AI Agent ไปจนถึงการเชื่อมต่อกับโมเดลภาษาขนาดใหญ่และการนำระบบไปใช้งานจริง หวังเป็นอย่างยิ่งว่าหนังสือเล่มนี้จะช่วยให้ผู้อ่านสามารถพัฒนาโซลูชัน AI ด้วย Dart ได้อย่างมั่นใจและมีประสิทธิภาพ

สารบัญ

แนะนำ Dartantic AI

การพัฒนาแอปพลิเคชันที่ขับเคลื่อนด้วยปัญญาประดิษฐ์จำเป็นต้องมีเฟรมเวิร์กที่สามารถจัดการความซับซ้อนในการทำงานร่วมกับโมเดลภาษาขนาดใหญ่ (Large Language Models: LLMs) อย่างมีประสิทธิภาพ

Dartantic AI คือเฟรมเวิร์กสำหรับภาษา Dart ออกแบบมาเพื่อวัตถุประสงค์นี้โดยเฉพาะ โดยใช้สถาปัตยกรรมเชิงเอเจนต์ (Agent-based Architecture) เพื่อลดความซับซ้อนในการจัดการผู้ให้บริการ การเรียกใช้เครื่องมือ และการควบคุมบริบทการสนทนา

เนื้อหาในบทนี้จะแนะนำแนวคิดหลัก คุณสมบัติ และโครงสร้างพื้นฐานของ Dartantic AI เพื่อเป็นรากฐานสำหรับการพัฒนาเอเจนต์ขั้นสูงในบทต่อ ๆ ไป

Dartantic AI คืออะไร

Dartantic AI คือ Agent Framework สำหรับภาษา Dart ที่ออกแบบมาเพื่อลดความซับซ้อนในการสร้างแอปพลิเคชัน Generative AI ทั้งฝั่งไคลเอนต์และเซิร์ฟเวอร์ โดยมีเป้าหมายหลัก

เพื่อให้นักพัฒนาสามารถมุ่งเน้นไปที่การออกแบบพฤติกรรมอัจฉริยะ (Agentic Behavior) และตรรกะทางธุรกิจ

Dartantic AI มีอินเทอร์เฟซที่เป็นมาตรฐานช่วยจัดการกับความแตกต่างของ API ของผู้ให้บริการ ลดความซับซ้อนในการจัดการ API ที่แตกต่างกัน การแปลงข้อมูล และการควบคุมลำดับการทำงานของเอเจนต์ ทำให้กระบวนการพัฒนารวดเร็วและเป็นระบบมากขึ้น

คุณสมบัติ

- **พฤติกรรมเชิงเอเจนต์และการเรียกใช้เครื่องมือหลายขั้นตอน (Multi-step Tool Calling)** เอเจนต์สามารถเรียกใช้เครื่องมือหลายชนิดต่อเนื่องกันโดยอัตโนมัติ เพื่อแก้ไขปัญหาที่ซับซ้อนโดยไม่ต้องอาศัยการสั่งงานจากผู้ใช้ในแต่ละขั้นตอน
- **รองรับผู้ให้บริการหลายราย (Multi-provider Support)** มีระบบรองรับผู้ให้บริการ Generative AI ชั้นนำมาพร้อมใช้งาน (Built-in) เช่น OpenAI, Google, Anthropic, Mistral, Cohere และ Ollama
- **มาตรฐาน OpenAI-compatible API** สามารถเชื่อมต่อกับผู้ให้บริการที่รองรับ OpenAI API ซึ่งเป็นมาตรฐานที่แพร่หลายในปัจจุบันได้อย่างสะดวก

- **การส่งผลลัพธ์แบบสตรีมมิง (Streaming Output)** รองรับการสร้างคำตอบแบบเรียลไทม์ เหมาะสำหรับแอปพลิเคชันที่ต้องการการตอบสนองอย่างต่อเนื่อง เช่น แชตบอต
- **ผลลัพธ์แบบมีชนิดข้อมูล (Typed Outputs)** ผสานระบบชนิดข้อมูลของภาษา Dart เข้ากับการแปลงข้อมูล JSON เพื่อให้ได้ผลลัพธ์ที่มีโครงสร้างชัดเจนและลดความผิดพลาด
- **อินพุตแบบมัลติมีเดีย (Multimedia Inputs)** สามารถประมวลผลอินพุตที่ประกอบด้วยข้อความ รูปภาพ และไฟล์ข้อมูลภายในคำสั่งเดียวกัน
- **การสร้างสื่อ (Media Generation)** รองรับการสร้างและสตรีมรูปภาพ, PDF และไฟล์ประเภทอื่น ๆ จากผู้ให้บริการอย่าง OpenAI, Google และ Anthropic
- **Embeddings** รองรับการสร้างเวกเตอร์สำหรับงานค้นหาเชิงความหมาย (Semantic Search) และการวิเคราะห์ความคล้ายคลึง
- **การแสดงผลกระบวนการคิดของโมเดล (Model Reasoning)** รองรับการแสดงผลกระบวนการให้เหตุผลของโมเดล (ที่มักเรียกว่า “Thinking”) จากผู้ให้บริการที่รองรับคุณสมบัตินี้

- **เครื่องมือฝั่งเซิร์ฟเวอร์ (Server-side Tools)** สามารถเรียกใช้เครื่องมือที่โฮสต์โดยผู้ให้บริการ เช่น การค้นหาเว็บ, การรันโค้ด, และการสร้างภาพ
- **รองรับ Model Context Protocol (MCP)** สามารถทำงานร่วมกับ MCP Server เพื่อขยายขีดความสามารถของเอเจนต์ได้อย่างยืดหยุ่น
- **พร้อมสำหรับใช้งานจริง (Production Ready)** มีระบบบันทึกข้อมูลเหตุการณ์ (Logging), การจัดการข้อผิดพลาด, และกลไกการลองใหม่ (Retry) ที่เหมาะสำหรับสภาพแวดล้อมโปรดักชัน
- **ขยายความสามารถได้ (Extensible)** โครงสร้างออกแบบมาให้สามารถเพิ่มผู้ให้บริการ, สร้างเครื่องมือเฉพาะทาง, หรือเชื่อมต่อกับ MCP Server ได้ง่าย

โครงสร้างพื้นฐานเอเจนต์

โครงสร้างพื้นฐานเอเจนต์ (Agent) เป็นจุดเริ่มต้นสำคัญในการพัฒนาแอปพลิเคชัน AI ด้วย Dartantic AI โดยแนวคิดหลักคือการห่อหุ้มความสามารถของโมเดลภาษา (LLM) ไว้ในอ็อบเจกต์เดียว เพื่อให้ผู้พัฒนาสามารถสื่อสารกับโมเดลได้อย่างเป็นระบบ เรียบง่าย และเปลี่ยนผู้ให้บริการได้โดยไม่กระทบกับโค้ดส่วนอื่นของระบบ

```dart
import 'package:dartantic_ai/dartantic_ai.dart';

final agent = Agent('claude'); // or 'openai',
'gemini', etc.
final result = await agent.send('Hello!');
print(result.output);
```

จากโค้ดข้างต้น นำเข้าแพ็กเกจ dartantic_ai ซึ่งเป็นไลบรารีหลักที่ใช้สำหรับสร้างและจัดการเอเจนต์ AI ในภาษา Dart

```dart
import 'package:dartantic_ai/dartantic_ai.dart';
```

สร้างอินสแตนซ์ของ Agent โดยกำหนดชื่อผู้ให้บริการโมเดลภาษา (Provider) เป็น 'claude' ซึ่งสามารถเปลี่ยนเป็น 'openai', 'gemini' หรือผู้ให้บริการอื่นที่รองรับได้ โดยไม่ต้องแก้ไขโครงสร้างโค้ดหลัก

```dart
final agent = Agent('claude'); // หรือ 'openai',
'gemini', etc.
```

ส่งข้อความไปยังเอเจนต์เพื่อให้โมเดลประมวลผล คำสั่งนี้เป็นแบบ asynchronous และจะคืนค่าเป็นอ็อบเจกต์ผลลัพธ์ (ChatResult) ที่บรรจุข้อมูลการตอบกลับจากโมเดล