

MongoDB Java Demo Web Application

建置&操作教學

Teddy Lai

目錄

- [前言介紹](#)
- [IDEA 安裝與JAVA環境建置](#)
- [Maven版控，啟動服務，設定檔修改](#)
- [PostMan & Demo AP 操作](#)
- [MongoDB Community Server & Atlas 使用](#)
- [使用Docker 快速搭建 MongoDB Single & Cluster](#)
- [Docker-Compose.yml With Application](#)
- [Ubuntu 22.0.4 建置 MongoDB Replica Set](#)
- [參考網站](#)

前言介紹

前言：

此 PowerPoint 為一簡易的指南

幫助非開發人員建置簡單的本地 JAVA Demo AP

附上簡易設定檔介紹 與 操作教學

使用者可依需求自行修改 MongoDB 數據來源

最後以本專案設定檔為範例

建置專案設定檔內的 MongoDB 單機與 Replica Set 環境

專案使用技術：

BackEnd :

Java 17 LTS , SpringBoot 3.1.0

DataBase :

MongoDB 6.0.6

MongoDB JPA :

MongoJDBC , MongoRepository , MongoTemplate

FrontEnd :

Thymeleaf , JQuery

IDEA 安裝與JAVA環境建置

1. 下載對應版本 JAVA JDK17

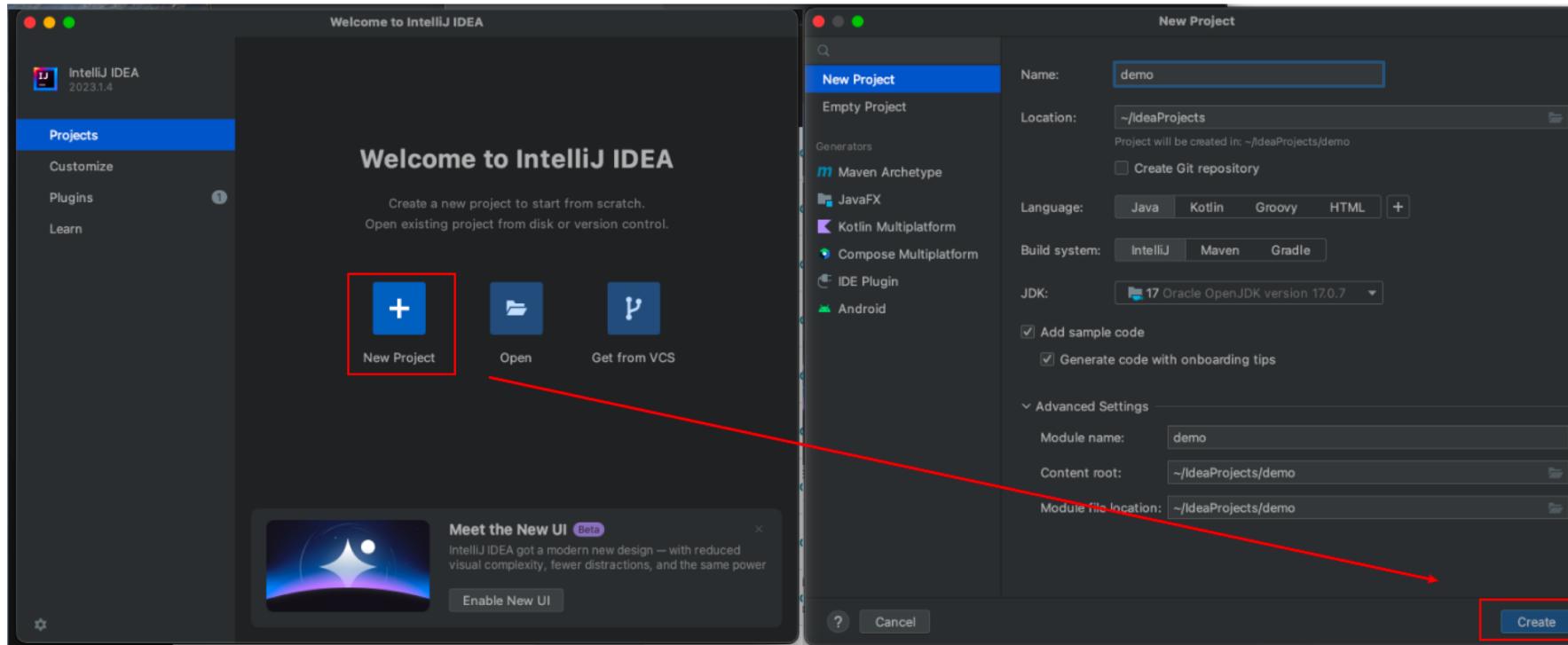
*若是Windows 需要另外額外設定 [JAVA_HOME](#) 環境變數 OS 才能找到JVM路徑執行java指令

測試是否設定成功 : java --version

2. 下載 IntelliJ Community Edition

*若為Windows import project 出現部分文字亂碼，可以[修改下文字編碼](#) (主流編碼為 UTF-8)

3. Create A New Demo Project



4. Hello World With First Project

a. 修改預設Demo代碼並儲存

b. 右鍵Run ‘Main.main()’

Main.java

```
1 // Press Shift twice to open the Search Everywhere dialog and type 'show whitespaces',
2 // then press Enter. You can now see whitespace characters in your code.
3 public class Main {
4     public static void main(String[] args) {
5         // Press Opt+Enter with your caret at the highlighted text to see how
6         // IntelliJ IDEA suggests fixing it.
7         System.out.printf("Hello and welcome!");
8
9         // Press Ctrl+R or click the green arrow button in the gutter to run the code.
10        for (int i = 1; i <= 5; i++) {
11
12            // Press Ctrl+D to start debugging your code. We have set one breakpoint
13            // for you, but you can always add more by pressing Cmd+F8.
14            System.out.println("hello world i = " + i);
15        }
16    }
17}
```

Run: Main

```
/Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=49662:/Applications/IntelliJ IDEA CE.app/Contents/bin -ea
```

Hello and welcome!
hello world i = 1
hello world i = 2
hello world i = 3
hello world i = 4
hello world i = 5
Process finished with exit code 0

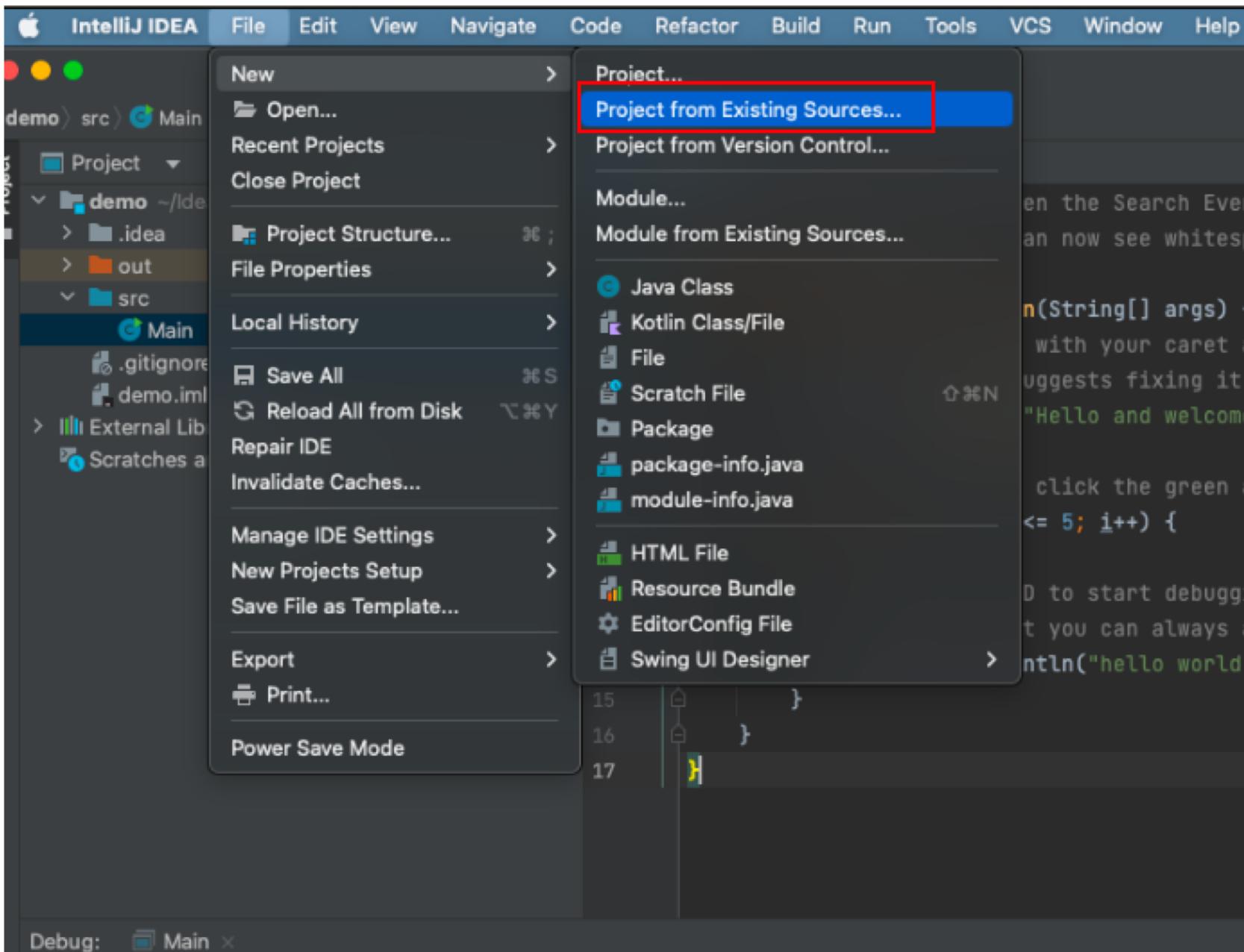
Show Context Actions

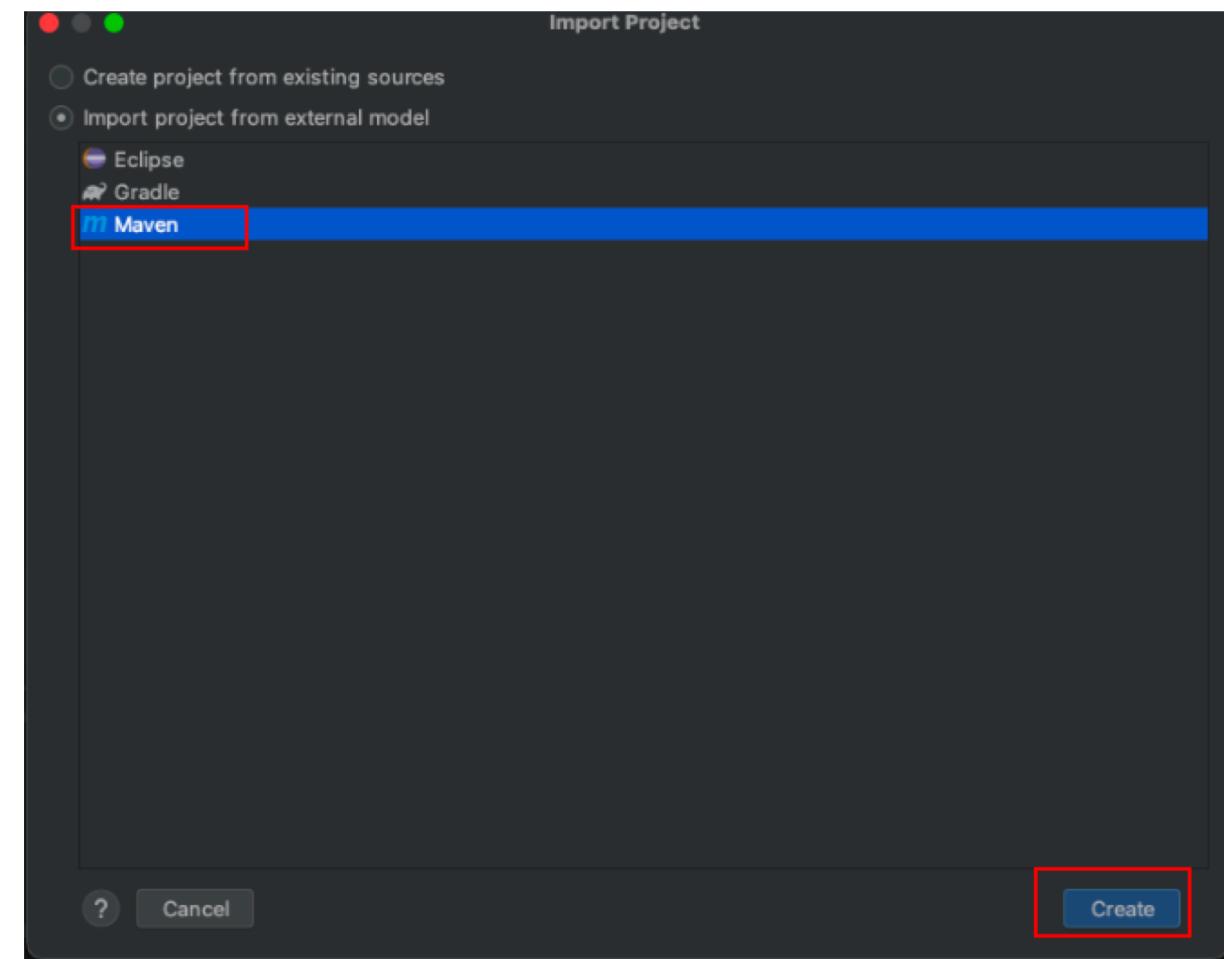
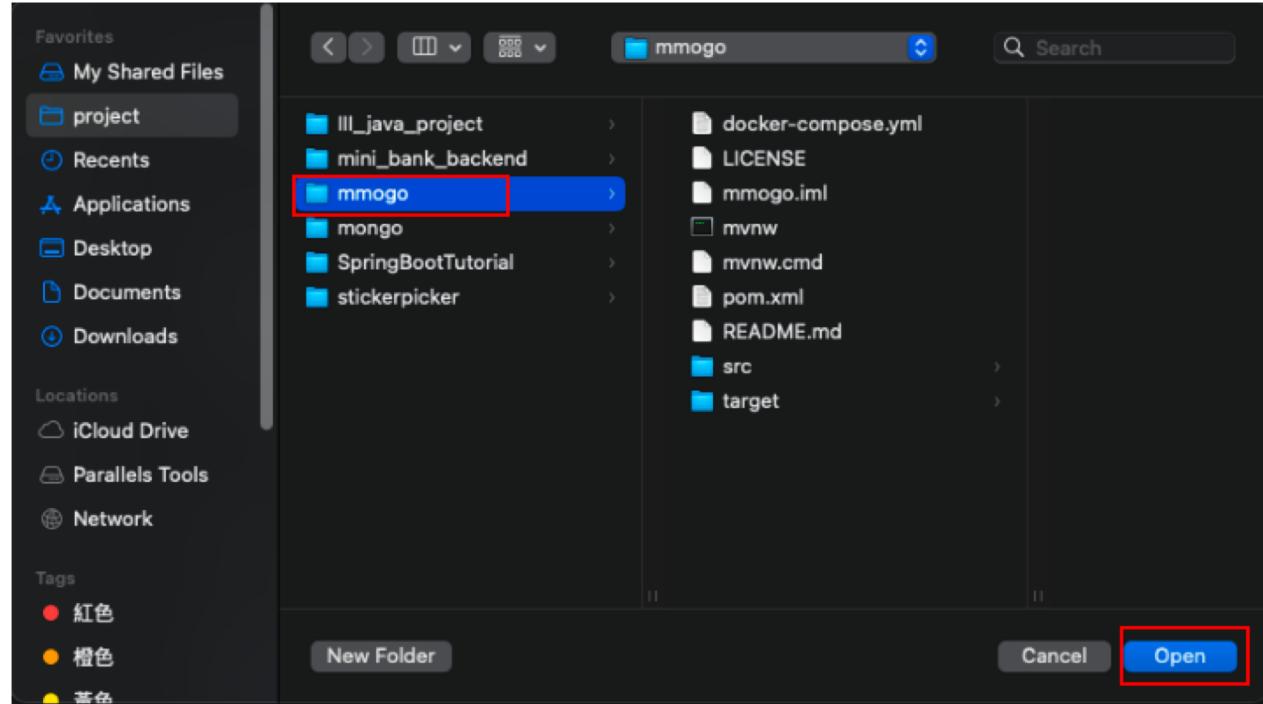
- Paste ⌘V
- Copy / Paste Special >
- Column Selection Mode ⇧⌘B
- Go To >
- Folding >
- Analyze >
- Refactor >
- Generate... ⌘N
- Run 'Main.main()' ⌘⌥R
- Debug 'Main.main()'" ⌘⌥D
- Run 'Main.main()' with Coverage >
- Modify Run Configuration...
- Open In >
- Local History >
- Compare with Clipboard >
- Create Gist...

Version Control Run TODO Problems Terminal Services Build

Build completed successfully in 6 sec, 461 ... (moments ago)

5. Import Existing Sources With Maven Project





5. IntelliJ 基本組件介紹

The screenshot displays the IntelliJ IDEA IDE interface with the following components:

- Project View:** On the left, the project structure is shown under the "Project" tab. A red box highlights the "Project" tab and the "mmogo" project folder. Inside the project folder, "pom.xml" is selected and highlighted with a red box.
- Code Editor:** The main area shows the content of the "pom.xml" file for the "mmogo" project. A red arrow points from the "target" node in the Project view to the corresponding node in the code editor. The code editor also has a red box highlighting the "Maven" tab in the top right corner.
- Services:** Below the code editor, the "Services" panel is visible, showing a message: "No services configured." with a "Add service (⌘N)" button.
- Bottom Navigation:** The bottom bar includes tabs for Git, TODO, Problems, Terminal, Build, Services, and Dependencies. It also shows a status message: "Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Maven library shared indexes // Always download // Download once // Don't show again // Configur..." and a commit message: "1:1 LF UTF-8 Tab* master".

mmogo - pom.xml (mmogo)

The screenshot shows a Maven project named "mmogo" in an IDE. The project structure on the left includes .idea, .mvn, src, target, .gitignore, docker-compose.yml, LICENSE, mmogo.iml, mvnw, mvnw.cmd, and pom.xml. The pom.xml file is open in the center editor, displaying its XML code. On the right, the Maven tool window shows the project tree with "mmogo" selected. A red box highlights the pom.xml tab in the editor and the "mmogo" entry in the Maven tool window. Another red box highlights the "Services" tab at the bottom. The status bar at the bottom shows a message about shared indexes and the current commit status.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.smart</groupId>
  <artifactId>mmogo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>mmogo</name>
  <description>mmogo</description>
  <url>http://maven.apache.org</url>
<properties>
  <maven.compiler.source>17</maven.compiler.source>
  <maven.compiler.target>17</maven.compiler.target>
  <java.version>17</java.version>
  <spring-boot.version>3.1.0</spring-boot.version>
</properties>
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>3.1.0</version>
</parent>
<build>
  <finalName>mmogo</finalName>

```

Services

No services configured.
Add service (⌘N)

Select service to view details

Bookmarks

- mmogo
- Breakpoints

Log & Service Status

Debug 中斷點

Build Services Dependencies

Git TODO Problems Terminal

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Maven library shared indexes // Always download // Download once // Don't show again // Configur... (8 minutes ago)

11:1 LF UTF-8 Tab* master

mmogo - pom.xml (mmogo)

Project Commit Maven

mmogo sources root, ~/project/mmogo

mmogo pom.xml (mmogo) x

```
<properties>
    <maven.compiler.source>17</maven.compiler.source>
    <maven.compiler.target>17</maven.compiler.target>
    <java.version>17</java.version>
    <spring-boot.version>3.1.0</spring-boot.version>
</properties>
<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.1.0</version>
</parent>
<build>
    <finalName>mmogo</finalName>
    <plugins>
        <!-- Spring Boot Maven Plugin -->
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <version>${spring-boot.version}</version>
        </plugin>
    </plugins>
</build>
```

target .gitignore docker-compose.yml LICENSE mmogo.iml mvnw mvnw.cmd pom.xml README.md

Debug: MmogoApplication x

Frames | Variables | Console

Evaluate expression (e) or add a watch (w)

Variables are not available

Switch frames from anywhere in the IDE w...

Dependencies Structure

debug 與版本控制

Git Debug Services Build Terminal

Build completed successfully in 260 ... (today 11:06)

mmogo - pom.xml (mmogo)

File 5 Project Errors

pom.xml ~/project/mmogo 5 problems

Provides transitive vulnerable dependency maven:org.yaml:snakeyaml:1.33 CVE-2022-41854 6.5 Out-of-bounds Write vulnerability with med

Type: In word 'mmogo' :8

Type: In word 'mmogo' :9

Type: In word 'mmogo' :24

Type: In word 'mongo' :60

TODO Bookmarks Problems

Notifications

Maven版控，啟動服務，設定檔修改

1. Maven 版控

Maven 介紹：

開發人員使用約定的 **pom.xml** 拉取 Maven 管理專案所依賴的庫
用來簡化 JAVA 項目構建與套件管理

Maven常用指令：

mvn clean: 清理項目，刪除生成的目錄和檔案。

mvn compile: 編譯項目的源代碼。

mvn package: 將項目打包，生成 JAR 或 WAR 檔案，放在 target 目錄中。

mvn install: 將打包的項目安裝到本地的 Maven 儲存庫中，供其他項目依賴使用。

mvn clean install: 進行 clean、compile 和 install 操作的合併指令。

mvn help:help: 顯示 Maven 幫助文檔，列出常用的 Maven 命令和插件。

2. Maven Install 專案

The screenshot shows an IDE interface with the following details:

- Project:** mmogo
- Maven:** Maven tool window is open.
- Code Editor:** The pom.xml file is open, showing Spring Boot dependencies. A context menu is open over the first dependency element, with the option "mvn install" highlighted.
- Services:** No services configured.
- Bookmarks:** mmogo
- Git:** Git is not installed message.
- Bottom Bar:** Git, TODO, Problems, Terminal, Services, Dependencies, Bookmarks.

```
</build>
<dependencies>
    <!-- Spring Boot Starter -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-boot</artifactId>
        <version>${spring-boot.version}</version>
    </dependency>
    <!-- Spring MVC Starter -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
        <version>${spring-boot.version}</version>
    </dependency>
    <!-- JPA -->
    <!--
        <dependency>-->
        <!--
            <groupId>org.springframework.boot</groupId>-->
            <!--
                <artifactId>spring-boot-starter-data-jpa</artifactId>-->
                <version>${spring-boot.version}</version>-->
            </dependency>-->
        <!-- thymeleaf -->
        <dependency>
```

下方 running log 看到 success 即為下載依賴庫成功

The screenshot shows the IntelliJ IDEA interface with the 'mmogo' project open. The 'pom.xml' file is selected in the editor. The build log in the bottom panel shows the Maven build process:

```
[INFO] Assembling webapp [mmogo] in [/Volumes/My Shared Files/Home/project/mmogo/target/mmogo]
[INFO] Processing war project
[INFO] Copying webapp resources [/Volumes/My Shared Files/Home/project/mmogo/src/main/webapp]
[INFO] Building war: /Volumes/My Shared Files/Home/project/mmogo/target/mmogo.war
[INFO]
[INFO] --- spring-boot:3.1.0:repackage (repackage) @ mmogo ---
[INFO] Replacing main artifact /Volumes/My Shared Files/Home/project/mmogo/target/mmogo.war with repackaged archive, adding nested dependency
[INFO] The original artifact has been renamed to /Volumes/My Shared Files/Home/project/mmogo/target/mmogo.war.original
[INFO]
[INFO] --- install:3.1.1:install (default-install) @ mmogo ---
[INFO] Installing /Volumes/My Shared Files/Home/project/mmogo/pom.xml to /Users/teddylai/.m2/repository/com/smartz/mmogo/0.0.1-SNAPSHOT/mmogo-0.0.1-SNAPSHOT.pom
[INFO] Installing /Volumes/My Shared Files/Home/project/mmogo/target/mmogo.war to /Users/teddylai/.m2/repository/com/smartz/mmogo/0.0.1-SNAPSHOT/mmogo-0.0.1-SNAPSHOT.war
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  7.504 s
[INFO] Finished at: 2023-07-17T23:00:00-07:00
[INFO] -----
[WARNING]
[WARNING] Plugin validation issues were detected in 1 plugin(s)
[WARNING]
[WARNING] * org.apache.maven.plugins:maven-war-plugin:3.3.2
[WARNING]
[WARNING] For more or less details, use 'maven.plugin.validation' property with one of the values (case insensitive): [BRIEF, DEFAULT, V
[WARNING]

Process finished with exit code 0
```

The 'BUILD SUCCESS' message is highlighted with a red box. The bottom status bar indicates 'Git not installed: xcode-select: note: No developer tools were found, requesting install.' and the current branch is 'master'.

3. 啟動服務

搜尋SpringBoot 的啟動入口 Class 關鍵字 **@SpringBootApplication**

搜尋快捷鍵：

command + shift + f (mac)

ctrl + shift + f (windows)

The screenshot shows an IDE interface with several windows open. On the left, the Project Explorer shows a Java project structure with a file named `MmogoApplication.java` selected. The code editor window displays the following Java code:

```
mmogo - MmogoApplication.java
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;
import java.util.List;
@TeddyLai
@SpringBootApplication
public class MmogoApplication extends ServletInitializer {
    public static void main(String[] args) {
        SpringApplication.run(MmogoApplication.class, args);
    }
}
```

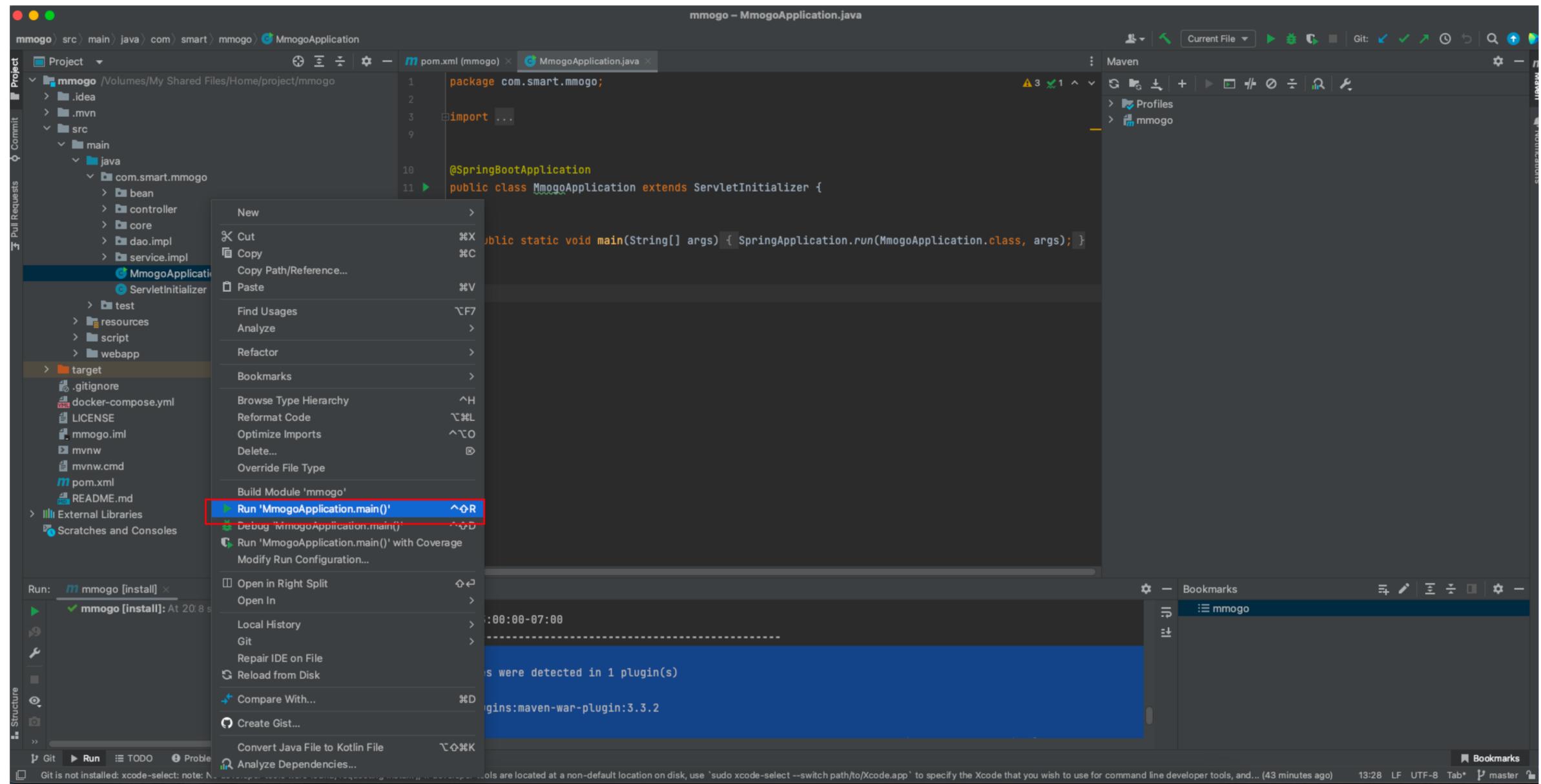
A search dialog titled "Find in Files 1 match in 1 file" is open, with the search term `@SpringBootApplication` highlighted. The results pane shows the match found in the `MmogoApplication.java` file.

On the right side of the interface, there are other windows showing configuration files like `log4j2.properties` and `application.yml`, and a terminal window showing the application's startup logs. The terminal output includes:

```
2023-07-18T11:51+08:00 INFO 39378 --- [           main] com.smart.mmogo.MmogoApplication      : Started MmogoApplication in 1.272 seconds (process running for : 1.272)
2023-07-18T11:51+08:00 INFO 39378 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path: /
Disconnected from the target VM, address: '127.0.0.1:52938', transport: 'socket'
Process finished with exit code 130 (interrupted by signal 2: SIGINT)
```

The bottom status bar indicates the build was successful: "Build completed successfully in 260 ms (today 11:06)".

初次使用 AP , 對著啟動 Class 右鍵 Run Application



查看下方 Log 區塊，啟動完成！

* 初始啟動完成後，右上方啟動區塊可自由選擇要啟動的服務，啟動模式，與關閉服務

The screenshot shows a Java Spring Boot project named "mmogo" in an IDE. The project structure includes a `src` folder with `main` and `java` subfolders containing `MmogoApplication.java` and various service classes. The `resources` folder contains `application.yml` and `log4j2.properties`. The `Run` dropdown menu is set to "MmogoApplication".

The `application.yml` file is open, showing configuration for MongoDB connections. A red box highlights the "啟動區塊" (Startup Block) at the top right of the code editor.

The bottom half of the screen shows the "Log" or "Output" tab, which displays the Spring Boot startup logs. A red box highlights the "啟動信息" (Startup Information) section. The logs indicate the application is starting with Java 17.0.7, using Spring Boot v3.1.0, and bootstrapping Spring Data MongoDB repositories. It also shows Tomcat initialization and the start of the Servlet engine.

```
mmogo - application.yml
-----
username: root
password: root
#####
ver 2.4以後用 uri string 連接 (Template && Repository && JDBC)
#####
local Standalone db 有密碼與無密碼連接方式
#####
DB啟動參考指令
#####
有密碼 :
mongod --auth --dbpath "/Users/teddylai/database/mongo/mongodb/"
沒密碼 :
mongod --dbpath "/Users/teddylai/database/mongo/mongodb/data"
#
uri: mongodb://localhost:27017/demo
#
uri: mongodb://root:root@127.0.0.1:27017/demo
#####
cloud atlas 生成的uri
uri: mongodb+srv://root:root@cloud-cluster0.prhwadb.mongodb.net/demo?retry
#####
docker Standalone & cluster
#
uri: mongodb://localhost:27010/demo
#
uri: mongodb://mongo1:27011,mongo2:27012,mongo3:27013/demo?replicaSet=myD
#####
docker-compose Standalone & cluster
Document 1/1

Run: MmogoApplication
/Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java ...
----- / ---'---_(_)_--_--_ \\\ \
(( )\_--|_-|_|'_`_/\_`_|\_\\ \
\_\_||_|_||_|_||_|_) ) )
' |___| .__|_|_|_||_\_, | / / /
=====|_|=====|_|=/|/_/_
:: Spring Boot :: (v3.1.0)

2023-07-17T23:46:01.915-07:00 INFO 1117 --- [           main] com.smart.mmogo.MmogoApplication        : Starting MmogoApplication using Java 17.0.7 with PID 1117 (/Vol
2023-07-17T23:46:01.917-07:00 DEBUG 1117 --- [           main] com.smart.mmogo.MmogoApplication        : Running with Spring Boot v3.1.0, Spring v6.0.9
2023-07-17T23:46:01.918-07:00 INFO 1117 --- [           main] com.smart.mmogo.MmogoApplication        : No active profile set, falling back to 1 default profile: "defau
2023-07-17T23:46:02.441-07:00 INFO 1117 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data MongoDB repositories in DEFAULT mode.
2023-07-17T23:46:02.504-07:00 INFO 1117 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 57 ms. Found 1 Mongo
2023-07-17T23:46:02.837-07:00 INFO 1117 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8085 (http)
2023-07-17T23:46:02.846-07:00 INFO 1117 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2023-07-17T23:46:02.846-07:00 INFO 1117 --- [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.8]
```

4. 重要設定檔

a. `~/mmogo/pom.xml`

maven 管理套件

b. `~/mmogo/docker-compose.yml`

docker 容器設定檔

c. `~/mmogo/src/resources/application.yml`

AP 設定檔 (包含連線設定)

5. pom.xml 內容

註解快捷鍵：

command + / (mac)

ctrl + / (windows)

V pom.xml (mmogo) X

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/pom-4.0.0.xsd">
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>com.smart</groupId>
4   <artifactId>mmogo</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6   <packaging>war</packaging>
7   <name>mmogo</name>
8   <description>mmogo</description>
9   <url>http://maven.apache.org</url>
10
11
12   <properties>
13     <maven.compiler.source>17</maven.compiler.source>
14     <maven.compiler.target>17</maven.compiler.target>
15     <java.version>17</java.version>
16     <spring-boot.version>3.1.0</spring-boot.version>
17   </properties>          jar 版本資訊
18   <parent>
19     <groupId>org.springframework.boot</groupId>
20     <artifactId>spring-boot-starter-parent</artifactId>
21     <version>3.1.0</version>
22   </parent>
23   <build>
24     <finalName>mmogo</finalName>
25     <plugins>
26       <!-- Spring Boot Maven Plugin -->
27       <plugin>
28         <groupId>org.springframework.boot</groupId>
29         <artifactId>spring-boot-maven-plugin</artifactId>
30         <version>${spring-boot.version}</version>
31       </plugin>
32     </plugins>
33   </build>
```

專案內容

修改依賴範例：

在 `<dependency>` 區塊修改 [java mongodb-driver](#)

```
35   <dependencies>
36     <!-- Spring Boot Starter -->
37     <dependency>
38       <groupId>org.springframework.boot</groupId>
39       <artifactId>spring-boot-starter</artifactId>
40       <version>${spring-boot.version}</version>
41     </dependency>
42     <!-- Spring MVC Starter -->
43     <dependency>
44       <groupId>org.springframework.boot</groupId>
45       <artifactId>spring-boot-starter-web</artifactId>
46       <version>${spring-boot.version}</version>
47     </dependency>
48     <!-- JPA -->
49     <!--
50       <dependency>-->
51       <!--
52         <groupId>org.springframework.boot</groupId>-->
53         <!--
54           <artifactId>spring-boot-starter-data-jpa</artifactId>-->
55         <!--
56           <version>${spring-boot.version}</version>-->
57         </dependency>-->
58       <!-- thymeleaf -->
59       <dependency>
60         <groupId>org.springframework.boot</groupId>
61         <artifactId>spring-boot-starter-thymeleaf</artifactId>
62         <version>${spring-boot.version}</version>
63       </dependency>
64     <!--spring mongo jpa-->
65     <dependency>
66       <groupId>org.springframework.boot</groupId>
67       <artifactId>spring-boot-starter-data-mongodb</artifactId>
68       <version>${spring-boot.version}</version>
69     </dependency>
```

java mongo-driver 依賴的 jar 版本

MongoDB Documentation[← Back To MongoDB Drivers](#)**Java Sync**

v4.10 (current) ▾

Quick Start

[Quick Reference](#)[What's New](#)[Usage Examples](#)[Fundamentals](#)[API Documentation](#)[FAQ](#)[Connection Troubleshooting](#)[Issues & Help](#)**Compatibility**[Upgrade Driver Versions](#)[Migrate from the Legacy API](#)[View the Source](#)

The first column lists the driver version(s).

Compatibility Table Legend

Icon	Explanation
✓	All features are supported.
⊕	The Driver version will work with the MongoDB version, but not all new MongoDB features are supported.
No mark	The Driver version is not tested with the MongoDB version.

Java Driver Version	MongoDB 7.0	MongoDB 6.1	MongoDB 6.0	MongoDB 5.0	MongoDB 4.4	MongoDB 4.2	MongoDB 4.0	MongoDB 3.6	MongoDB 3.4
4.10	✓	✓	✓	✓	✓	✓	✓	✓	✓
4.9	⊕	✓	✓	✓	✓	✓	✓	✓	✓
4.8	⊕	✓	✓	✓	✓	✓	✓	✓	✓
4.7	⊕	⊕	✓	✓	✓	✓	✓	✓	✓
4.6	⊕	⊕	⊕	✓	✓	✓	✓	✓	✓
4.5	⊕	⊕	⊕	✓	✓	✓	✓	✓	✓
4.4	⊕	⊕	⊕	✓	✓	✓	✓	✓	✓
4.3	⊕	⊕	⊕	✓	✓	✓	✓	✓	✓
4.2	⊕	⊕	⊕	⊕	✓	✓	✓	✓	✓

替換 dependency 區塊並且重新 mvn install , 更換 driver 版本完成

*此 driver 只適用於 MongoDB , 並不適用 MongoRepository & MongoTemplate 故本次不更換

The screenshot shows the MongoDB Documentation website for the Java Sync driver. The left sidebar has a dropdown for 'Java Sync' currently set to 'v4.10 (current)'. The 'Quick Start' section is highlighted. The main content area starts with instructions to 'Install the Java Development Kit (JDK)' and 'Create the Project'. It provides Maven and Gradle code snippets for adding dependencies. A red box highlights the Maven code snippet.

Install the Java Development Kit (JDK)

Make sure that your system has JDK 8 or later installed. For more information on how to check your version of Java and install the JDK, see the [Oracle Overview of JDK Installation documentation](#).

Create the Project

This guide shows you how to add the MongoDB Java driver dependencies using Maven or Gradle. We recommend that you use an integrated development environment (IDE) such as IntelliJ IDEA or Eclipse IDE make it more convenient to configure Maven or Gradle to build and run your project.

If you are not using an IDE, see [Building Maven](#) or [Creating New Gradle Builds](#) for more information on how to set up your project.

Add MongoDB as a Dependency

If you are using [Maven](#), add the following to your `pom.xml` dependencies list:

```
<dependencies>
    <dependency>
        <groupId>org.mongodb</groupId>
        <artifactId>mongodb-driver-sync</artifactId>
        <version>4.10.2</version>
    </dependency>
</dependencies>
```

If you are using [Gradle](#), add the following to your `build.gradle` dependencies list:

```
dependencies {
    implementation 'org.mongodb:mongodb-driver-sync:4.10.2'
}
```

6. docker-compose.yml 內容

若想使用 Cluster 開啟對應區塊註解，並註解舊的區塊
執行對應指令即可，反之亦然

*注意 .yml/.yaml 有嚴格縮排限制

本質上 .yml 就是一種簡潔易懂的 .properties 因此需要小心 [檢查](#)

```
1 ##### Memo #####
2 #先啟動docker container 再這個.yml 路徑下使用docker-compose
3 #指令
4 #啟動且在背景執行
5 #docker-compose up -d
6 #關閉
7 #docker-compose stop
8 #刪除
9 #docker-compose rm
10 #關閉並刪除
11 #docker-compose down
12
```

常用指令

```
14 ##### SingleAlone With Auth #####
15 #version: '3.5'
16 services:
17 # mongodb_single:
18 #   image: mongo:6.0.6
19 #   container_name: mongodb_single1
20 #   ports:
21 #     - "17010:17010"
22 #   environment:
23 #     - MONGO_INITDB_ROOT_USERNAME=root
24 #     - MONGO_INITDB_ROOT_PASSWORD=root
25 #   command: mongod --auth --port 17010 --bind_ip_all
26
27
28 ##### Easy ReplicaSet Cluster #####
29 version: '3.5'
30 services:
31 mongo1:
32   container_name: mongo11
33   image: mongo:6.0.6
34   ports:
35     - 17011:17011
36   command: mongod --port 17011 --bind_ip_all --replSet myReplicaSet
37 mongo2:
38   container_name: mongo12
39   image: mongo:6.0.6
40   ports:
41     - 17012:17012
42   command: mongod --port 17012 --bind_ip_all --replSet myReplicaSet
43 mongo3:
44   container_name: mongo13
45   image: mongo:6.0.6
46   ports:
47     - 17013:17013
48   command: mongod --port 17013 --bind_ip_all --replSet myReplicaSet
```

單機

Cluster

7. application.yml 內容

* Spring 預設配置為
application.yml or application.properties

```
pom.xml (mmogo) x application.yml x docker-compose.yml x
1 ######
2 # 设置服务器端口
3 server:
4   port: 8085          服务port號，可自行更換
5
6 #####
7
8 spring:
9   # JSP 配置
10  #mvc:
11    # view:
12      #   prefix: /WEB-INF/templates/
13      #   suffix: .jsp
14    # thymeleaf 配置
15    thymeleaf:
16      cache: false # 禁用缓存，修改完实时展示数据
17      prefix: /WEB-INF/templates/ # 文件所在位置
18      suffix: .html # 后缀
19    web:
20      resources:
21        static-locations: /WEB-INF/static/ # 静态资源
22
23    # 配置数据库连接
24    #MySQL
25    datasource:
26      #   url: jdbc:mysql://localhost:3306
27      #   username: root
28      #   password: "!QAZ2wsx"
29
```

MongoDB Connect URI 修改

```
pom.xml (mmogo) × application.yml × docker-compose.yml ×
47
30     #####MongoDB
31     data:
32         mongodb:
33             database: demo
34             authentication-database: admin
35             ##### ver 2.4前用這個
36             host: 127.0.0.1
37             port: 27017
38             username: root
39             password: root
40             ##### ver 2.4以後用 uri string 連接 (Template & Repository & JDBC)
41             ##### local Standalone db 有密碼與無密碼連接方式
42             ##### DB啟動參考指令
43             ##### 有密碼：
44             mongod --auth --dbpath "/Users/teddylai/database/mongo/mongodb/data" --logpath "/Users/teddylai/database/mongo/mongodb/log/mongo.log" --fork
45             ##### 沒密碼：
46             mongod --dbpath "/Users/teddylai/database/mongo/mongodb/data" --logpath "/Users/teddylai/database/mongo/mongodb/log/mongo.log" --fork
47             # uri: mongodb://localhost:27017/demo
48             # uri: mongodb://root:root@127.0.0.1:27017/demo
49             ##### cloud atlas 生成的uri
50             uri: mongodb+srv://root:root@cloud-cluster0.prhwadb.mongodb.net/demo?retryWrites=true&w=majority
51             ##### docker Standalone & cluster
52             # uri: mongodb://localhost:27010/demo
53             # uri: mongodb://mongo1:27011,mongo2:27012,mongo3:27013/demo?replicaSet=myDockerSet
54             ##### docker-compose Standalone & cluster
55             # uri: mongodb://root:root@localhost:17010/demo?authSource=admin
56             # uri: mongodb://mongo11:17011,mongo12:17012,mongo13:17013/demo?replicaSet=myReplicaSet
57             ##### linux vm cluster
58             # uri: "mongodb://root:root@mongo0.replset.member:37010,mongo1.replset.member:37011,mongo2.replset.member:37012/demo?authSource=admin&replicaSet=mongoCluster"
59
60
```

uri 連線字串
若要更換數據源,打開對應uri並註解原本的即可

PostMan & Demo AP 操作

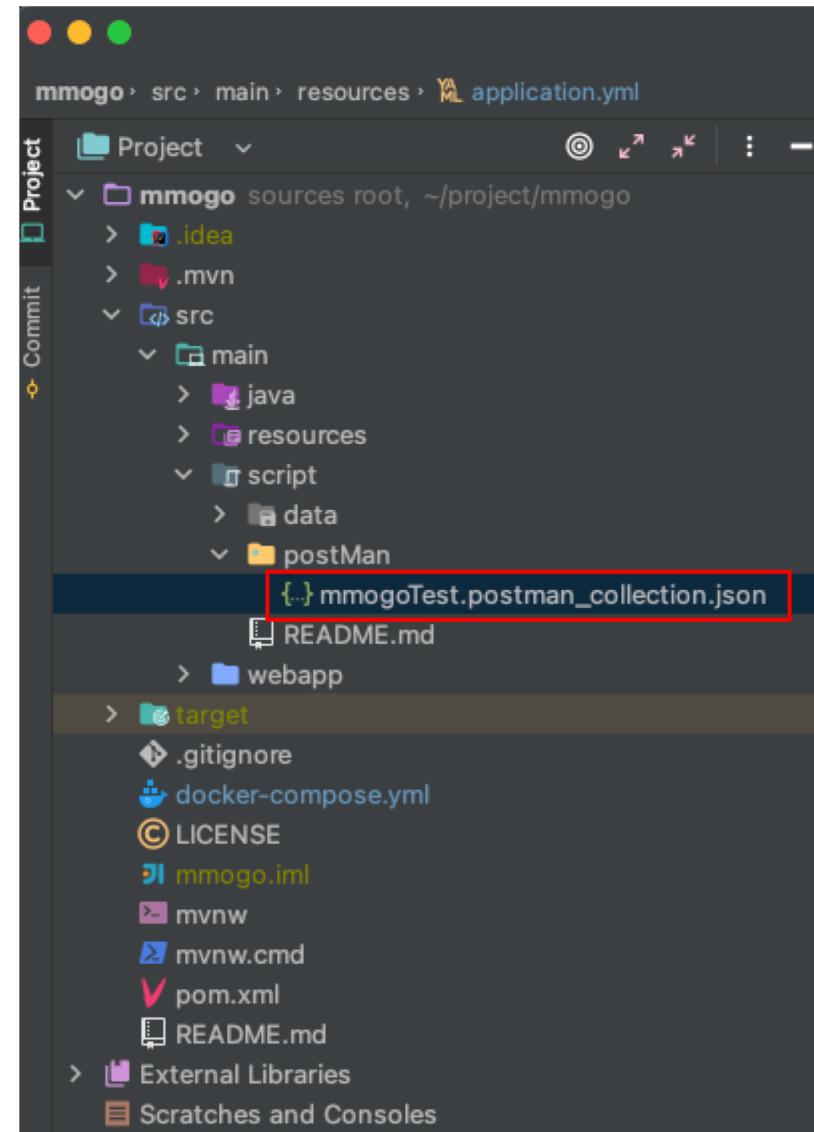
PostMan 優點：

1. 不需使用 Web 畫面操作即可操作 API
2. 後端預先開發 API 接口，自動化測試撰寫
3. Workspaces Collections 可導入與導出，支援團隊協作
4. 複製瀏覽器 API 交互，重現操作，可用於多種用途

1. [下載 PostMan](#)

2. 已導出的 PostMan Collections 位置

~/mmogo/src/main/script/postman/mmogoTest.postman_collection.json



3. Create Workplace

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, Environments, History, and a search bar labeled "Search workspaces". A red box highlights the "Create Workspace" button. The main workspace area is titled "mmogoTest" and contains a "Team Workspace" section. At the top right, there are various buttons like "Invite", "Share", "Fork", "Run", and "Save". Below the workspace title, it says "Created by You" and "Created on 13 Jun 2023, 4:42 PM". A link "View complete documentation →" is also present.

Home Workspaces API Network Explore

Search Postman

Mongo

Collections

Environments

History

Search workspaces

Create Workspace

Recently visited

> mmogoTest

Mongo

Team Workspace

GPO & CB

More workspaces

No workspaces found

View all workspaces →

mmogoTest

POST json_select POST json_delete POST json_update

mmogoTest

Share Fork 0 Run Save

Created by You

Created on 13 Jun 2023, 4:42 PM

View complete documentation →

Create your workspace

Get the most out of your workspace with a template.

 Blank workspace ✓

Explore our templates

 API Demos

 API Development

 API Testing

 API Security

 Incident Response

 Cloud Infrastructure Management

Next

Cancel

Step 1 of 2

Create your workspace

Name

MongoTest

Summary

Who can access your workspace?

Personal

Only you can access

Private

Only invited team members can access

Team

All team members can access

Partner

Only invited partners and team members can access

Public

Everyone can view

Create

Back

Step 2 of 2

4. Import Collections

Select files “mmogoTest.postman_collection.json” to import

The screenshot shows the Postman application interface with a dark theme. On the left, the sidebar includes 'Collections' (selected), 'Environments', and 'History'. The main workspace displays a collection named 'MongoTest' containing a single folder 'My first collection' with two sub-folders: 'First folder inside collection' and 'Second folder inside collection', each containing several requests. A modal dialog titled 'Import' is open in the center-right. It contains a 'Workspace description' section with a note about adding quick access information, a 'Pinned collections' section for discoverability, and a large input field with a blue border labeled 'Paste cURL, Raw text or URL...'. Below this field is a tip: 'Tip : You can also paste cURL in the request bar to import' with a 'Dismiss' button. A red box highlights a 'Drop anywhere to import' area with a checkmark icon and the text 'Or select files or folders'. At the bottom of the dialog are icons for GitHub, Bitbucket, and Google Drive, followed by 'Other Sources' and a link to 'Learn more about importing data'.

5. PostMan 介紹

寫好的 Script

參數

協議類型 (大部分是 json)

AP Reponse 區塊

Click Send to get a response

```
1 {  
2   "dbName": "demo",  
3   "collection": "test",  
4   "type": "insert",  
5   "documents": [  
6     {  
7       "title": "hello",  
8       "description": "database",  
9       "likes": 100,  
10      "by": "teddy"  
11    }]  
12 }
```

6. Post 操作 (以 Select 為例)

The screenshot shows the Postman application interface for a MongoDB test collection named "MongoTest". A POST request titled "POST json_select" is selected. The "Body" tab is active, showing a JSON document with a red box around it. The JSON content is:

```
1 {  
2   "db": "demo",  
3   "collection": "employee",  
4   "type": "select",  
5   "filter": {"job": "HR"}  
6 }
```

A red arrow points from the "Send" button to the JSON body area. Another red box highlights the "filter" field. A red annotation text "修改想查詢的條件" (Modify the query conditions) is placed near the JSON body.

In the bottom right corner of the main window, there is a red box highlighting the "AP 回應 DATA" (API Response Data) section, which displays the results of the query:

```
1 Document{{_id=649266c366049d21b334ca74, first_name=CINDY, last_name=PAI, job=HR, salary=1000, internship=false, regular_date=Wed Jun 21 10:56:03 CST 2023}}  
2 Document{{_id=64b10a0573bbe692f328a804, first_name=CINDY, last_name=PAI, job=HR, salary=1000, internship=false, regular_date=Fri Jul 14 16:40:37 CST 2023}}
```

7. Web CRUD 操作

輸入本機 IP 與服務 Port :

<http://localhost:8085/> or <http://127.0.0.1:8085/>

MongoDB Demo Portal

查詢欄位		查詢	新增				
查詢內容							
firstName & lastName & job 支援模糊查詢 , regularDate : yyyy/MM/dd HH:mm:ss							
id	firstName	lastName	job	salary	internship	regularDate	operate
649266c366049d21b334ca70	CHRIS1	LIN	Project Manager	9500	false	2020/05/18 08:00:00	<button>修改</button> <button>刪除</button>
649266c366049d21b334ca73	AMY	WANG	PART TIME EMPLOYEE	500	true	2011/06/18 22:10:30	<button>修改</button> <button>刪除</button>
649266c366049d21b334ca74	CINDY	PAI	HR	1000	false	2023/06/21 10:56:03	<button>修改</button> <button>刪除</button>
649266c366049d21b334ca72	BOSS	HUGO	BIG BOSS	50000	false	2000/10/01 08:00:00	<button>修改</button> <button>刪除</button>

8. Web CRUD With PostMan

開啟瀏覽器開發工具，打開 Network 查找 API 接口

MongoDB Demo Portal

salary

查詢內容

id	firstName	lastName	job	salary	internship
649266c366049d21b334ca73	AMY	WANG	PART TIME EMPLOYEE	500	true
64b10a0573bbe692f328a803	AMY	WANG	PART TIME EMPLOYEE	500	true

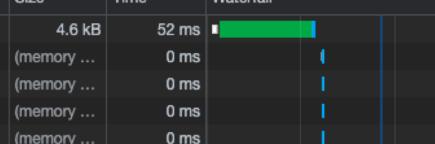
Network Performance Memory Application >>

Search Console Sources Performance insights Network Performance Memory Application >>

Filter Preserve log Disable cache No throttling Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

Has blocked cookies Blocked Requests 3rd-party requests

10 ms 20 ms 30 ms 40 ms 50 ms 60 ms 70 ms 80 ms 90 ms 100 ms 110 ms

Name	Status	Type	Initiator	Size	Time	Waterfall
hello?data=%7B%22selectField%22%... bootstrap.min.css bootstrap-grid.min.css bootstrap-reboot.min.css bootstrap.bundle.min.js bootstrap.min.js code.jquery.com_jquery-3.7.0.min.js employee.js frontend.bundle.rollup.js	200 200 200 200 200 200 200 200	document stylesheet stylesheet stylesheet script script script script	:8085/hello?data=... :8085/hello?data=... :8085/hello?data=... :8085/hello?data=... :8085/hello?data=... :8085/hello?data=... :8085/hello?data=... :8085/hello?data=...	4.6 kB 0 ms 0 ms 0 ms 0 ms 0 ms 0 ms 0 ms	52 ms 0 ms 0 ms 0 ms 0 ms 0 ms 0 ms 0 ms	

9 requests 16.1 kB transferred 408 kB resources Finish: 104 ms DOMContentLoaded: 91 ms Load: 136 ms

對著 API 右鍵選擇

Copy With Curl (bash)

The screenshot shows the Chrome DevTools Network tab with a list of network requests. A context menu is open over a request labeled "hello?data=%7B%22selectF". The menu is organized into several sections:

- Copy**:
 - Copy link address
 - Copy request headers
 - Copy response headers
 - Copy response
 - Copy stack trace
- Copy as**:
 - Copy as PowerShell
 - Copy as fetch
 - Copy as Node.js fetch
 - Copy as cURL** (highlighted with a red box)
 - Copy all as PowerShell
 - Copy all as fetch
 - Copy all as Node.js fetch
 - Copy all as cURL
 - Copy all as HAR

The main content area of the DevTools shows a table titled "goDB Demo Portal" with some sample data:

firstName	lastName	job	salary
WANG	WANG	PART TIME EMPLOYEE	500
WANG	WANG	PART TIME EMPLOYEE	500

At the bottom of the DevTools interface, there are links for "Console" and "What's New", and a note about "Highlights from the Chrome 114 update".

獲得類似如此的內容再匯入到 PostMan

```
curl 'http://127.0.0.1:8085/hello?data=%7B%22selectField%22%3A%22salary%22%2C%22selectContent%22%3A%22500%22%7D' \
-H 'Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.7' \
-H 'Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7,zh-CN;q=0.6' \
-H 'Connection: keep-alive' \
-H 'DNT: 1' \
-H 'Referer: http://127.0.0.1:8085/hello' \
-H 'Sec-Fetch-Dest: document' \
-H 'Sec-Fetch-Mode: navigate' \
-H 'Sec-Fetch-Site: same-origin' \
-H 'Sec-Fetch-User: ?1' \
-H 'Upgrade-Insecure-Requests: 1' \
-H 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0
Safari/537.36' \
-H 'sec-ch-ua: "Not.A/Brand";v="8", "Chromium";v="114", "Google Chrome";v="114"' \
-H 'sec-ch-ua-mobile: ?0' \
-H 'sec-ch-ua-platform: "macOS"' \
--compressed
```

Mongo

New Import



Collections



mmongoTest

POST json_insert

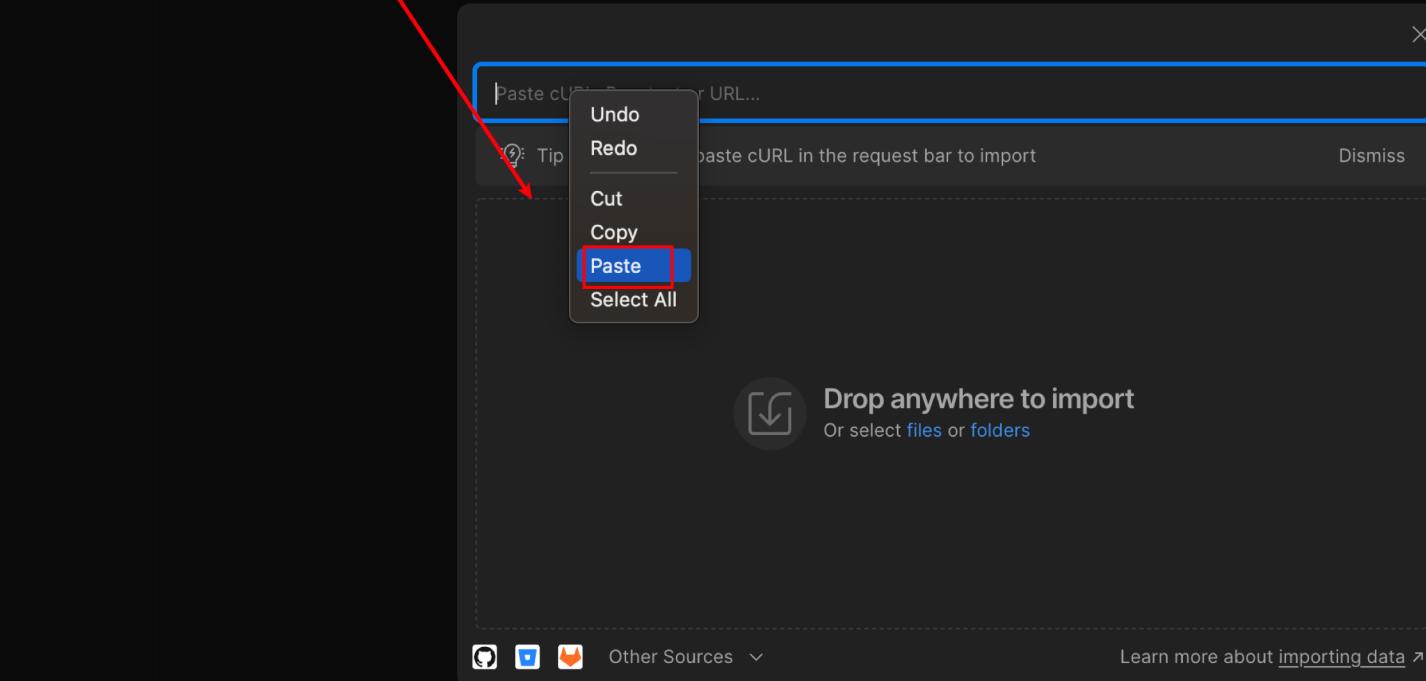
POST json_select

POST json_delete

POST json_update

Environments

History



選擇送出即可重現 Request

The screenshot shows the Postman application interface. A red box highlights the URL bar containing the request URL: `http://127.0.0.1:8085/hello?data=%7B%22selectField%22%3A%22salary%22%2C%22selectContent%22%3A%22500%22%7D`. Another red arrow points from this URL to the 'Send' button in the top right corner of the request details panel. The 'Params' tab is selected, showing a single parameter named 'data' with the value `%7B%22selectField%22%3A%22salary%22%2C%22selectContent%22%3A%22500%22%7D`. The 'Body' tab is also visible below. The response section at the bottom shows a status of `200 OK`, a time of `125 ms`, and a size of `4.52 KB`. The response body is displayed in 'Pretty' format, showing an HTML table row (`<tr>`) with several data cells (`<td>`) and two buttons (`<button>`). A large red box highlights this entire row.

```
http://127.0.0.1:8085/hello?data=%7B%22selectField%22%3A%22salary%22%2C%22selectContent%22%3A%22500%22%7D
```

Send

Key	Value	Description
data	%7B%22selectField%22%3A%22salary%22%2C%22selectContent%22%3A%22500%22%7D	
Key	Value	Description

Status: 200 OK Time: 125 ms Size: 4.52 KB

```
Pretty Raw Preview Visualize HTML
```

```
<tr>
<td>649266c366049d21b334ca73</td>
<td>AMY</td>
<td>WANG</td>
<td>PART TIME EMPLOYEE</td>
<td>500</td>
<td>true</td>
<td>2011/06/18 22:10:30</td>
<td>
<button class="btn btn-outline-secondary" type="button" onclick="updateEmployeePage("649266c366049d21b334ca73")">修改</button>
</td>
<td>
<button class="btn btn-dark" onclick="deleteEmployee("649266c366049d21b334ca73")" type="button">刪除</button>
</td>
</tr>
<tr>
<td>64b16a0573bbe692f328a803</td>
<td>AMY</td>
<td>WANG</td>
<td>PART TIME EMPLOYEE</td>
<td>500</td>
<td>true</td>
<td>2011/06/18 22:10:30</td>
<td>
```

MongoDB Community Server & Atlas 使用

MongoDB Community

1. 安裝 MongoDB Community Server 6.0.6 & MongoDB Shell

The screenshot shows the MongoDB website's product selection interface on the left and the MongoDB Shell download page on the right.

Left Side (Product Selection):

- MongoDB Atlas
- MongoDB Enterprise Advanced
- MongoDB Community Edition** (highlighted with a red box)
- Tools
 - MongoDB Shell** (highlighted with a red box)
 - MongoDB Compass (GUI) (highlighted with a blue box)
- Atlas CLI
- Atlas Kubernetes Operator
- MongoDB CLI for Cloud Manager and Ops Manager
- MongoDB Cluster-to-Cluster Sync
- Relational Migrator
- MongoDB Database Tools
- MongoDB Connector for BI

Right Side (MongoDB Shell Download Page):

TOOLS

MongoDB Shell Download

MongoDB Shell is the quickest way to connect to (and work with) MongoDB. Easily query data, configure settings, and execute other actions with this modern, extensible command-line interface – replete with syntax highlighting, intelligent autocomplete, contextual help, and error messages.

Note: MongoDB Shell is an open source (Apache 2.0), standalone product developed separately from the MongoDB Server.

Learn more

Version
1.10.1

Platform
MacOS 64-bit (10.14+)

安裝完成放進自定義資料夾路徑 (這裡以~/database/mongo 為例)

打開 Mongo Shell CLI 命令行指令

~/database/mongo/mongosh-1.9.1-darwin-arm64/bin/mongosh

啟動 MongoDB Daemon Process 指令

~/database/mongo/mongodb/bin/mongod

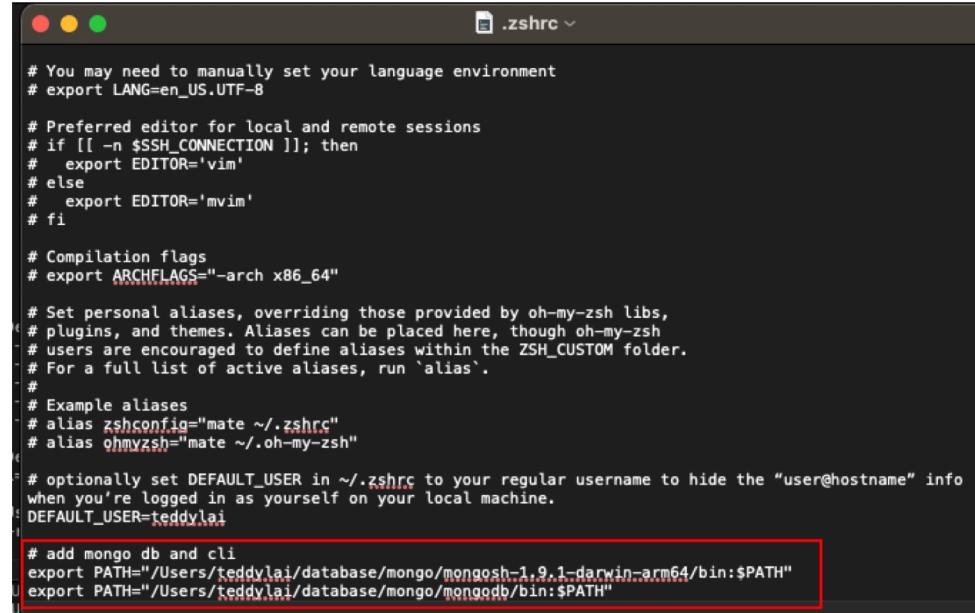
2.修改環境變數 方便在其他路徑也可運行 (mac os)

open ~/.zshrc

```
# add mongo db and cli
```

```
export PATH="/Users/teddylai/database/mongo/mongosh-1.9.1-darwin-arm64/bin:$PATH"
```

```
export PATH="/Users/teddylai/database/mongo/mongodb/bin:$PATH"
```



```
# You may need to manually set your language environment
# export LANG=en_US.UTF-8

# Preferred editor for local and remote sessions
# if [[ -n $SSH_CONNECTION ]]; then
#   export EDITOR='vim'
# else
#   export EDITOR='mvim'
# fi

# Compilation flags
# export ARCHFLAGS="-arch x86_64"

# Set personal aliases, overriding those provided by oh-my-zsh libs,
# plugins, and themes. Aliases can be placed here, though oh-my-zsh
# users are encouraged to define aliases within the ZSH_CUSTOM folder.
# For a full list of active aliases, run `alias`.
#
# Example aliases
# alias zshconfig="mate ~/.zshrc"
# alias ohmyzsh="mate ~/.oh-my-zsh"

# optionally set DEFAULT_USER in ~/.zshrc to your regular username to hide the "user@hostname" info
# when you're logged in as yourself on your local machine.
DEFAULT_USER=teddylai

# add mongo db and cli
export PATH="/Users/teddylai/database/mongo/mongosh-1.9.1-darwin-arm64/bin:$PATH"
export PATH="/Users/teddylai/database/mongo/mongodb/bin:$PATH"
```

3.在背景執行MongoDB 參考指令

沒密碼：

```
mongod --dbpath "/Users/teddylai/database/mongo/mongodb/data" --logpath  
"/Users/teddylai/database/mongo/mongodb/log/mongo.log" –fork  
//Add user  
mongosh  
    use admin  
    db.createUser({  
        user: "root",  
        pwd: "root",  
        roles: [ "root"]  
    })  
    db.auth("root","root")
```

有密碼：

```
mongod --auth --dbpath "/Users/teddylai/database/mongo/mongodb/data" --logpath  
"/Users/teddylai/database/mongo/mongodb/log/mongo.log" –fork
```

檢查mongod是否有運行

```
ps -ef | grep mongod
```

```
mongod --version
```

啟動mongo shell

```
mongosh --host 127.0.0.1 --port 27017
```

關閉mongod

```
db.shutdownServer()
```

MongoDB Atlas

MongoDB Atlas M0 優點：

無需信用卡完全免費使用，提供三大公有雲托管，免費管理和監控工具，可創建多個免費的M0 資料庫

MongoDB Atlas M0 缺點：

不支援Replica Set & Sharding，出現故障時，無法自動進行故障轉移和數據恢復，DB 有容量限制

1. [參考課程\(有中文字幕\)](#) 創建M0 DataBase

2. 取得Connect URI

```
mongosh "mongodb+srv://cloud-cluster0.prhwadb.mongodb.net/" --apiVersion 1 --username root
```

The screenshot shows the MongoDB Atlas interface. On the left, there's a sidebar with various project settings like Deployment, Services, Security, and Database Access. The main area is titled 'Database Deployments' and shows a single cluster named 'cloud-cluster0'. A red box highlights the 'Connect' button next to the cluster name. Below the cluster details, there's an 'Enhance Your Experience' section with an 'Upgrade' button. On the right, there's a 'Connect to cloud-cluster0' dialog box. It has three steps: 'Set up connection security' (done), 'Choose a connection method' (selected), and 'Connect' (not yet reached). Under 'Connect to your application', there's a 'Drivers' section. Under 'Access your data through tools', there are several options: 'Compass', 'Shell' (highlighted with a red box), 'MongoDB for VS Code', and 'Atlas SQL'. At the bottom of the dialog are 'Go Back' and 'Close' buttons.

```
mongosh "mongodb+srv://cloud-cluster0.prhwadb.mongodb.net/" --apiVersion 1 --username root
Enter password: ****
Current Mongosh Log ID: 64b0f12373bbe692f328a7fd
Connecting to: mongodbsrv://<credentials>@cloud-cluster0.prhwadb.mongodb.net/?appName=mongosh+1.9.1
Using MongoDB: 6.0.6 (API Version 1)
Using Mongosh: 1.9.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/
Atlas atlas-547994-shard-0 [primary] test>
```

使用Docker 快速搭建 MongoDB Single & Cluster

Docker 相對傳統VM優勢：

- 1.輕量化,啟動資源速度相較傳統VM少很多, mac windows linux 皆有支援
若是安裝在Linux會自動管理Docker container資源調度
- 2.自己的MongoDB Container config完成後可以打包成Image 做到重複使用, 部屬多個環境不需要再重新配置
配合 k8s 可以實現 DevOps 自動部署

使用教學：

1.自行安裝好Docker

2.拉官方的 MongoDB image 這裡版本用6.0.6

```
docker pull mongo:6.0.6
```

3.創建network && 啟動 container

單機

```
docker network create mongoSingle
docker run -i -p 27010:27010 -t --name mongodb_single --network mongoSingle mongo:6.0.6 bash
mongod --port 27010 --logpath "/var/log/mongodb/mongod.log" --dbpath "/data/db" --fork --bind_ip_all
mongosh --port 27010
```

Replica Set

```
docker network create mongoCluster
docker run -d -p 27011:27011 --name mongo1 --network mongoCluster mongo:6.0.6 mongod --replSet myDockerSet --bind_ip localhost,mongo1 --port 27011 --logpath "/var/log/mongodb/mongod.log" --dbpath "/data/db" --bind_ip_all
docker run -d -p 27012:27012 --name mongo2 --network mongoCluster mongo:6.0.6 mongod --replSet myDockerSet --bind_ip localhost,mongo2 --port 27012 --logpath "/var/log/mongodb/mongod.log" --dbpath "/data/db" --bind_ip_all
docker run -d -p 27013:27013 --name mongo3 --network mongoCluster mongo:6.0.6 mongod --replSet myDockerSet --bind_ip localhost,mongo3 --port 27013 --logpath "/var/log/mongodb/mongod.log" --dbpath "/data/db" --bind_ip_all
```

1. Docker Replica Set 初始指令

增加節點

```
mongosh --port 27011
```

```
use admin
```

```
rs.initiate()
```

```
rs.conf()
```

```
rs.add("mongo2:27012")
```

```
rs.add("mongo3:27013")
```

```
rs.status()
```

```
cfg = rs.conf()  
cfg.members[0].host = "mongo1:27011"  
rs.reconfig(cfg)
```

備庫查看同步

```
mongosh --port 27012
```

```
//給予session 權限 ·
```

```
db.getMongo().setReadPref("secondary")
```

```
use demo
```

```
db.employee.find({})
```

2.Host PC Add Docker Hosts

修改主機 host

```
sudo vim /etc/hosts
```

```
127.0.0.1 mongo1
```

```
127.0.0.1 mongo2
```

```
127.0.0.1 mongo3
```

Docker-Compose.yml With Application

Docker-Compose 優點：

1. 簡化多個容器Container的管理與配置：Docker Compose 允許你使用一個單獨的 YAML 文件來定義和配置多個容器信息。
並將容器定義集中在一個地方，更容易管理整個應用。
2. 一鍵部署和擴展：Docker Compose 提供了簡單的命令用來一鍵部署整個Service。

常用指令：

#啟動且在背景執行	#關閉	#刪除	#關閉並刪除
<code>#docker-compose up -d</code>	<code>#docker-compose stop</code>	<code>#docker-compose rm</code>	<code>#docker-compose down</code>

使用教學：

1. CD 至 docker-compose.yml 所在位置 (通常是專案根目錄)

2. 修改 docker-compose.yml 內容並下啟動指令

```
cd project/mmogo
```

```
docker-compose up -d
```

1. Docker-Compose Replica Set 初始指令

增加節點

```
mongosh --port 17011
```

```
use admin
```

```
rs.initiate()
```

```
rs.conf()
```

```
rs.add("mongo12:17012")
```

```
rs.add("mongo13:17013")
```

```
rs.status()
```

```
cfg = rs.conf()  
cfg.members[0].host = "mongo11:17011"  
rs.reconfig(cfg)
```

備庫查看同步

```
mongosh --port 17012
```

```
//給予session 權限 ·
```

```
db.getMongo().setReadPref("secondary")
```

```
use demo
```

```
db.employee.find({})
```

2.Host PC Add Docker-Compose Hosts

修改主機 host

```
sudo vim /etc/hosts
```

```
127.0.0.1 mongo11
```

```
127.0.0.1 mongo12
```

```
127.0.0.1 mongo13
```

Ubuntu 22.04 建置 MongoDB Replica Set

建置指令

apt 套件update & install

```
sudo apt-get update  
sudo apt-get install r-base  
sudo apt-get install r-recommended  
sudo apt-get install gnupg curl
```

取得mongo公鑰

```
curl -fsSL https://pgp.mongodb.com/server-6.0.asc \  
sudo gpg -o /usr/share/keyrings/mongodb-server-6.0.gpg \  
--dearmor
```

安裝 mongo

```
sudo apt --fix-broken install  
echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-6.0.gpg ] https://repo.mongodb.org/apt/ubuntu/jammy/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list  
sudo apt-get update
```

改用特定mongodb 版本

```
sudo apt install -y mongodb-org=6.0.6 mongodb-org-database=6.0.6 mongodb-org-server=6.0.6 mongodb-org-mongos=6.0.6  
mongodb-org-tools=6.0.6
```

安裝完成後 常用路徑

```
apt package /usr/bin/mongo*  
yml config file /etc/mongod.conf  
default data directory /var/lib/mongodb  
default log directory /var/log/mongodb
```

啟動mongod && 確保開機重啟

```
sudo systemctl enable mongod  
sudo systemctl start mongod
```

新增root user

```
mongosh  
use admin
```

```
db.createUser({  
    user: "root",  
    pwd: "root",  
    roles: [ "root" ]  
})
```

```
db.auth("root","root")
```

```
db.shutdownServer()
```

增加三台 vm 與 ap 的 hosts

sudo vim /etc/hosts

```
10.211.55.12 mongo0.replset.member
10.211.55.9 mongo1.replset.member
10.211.55.10 mongo2.replset.member
```

生成ssl key 並同步

//選一台生成

```
openssl rand -base64 756 | head -c 756 > /var/lib/mongodb/mongodb-keyfile
```

//複製到另外兩台

```
scp /var/lib/mongodb/mongodb-keyfile teddy@mongo1.replset.member:/tmp
mv /tmp/mongodb-keyfile /var/lib/mongodb/mongodb-keyfile
scp /var/lib/mongodb/mongodb-keyfile teddy@mongo2.replset.member:/tmp
mv /tmp/mongodb-keyfile /var/lib/mongodb/mongodb-keyfile
```

//給予每台讀寫權限

```
chmod 600 /var/lib/mongodb/mongodb-keyfile
chown mongodb.mongodb /var/lib/mongodb/mongodb-keyfile
```

mongod.conf.yml config 修改如圖

sudo vim /etc/mongod.conf

port 分別改成 37010 , 37011 , 37012
並開啟 bindIp && security

```
ubuntu vm
# mongod.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# Where and how to store data.
storage:
  dbPath: /var/lib/mongodb
#   engine:
#     wiredTiger:

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# network interfaces
net:
  port: 37010
  bindIp: 0.0.0.0

# how the process runs
processManagement:
  timeZoneInfo: /usr/share/zoneinfo

#security:
  security:
    authorization: enabled
    keyFile: /var/lib/mongodb/mongodb-keyfile

#operationProfiling:

#replication:
  replication:
    replSetName: mongoCluster

#sharding:

## Enterprise-Only Options:

#auditLog:

#snmp:
~
```

修改完測試 && 重啟

reboot

systemctl start mongod

mongosh --host 127.0.0.1 --port 3701x

use admin

db.auth("root", passwordPrompt())

Replica Set init && Add node

//挑一台操作 init cluster

rs.initiate()

rs.conf()

rs.add("mongo1.replset.member:37011")

rs.add("mongo2.replset.member:37012")

rs.status()

//host0 域名無法辨別，導致 statusStr: 'STARTUP'

//需要重新配置下第一台 hostname

cfg = rs.conf()

cfg.members[0].host = "mongo0.replset.member:37010"

rs.reconfig(cfg)

```
ubuntu vm
lastElectionDate: ISODate("2023-07-11T06:29:07.712Z"),
electionTerm: Long("1"),
lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1689056947, i: 1 }), t: Long("-1") },
lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1689056947, i: 1 }), t: Long("-1") },
numVotesNeeded: 1,
priorityAtElection: 1,
electionTimeoutMillis: Long("10000"),
newTermStartDate: ISODate("2023-07-11T06:29:07.725Z"),
wMajorityWriteAvailabilityDate: ISODate("2023-07-11T06:29:07.733Z")
},
members: [
{
_id: 0,
name: 'ubuntu:37010',
health: 1,
state: 1,
stateStr: 'PRIMARY',
uptime: 3982,
optime: { ts: Timestamp({ t: 1689060768, i: 1 }), t: Long("1") },
optimeDate: ISODate("2023-07-11T07:32:48.000Z"),
lastAppliedWallTime: ISODate("2023-07-11T07:32:48.566Z"),
lastDurableWallTime: ISODate("2023-07-11T07:32:48.566Z"),
syncSourceHost: '',
syncSourceId: -1,
infoMessage: '',
electionTime: Timestamp({ t: 1689056947, i: 2 }),
electionDate: ISODate("2023-07-11T06:29:07.000Z"),
configVersion: 7,
configTerm: 1,
self: true,
lastHeartbeatMessage: ''
},
{
_id: 2,
name: 'mongo2.replset.member:37012',
health: 1,
state: 0,
stateStr: 'STARTUP',
uptime: 1355,
optime: { ts: Timestamp({ t: 0, i: 0 }), t: Long("-1") },
optimeDurable: { ts: Timestamp({ t: 0, i: 0 }), t: Long("-1") },
optimeDate: ISODate("1970-01-01T00:00:00.000Z"),
optimeDurableDate: ISODate("1970-01-01T00:00:00.000Z"),
lastAppliedWallTime: ISODate("1970-01-01T00:00:00.000Z"),
lastDurableWallTime: ISODate("1970-01-01T00:00:00.000Z"),
lastHeartbeat: ISODate("2023-07-11T07:32:47.002Z"),
lastHeartbeatRecv: ISODate("1970-01-01T00:00:00.000Z"),
pingMs: Long("0"),
lastHeartbeatMessage: '',
syncSourceHost: '',
syncSourceId: -1,
infoMessage: '',
configVersion: -2,
configTerm: -1
}
]
```

Replica Set Connect URI Sample (可直接用在 Mongo Compass && AP Config)

Mongo Compass Connect URI :

```
mongodb://root:root@mongo0.replset.member:37010,mongo1.replset.member:37011,mongo2.replset.member:37012/?authSource=admin&replicaSet=mongoCluster
```

JAVA AP Config:

```
uri:"mongodb://root:root@mongo0.replset.member:37010,mongo1.replset.member:37011,mongo2.replset.member:37012/demo?authSource=admin&replicaSet=mongoCluster"
```

Mongo Shell 連線：

```
mongosh  
"mongodb://root:root@mongo0.replset.member:37010,mongo1.replset.member:37011,mongo2.replset.member:37012/?authSource=admin&replicaSet=mongoCluster"
```

備庫設置讀取權限

```
//單次  
db.employee.find({ }).readPref( "secondary")  
//給予session 權限  
db.getMongo().setReadPref("secondary")
```

Ubuntu 參考指令

Uninstall MongoDB

```
apt-get purge mongodb-org*
rm -r /var/log/mongodb /var/lib/mongodb
systemctl disable mongod
service mongod status
rm /tmp/mongodb-*.sock
```

See Service PID

```
ps aux | grep "mongod"
//盡量用 (systemctl stop mongod)
kill -9 xxxx
```

檢查mongod 狀態 & & 通過配置文件啟動 MongoDB 服務器

```
mongod -f /etc/mongod.conf --repair
mongod -f /etc/mongod.conf
sudo systemctl status mongod
sudo systemctl start mongod
sudo systemctl restart mongod
```

mongod 手動重啟參考指令

```
mongod -f /etc/mongod.conf --repair
mongod --config /etc/mongod.conf
mongod --fork --logpath /var/log/mongodb/mongod.log --dbpath /var/lib/mongodb
mongod --auth --config /etc/mongod.conf --fork --logpath /var/log/mongodb/mongod.log --dbpath /var/lib/mongodb
```

參考網站

Jetbrains IntelliJ 下載

<https://www.jetbrains.com/idea/download>

IntelliJ 編碼修改

https://blog.csdn.net/m0_38132361/article/details/80628203

IntelliJ 常用快捷鍵

<https://einverne.gitbook.io/intellij-idea-tutorial/keymap-mac-introduce>

JAVA JDK 17 Downloads

<https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>

JDK && Java Home 設定

<http://www.tastones.com/zh-tw/stackoverflow/java-language/installing-java-standard-edition>

PostMan 下載

<https://www.postman.com/downloads/>

yml / xml 檢查

<https://mageddo.com/tools/yaml-converter>

MongoDB 下載資源列表

<https://www.mongodb.com/try/download/tools>

MongoDB University

<https://learn.mongodb.com/>

MongoDB Atlas 教學

<https://learn.mongodb.com/learn/course/getting-started-with-mongodb-atlas/lesson-2-creating-and-deploying-at-atlas-cluster/learn?client=customer&page=1>

MongoDB Java Driver 兼容列表

<https://www.mongodb.com/docs/drivers/java/sync/current/compatibility/>

Docker 基本操作

<https://github.com/twtrubiks/docker-tutorial>

Docker mongodb cluster 建置

<https://www.mongodb.com/compatibility/deploying-a-mongodb-cluster-with-docker>

Ubuntu MongoDB Community or Enterprise Insatall

<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/>

<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-enterprise-on-ubuntu/>