

# Java EE Application 1z0-900

講師:江聖榮

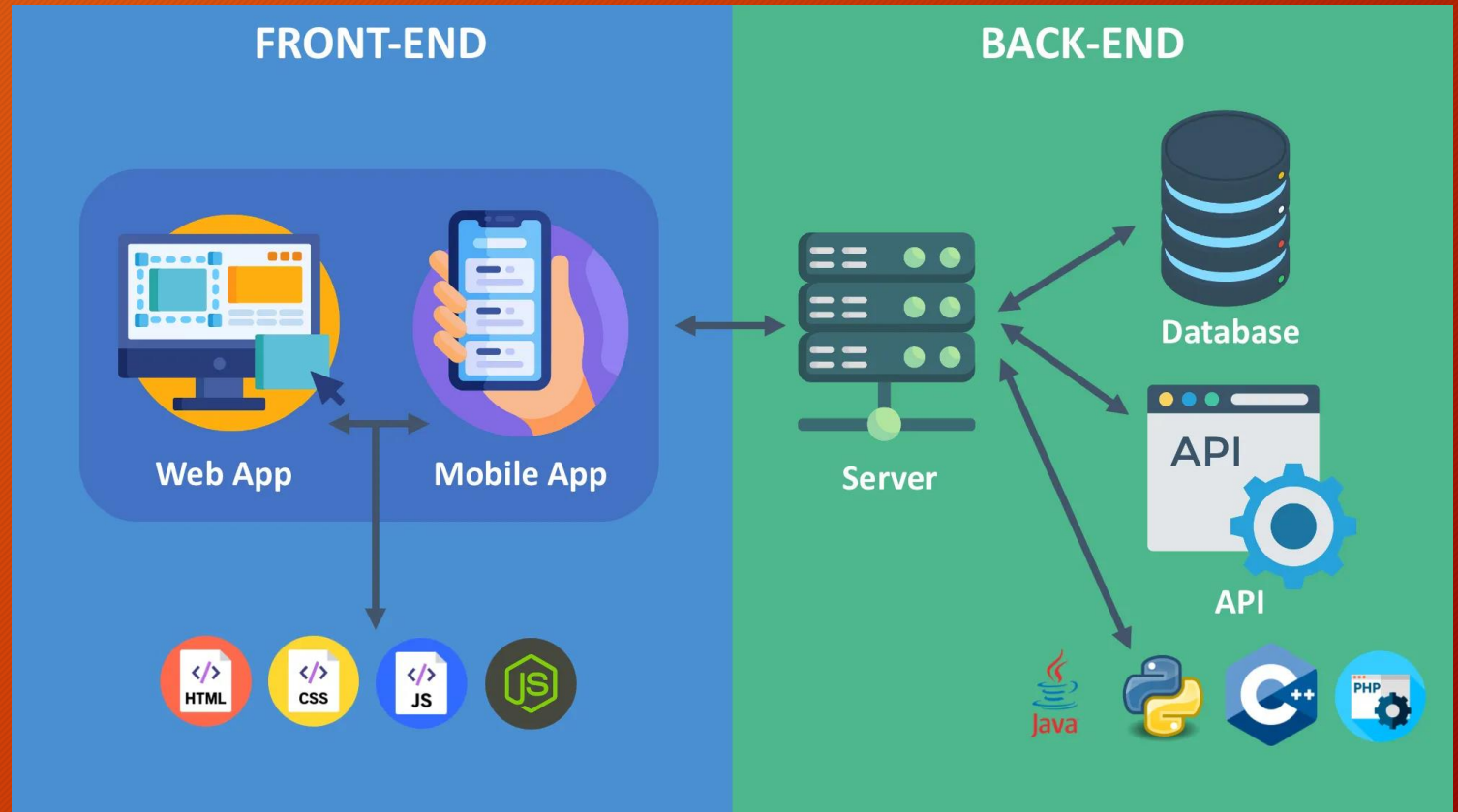
# Mysql database Server 8.0

- Schema->專案名
- Table
- Ex:會員管理
  - Id
  - Name
  - Username
  - Password
  - Address
  - phone

id	name	username	password	address	phone
Int Pk Nn ai	varchar	varchar	varchar	varchar	varchar

# JDBC(Java DataBase Connection)

- backEnd後端
- Class.forName
- Java.sql.\*
- DriverManager
- SQL
- Statement
- PreparedStatement
- ResultSet





# 檢查是否有database廠商提供的Driver

- Class.forName
- Mysql 8.0
- com.mysql.cj.jdbc.Driver
- 非常重要

## MySQL Product Archives

MySQL Connector/J (Archived Versions)

Product Version: 8.4.0

Operating System: Platform Independent

Platform Independent (Architecture Independent), ZIP Archive

Mar 12, 2024

4.9M

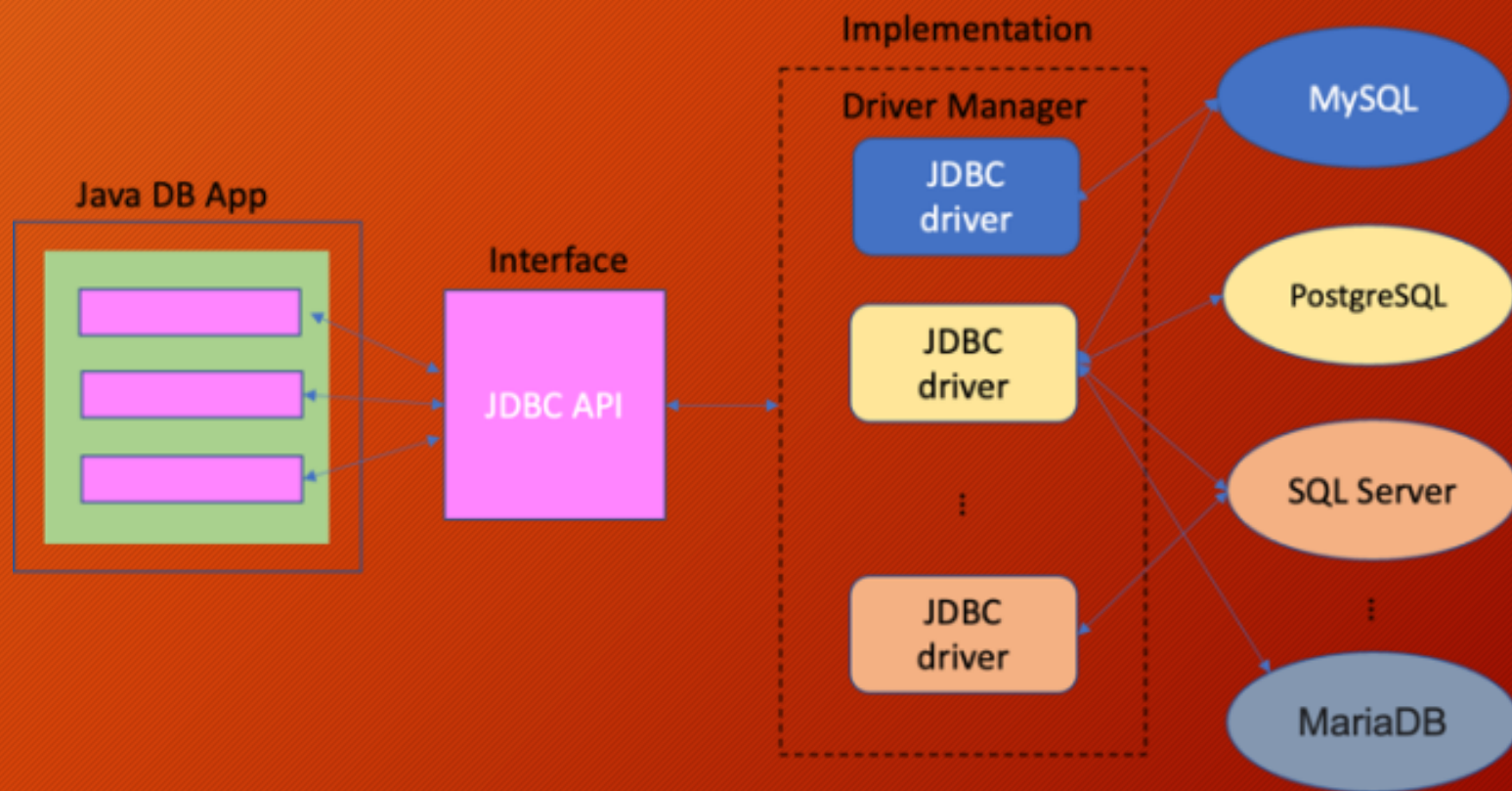
Download

(mysql-connector-j-8.4.0.zip)

MD5: 5d426b639f7b4c314492edff07c615dc | [Signature](#)

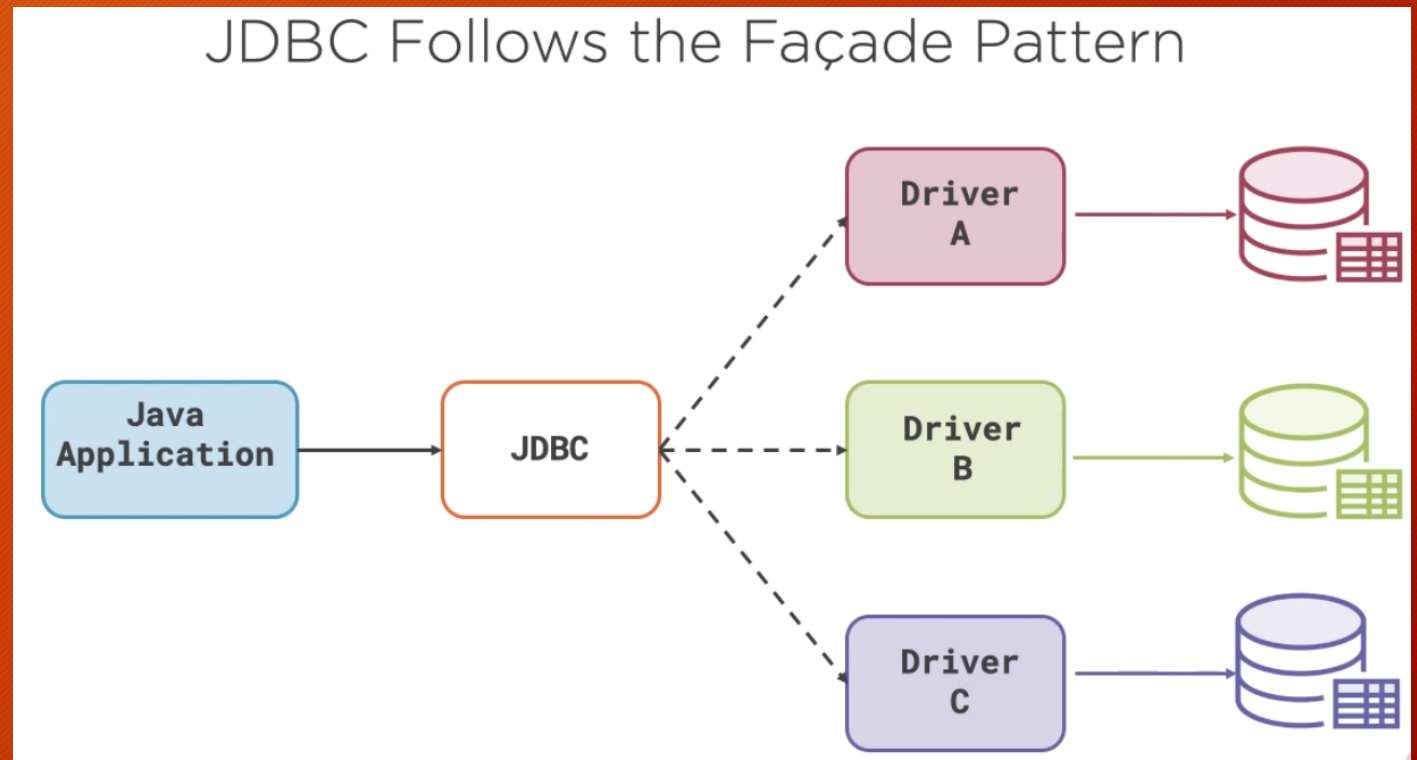
# DriverManager

- getConnection方法
- url:主機名+schema
- User:root
- Password:



# Jdbc connection

- Connection
- Java API
- Java.sql.\*
- Sql語法

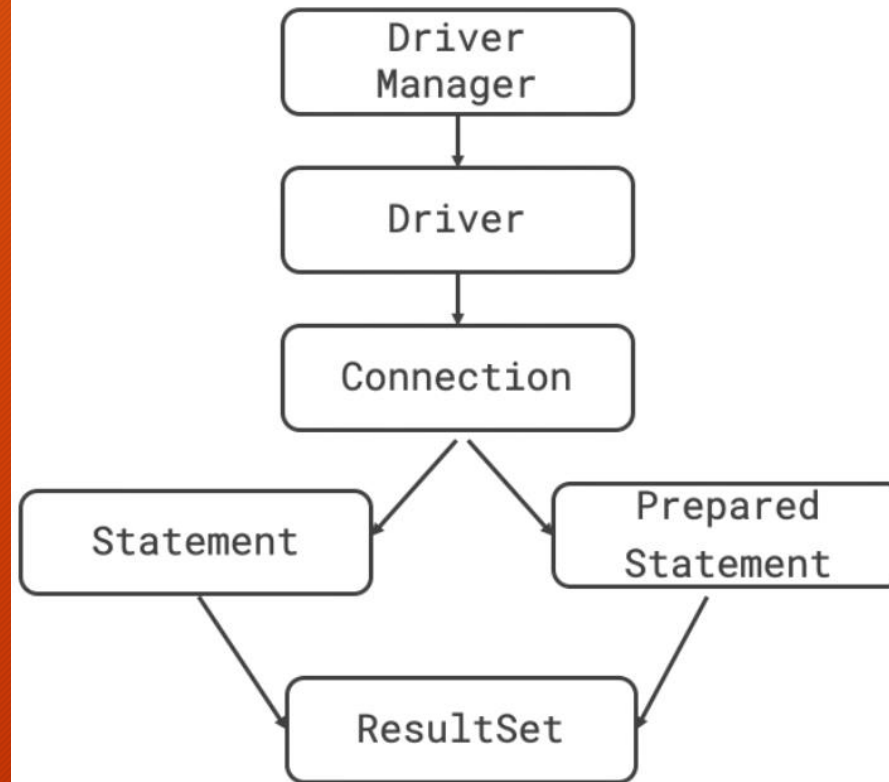




# Jdbc API

- Java.sql.\*
- Connection
- Statement
- preparedStatement

## JDBC Interfaces



# Statement

- Ex:
  - Insert into table() values()
- DriverManager
- Connection
- Statement
- executeUpdate()

```
int executeUpdate(String sql)  
    throws SQLException
```



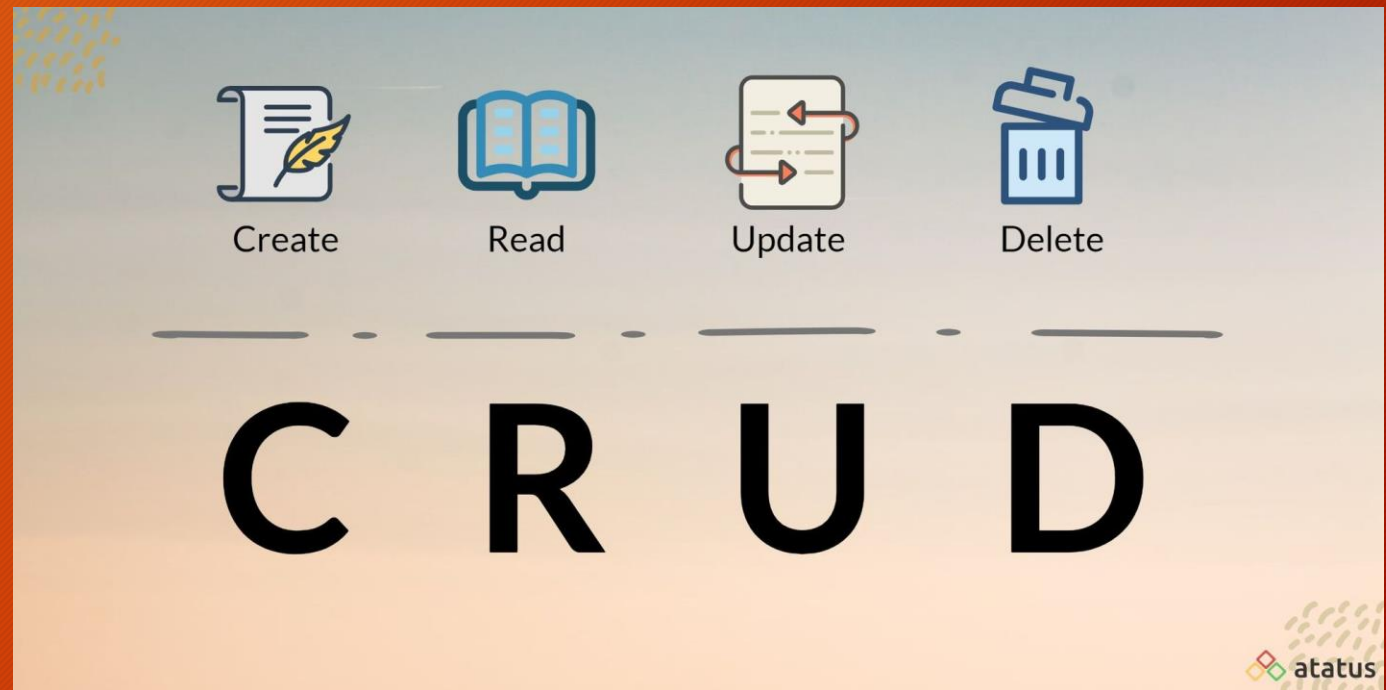
# PreparedStatement

- Ex:
  - Insert into table(name,username,password) values(?,?,?)
- DriverManager
- Connection
- PreparedStatement
- executeUpdate()

```
int executeUpdate()  
    throws SQLException
```

# SQL CRUD

- Create
- Read
- Update
- Delete



# PreparedStatement-create

- 語法
- Insert into 表格(欄位1,欄位2..) values(?,?..)
- Set類型 (索引,值).
- excuteUpdate();



# PreparedStatement-read

- 語法:
- `Select * from 表格`
- `Select * from 表格 where 條件=?`
- `set類型(索引,值);`
- `executeQuery()`方法
- 搭配 `ResultSet`
  - `next()`
  - `get類型("欄位")`

# PreparedStatement-update

- 語法
- Update 表格 set 欄位=?,....where id=?
- Set類型(索引,值)
- excuteUpdate()

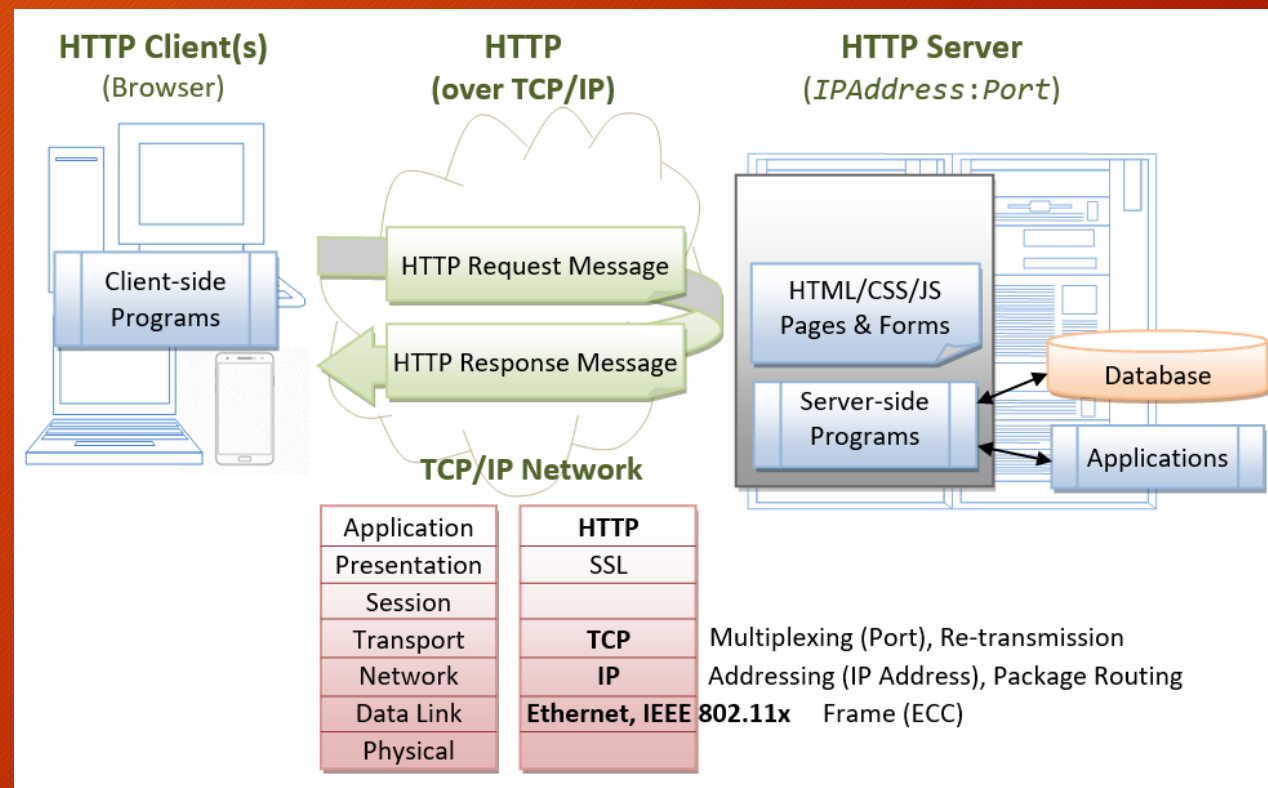
# PreparedStatement-delete

- 語法:
- Delete from 表 where 條件=?
- Set類型(索引,值)
- executeUpdate();

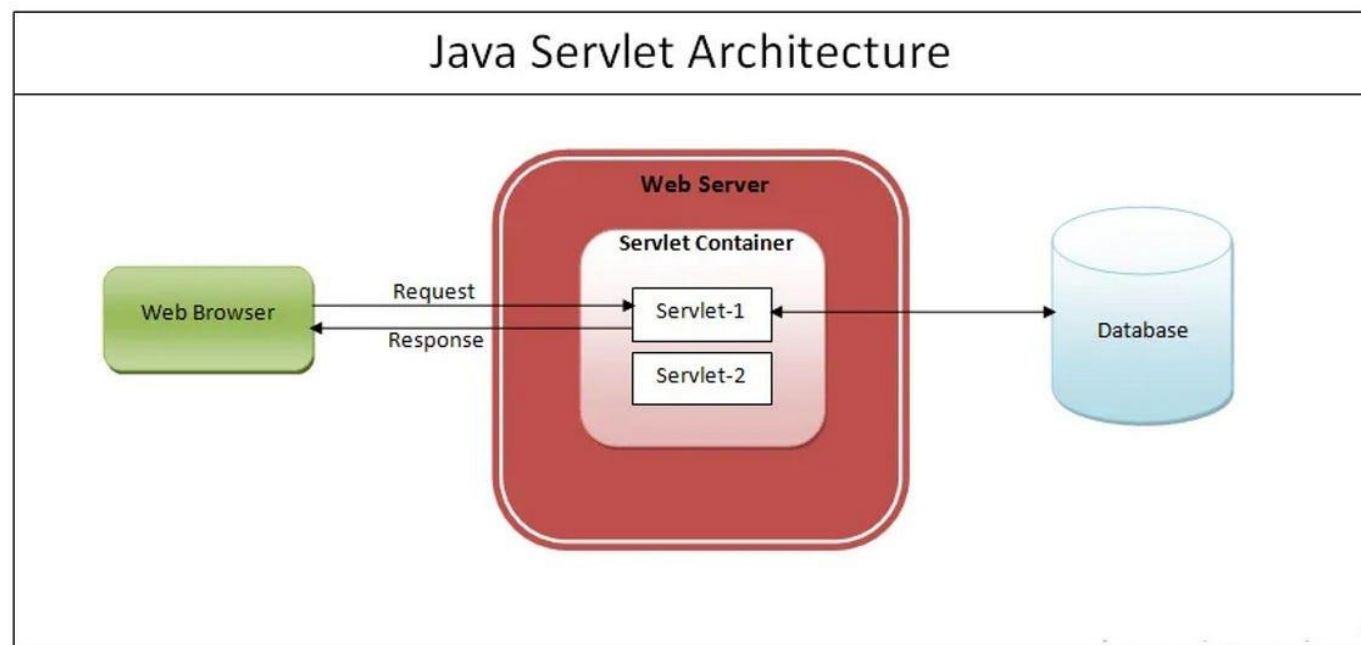


# servlet

- Java
  - 後端web server(tomcate)
  - 與前端 UI(browser)溝通的 api
- TCP/ip協定
  - Request->前端送請求
  - Response->後端作回應
- 擔任UI與資料庫中間的橋樑
- 接收前端經過檢查,計算..再將結果回覆給UI



# Servlet的運作方式



# Servlet 物件生命週期

- Init()
- Service()
  - doGet
  - doPost
- Destroy()
- HttpServletRequest
- HttpServletResponse

