

資料庫 - 概述

原文：[ENG-08-0-Database-General.md](#)

概述

Drogon 內建資料庫讀寫引擎，連線操作採用非阻塞 I/O 技術，應用程式自底層至上層皆以高效非阻塞非同步模式運作，確保高併發效能。目前支援 PostgreSQL 與 MySQL 資料庫。若需使用資料庫，必須先安裝對應資料庫的開發環境，Drogon 會自動偵測相關標頭與函式庫並編譯對應部分。資料庫開發環境準備請參考[開發環境](#)。

Drogon 亦支援 sqlite3 資料庫，適合輕量級應用。非同步介面以執行緒池實作，與上述資料庫介面一致。

DbClient

Drogon 資料庫操作的基本類別為 **DbClient**（抽象類別，具體型別依建構介面而定）。與一般資料庫介面不同，**DbClient** 物件不代表單一連線，而是可包含多個連線，可視為**連線池物件**。

DbClient 提供同步與非同步介面，非同步介面同時支援阻塞與非阻塞模式。建議配合 Drogon 非同步框架時使用非阻塞非同步介面。

通常呼叫非同步介面時，**DbClient** 會隨機選擇管理的閒置連線執行查詢，結果回傳後由 **DbClient** 處理資料並透過 callback 回傳給呼叫者；若無閒置連線，執行內容會暫存，待有連線完成自身 SQL 請求後，會從暫存取出待執行命令。

DbClient 詳細用法請參考[DbClient](#)。

Transaction

可由 **DbClient** 產生 transaction 物件以支援交易操作。除多出 **rollback()** 介面外，transaction 物件基本用法與 **DbClient** 相同。transaction 類別為 **Transaction**，詳情請參考[Transaction](#)。

ORM

Drogon 亦支援 **ORM**。使用者可用 drogon_ctl 指令讀取資料庫表格並產生對應 model 原始碼，再透過 **Mapper<MODEL>** 類別模板執行 model 的資料庫操作。Mapper 提供標準資料庫操作的簡易介面，讓使用者無需撰寫 SQL 即能對表格進行增刪改查。ORM 詳細用法請參考[ORM](#)。

下一步: [DbClient](#)